Final report for Diversification for Keyword Search over Structured Databases

He Niu
Department of Computer Science
Saarland University
Tonyniu2008@gmail.com

Abdur Raafiu
Department of Computer and Communication Technology
Saarland University
abdur.raafiu@gmail.com

Abstract

From the original paper, the author provides us an outstanding approach for queries in structured database. However in this report, we will focus on the discussion about this approach, the advantages and drawbacks, and also the comparison between the queries over structured database and in traditional information retrieval. Also the evaluation metrics the author adapts are particular, so we will discuss this metrics in later part. At last, we will provide some of our own opinions towards this approach.

1 Brief Introduction

Compared with the approaches in traditional information, the most important difference of this scheme is that the data we need to analyze is an organized collection of data, and this organized data not only provides us advantages, but also some drawbacks as well.

Advantage: if we know the clear and specific semantic of the query, it is easy for us to retrieve the data. We just need to match the keyword with the specific attribute in database and then we can get the information related to this attribute

Disadvantage: Data may be stored in different tables, and it is very computationally expensive if too much data need to be retrieved, because we need to obtain the data by joining multiple tables.

According to the advantage and disadvantage, the author has two basic ideas in the *Div*Q approach:

- Specify the semantic of the query as clearly as possible before retrieving the data.
- Reduce the amount of data we need to retrieve before operate the retrieving execution. In other word, retrieve the results at the end of the approach.

The problem we meet during the structured database query:

- Ø Single interpretation of a keyword query is not enough;
- Ø Multiple interpretations will yield to overlapping results.

For dealing with the problems above, the author takes the advantage of diversification scheme, to remove some similar interpretations from the results list, provide users a quick glance of the major plausible interpretations, so that the users can simply choose the intended interpretation. Also here diversification should take advantage of the structure of the database:

- Ø Query disambiguation before actual execution, so that we can void computational overhead for retrieving and filtering actual result.
- Ø The interpretation should be based on the structure of the database.

Two main part of the *Div*Q approach:

1. A probabilistic model helps to rank the possible interpretations, to create semantic interpretations. (Ranking based on relevance)
2. A scheme to diversify the search results by re-ranking query interpretations, generating the top-k most relevant and diverse query interpretations. (Take diversification into consideration, ranking based on relevance and diversification)

2 Some basic ideas from traditional IR

1. Diversification
   a) Diversification by classifying search results
      1. based on similarity
      2. understandable for end user
      3. classes are usually pre-defined

2. Take relevance and variance into consideration
   a) to select top-n documents first
   b) order them by balancing the overall relevance of the list against its variance

3. Categorization
   a) takes into account user preferences
   b) Pre-indexing approach for efficient diversification of query results on relational databases

4. Specify the interpretation first
   a) Translate a keyword query into a ranked list of structured queries

3 The DivQ scheme

For a given keyword query in *Div*Q, the scheme first translates it into a set of structured queries

(query interpretations). Then the database will return a broad range of structured query with various semantics. Also unlike existing query disambiguation approaches, *Div*Q also considers diversification into consideration.

Table 1. Structured Interpretations for a Keyword Query

Keyword query: CONSIDERATION CHRISTOPHER GUEST			
Relevance	Top-3 interpretations ranking	Relevance	Top-3 interpretations diversification
0.9	A director CHRISTOPHER GUEST of a movie CONSIDERATION	0.9	A director CHRISTOPHER GUEST of a movie CONSIDERATION
0.5	A director CHRISTOPHER GUEST	0.4	An actor CHRISTOPHER GUEST
0.8	An actor CHRISTOPHER GUEST in a movie CONSIDERATION	0.2	A plot containing CHRISTOPHER GUEST of a movie
…	…	…	…

This Table gives an example of the query interpretations for the keyword query "CONSIDERATION HRISTOPHER GUEST", once ranked only by relevance, and once re-ranked by diversification. Here ranking providing a quick overview over the available classes of results and it is like faceted search, where user navigates and chooses relevant query interpretations.

Bringing Keywords into Structure

A keyword query K is translated in to a structured query Q by the following steps,

- A set of keyword interpretations A_i:k_i, which map each keyword k_i to A_i of an algebraic expression.
- Then joins the keyword interpretations using a predefined query template T

However bring the structure to keywords is limited to the database structure, for example when there is database related to music, the templates should be formed based on that database.

