International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

# A Technique to Improve Difficult Keyword Queries Over Database

**Deepti S. Deshmukh[1], Simran Khiani[2]**

[1]M.E. Computer Networks Student, G.H. Raisoni Collage of Engineering and Management, Wagholi, Pune, India

[2]Assistant Professor, Department of Information Technology, G.H. Raisoni Collage of Engineering and Management, Wagholi, Pune, India

**Abstract:** ***Estimating query performance is the job of predicting the excellence of results returned to examine in response of a query. Keyword queries on databases provide easy access to data, but often it goes through from low ranking quality. It is defined to get queries with low ranked result, quality to improve the user satisfaction. Post-retrieval predictors analyze the o of top-retrieved documents. This paper introduced a new technique to get high-performance named as NASA, this method is based on k-Nearest Neighbor (k-NN) search on the top-k results of the corrupted version of database. k-NN handles complex functions during the execution and it improve the loss of information. Simultaneously it helps to reduce the execution time.***

**Keywords:** Query Prediction , Top-K , Structured Robustness (SR), k-Nearest Neighbor .

## 1. Introduction

The classical iterated query processing is easy to manipulate, the disadvantage is it gives low performance on modern CPUs due to lack of locality and frequent instruction mispredictions. Several techniques proposed in the past to improve this situation, but some techniques are frequently outperformed a user forms a query according to his information need and a number of documents are presented to the