Estimating Query Relevance

Query Relevance of a query interpretation Q is estimated as the conditional probability P(Q|K) that, given keyword query K, Q is the user intended interpretation of K. And here P(Q|K) can be expressed as P(Q | K) = P(I,T | K). Query interpretation Q is composed of a query template T and a set of keyword interpretations I.

$$I=\{A_j:\{k_{j1},k_{jn}\}\mid A_j\in T,\{k_{ji},k_{jn}\}\subset K,\{k_{i1},k_{im}\}\ \mathbf{I}\ \{k_{j1},k_{jn}\}=\{\}\ for\ i\neq j\}$$

To simplify the computation, they assume that

- Each keyword has one particular interpretation intended by the user
- The probability of a keyword interpretation is independent from the part of the query

interpretation the keyword is not interpreted to

Based on these assumptions and Bayes' rule, we can transform the above Formula as

$$P(Q \mid K) \propto \left(\prod_{A_j \in T} P\left(A_j : \{k_{j1}, k_{jm}\} \mid A_j\right) \right) \times \left(\prod_{k_u \in K \cap k_u \in Q} P_u \right) \times P(T)$$

Estimating Query Similarity

The resulting query interpretations should be not only relevant but also as dissimilar to each other. The Jaccard coefficient between the sets of keyword interpretations I contained by Q1 and Q2:

$$Sim(Q_1, Q_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}.$$

The resulting similarity value should always fall in [0,1], 1 means the highest possible similarity.

Combining Relevance and Similarity

- Select the most relevance interpretation as the first interpretation presented to the user
- Each of the following interpretations is selected based on both its relevance and novelty

$$Score(Q) = \lambda \cdot P(Q \mid K) - (1-\lambda) \cdot \sum_{q \in QI} \frac{Sim(Q, q)}{|QI|}.$$

The Diversification Algorithm

- Start with the most relevant query interpretation at the top of L (the ranking list based on relevance)
- Scan the remaining candidate elements in *L*, compare their scores according to the formula above
- Add item to the result list

4. EVALUATION METRICS

In traditional informational retrieval, α-NDCG and S-recall are established evaluation metrics in presence of diversity and relevance. Here the notion of primary key in the database equals the notion of information nugget in α-NDCG metrics and subtopics in S-recall separately.

What is α-NDCG:

- CG (Cumulative Gain):

It is the sum of the graded relevance values of all results in a search result list. And moving a high relevant document d_i above a higher ranked, less relevant, document d_j does not change the computed value for CG. The CG value is defined as

D1	D2	D3	D4	D5	D6
3	3	3	0	1	2

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i$$

So the CG value for this example is CG = 3+2+3+0+1+2 = 11

Ø DCG (Discounted Cumulative Gain)

DCG will take the position of the results into consideration. The results with low ranking will receive low value in this metrics. It is appropriate for the users' preference, that just pay more attention on the top results.

$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

Ø nDCG (Normalized DCG Discounted Cumulative Gain)

For calculating nDCG, we first sort the documents with the decreasing evaluation value order, and then use the DCG formula to calculate IDCG (ideal DCG) value. At last, we obtain the nDCG value by the following formula

$$\mathrm{nDCG_6} = \frac{DCG_6}{IDCG6}$$

Ø α-NDCG (α-Normalized DCG Discounted Cumulative Gain)

Every $G[k]$ is extended with a parameter α, which is a trade-off between relevance and novelty. α is in the interval [0, 1]. 0 means that just care about the relevance, and equals to nDCG in this situation. However increasing α, novelty is rewarded with more credit.

α-NDCG-W

For reflecting the graded relevance assessment on the key words, here we need to take the importance of each keyword into consideration, so we have this weighted α-NDCG

$$G[k] = relevance(Q_k) \cdot (1-\alpha)^r$$

In this formula, r expresses overlap in result list at ranks 1…k-1. For each primary key pk_i in the result of Q_k, count how many query interpretations with pk_i were seen before.

Weighted S-Recall

When search results are related several subtopics, instance recall at rank k (S-recall) is established. It is the number of unique subtopics covered by the first k results, divided by the total number of subtopics.

In database keyword search, a single primary key in the search result corresponds to a subtopic in S-recall. WS-Recall is computed as the aggregated relevance of the subtopics divided by the maximum possible relevance, and it is given as

$$WS-recall@k = \frac{\sum_{pk \in Q_{1\ldots k}} relevance(pk)}{\sum_{pk \in U} relevance(pk)}$$

Here U is the set of relevant subtopics (primary keys).

5 Some discussion about experiments

The datasets in this experiment are two real-world databases from Internet, movie database and lyrics for each. Associated query logs are from MSN and AOL

- 25 single concept queries and 25 multi-queries
- The choose of queries are based on high entropy, so that the effect of diversification will be obvious

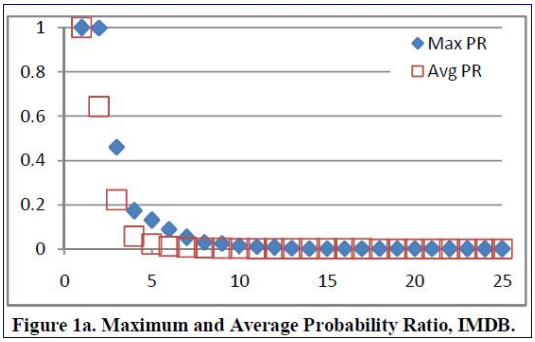


Figure 1a. Maximum and Average Probability Ratio, IMDB.

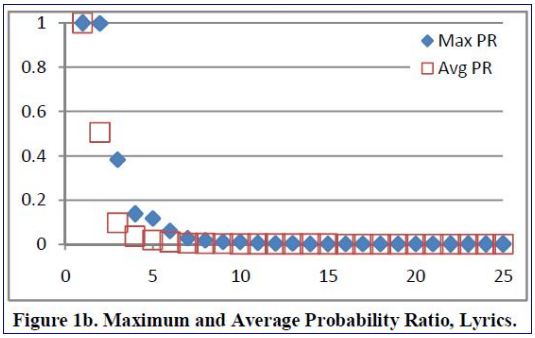
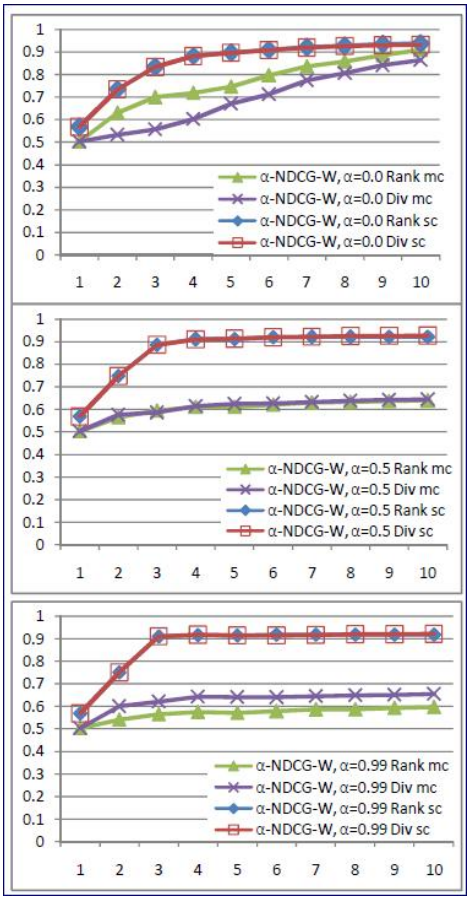


Figure 1b. Maximum and Average Probability Ratio, Lyrics.

Ratio of the probability of a query at rank *i* falls very quickly with their rank, in most of the cases, only top-5 interpretations are meaningful for the end users. And in our opinions, this is because the specific datasets that the author choose. These two datasets are not complicated, so that the queries can't be interpreted to many different semantics. So for each queries, there are only few possible interpretations are meaningful for the end user.



From the α-NDCG-w values comparison in the left table, we can see this diversification algorithm doesn't help much for single-concept interpretation, the effect is very little even invisible in this experiment. Because for single keyword search, the result is either the same with another one or totally different with other result because of the different interpretation, and the algorithm based on the relevance won't provide some results which are same to each other. So the diversification can't provide additional gain for such situation.

With the increasing of α value, we pay more attention on the novelty, when α =0.99 results without novelty are regarded as redundant. And we can see, for multi-concept queries, the effect of diversification is also not obvious. When α =0.99 and k>3, diversification on mc queries outperforms by about 7%. However

We need to mention that we can't set α with so high a value, because we still need to care about relevance more than the novelty.

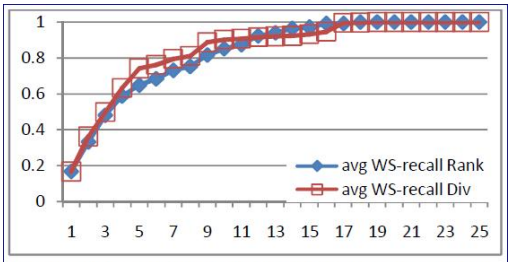


Figure 3a. WS-recall for Ranking and Diversification, IMDB.

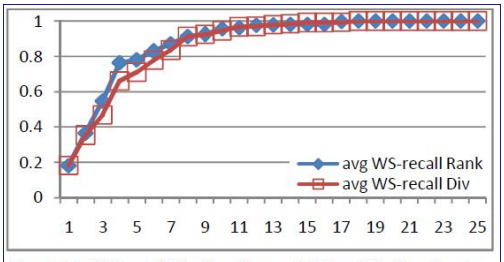


Figure 3b. WS-recall for Ranking and Diversification, Lyrics.

From the result of the SW-recall above, we can see that there is still little improvement. And in our opinions, the problem is still the same with the results of α -NDCG-w, that the diversification doesn't help much for single-concept query, and also considering that ratio of the probability of a query at rank *i* falls very quickly with their rank in user case, it is not hard to imaging the result above.

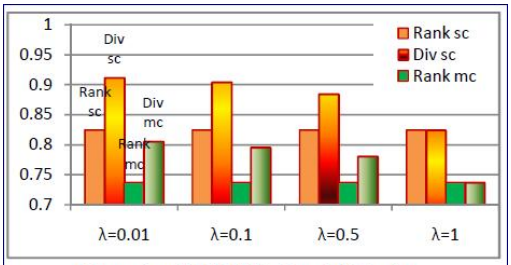


Figure 4. *α*-NDCG-W, *k*=5, *α*=0.99, Lyrics.

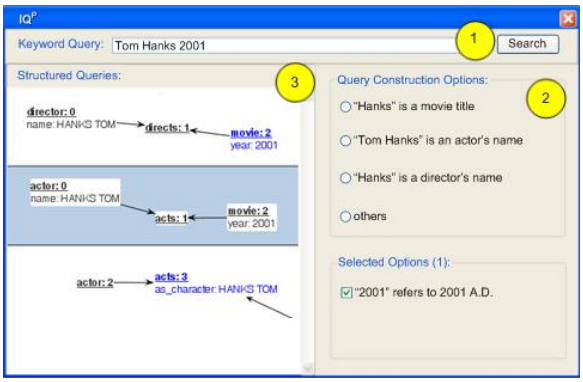
From the result in the left table, we can see the effect from the diversification, the small the λ is, the effect is more obvious. However we need to mention that this initial α -NDCG-w value in Y-axis is 0.7, and this result is under the condition that α = 0.99, and k = 5. So we get the result that still not convincing.

Some opinions about the experiment:

- The diversification doesn't help much for single-concept query. We didn't see much effect from the result, because half of the queries are single-concept query. If we can make the query more complicated, the result maybe more convincing.
- The datasets are two small and simple, but diversification will provide more visible effect for complicated interpretations. In this experiment, the author provides us two datasets and just few tables for each. So we can imagine that the structure of queries will not be complicated for such datasets. If we can find more complicated datasets, maybe the result will be better.
- However the evaluation step has to be done by human, so the complicated datasets may lead to heavier workload, this is a bottleneck of the experiment.

6 A mistake in the presentation

The author has the basic idea that we should retrieve the data as late as possible, so that we can know more about the interpretation and reduce the workload for joining the tables in database.



From the interface, we can see that until this step, we didn't retrieve any data from database. The keywords that shown in the left part are all from the keywords the users typed in, and the system just provides the interpretations. Only after the users choose one of the interpretations in the list, the system will retrieve the data and provide more information. So from the beginning of the algorithm to the end, the authors keep this basic idea that find out the specific interpretation, and only at last retrieve the data.

7 Conclusions and discussions

Advantages

- A good attempt for queries under structured database
- Take diversification into consideration, so that the users can obtain more information
- Evaluation results demonstrate that the novelty of keyword search results improved, even thought the improvement is not very obvious
- The adaptions of evaluation metrics are creative and adaptive in this algorithm.

Disadvantages

- No significant improvement according to the evaluation, as discussed in part 5, we have provided some ideas, hopefully useful for this situation.
- As far as I understood, the templates used in the algorithm have to be edited by human according to the structure of database before executing DivQ. This disadvantage has extremely reduced the applicability and efficiency of this algorithm. So maybe in further work, we can generate the templates automatically.
- Still need more improvements in further work.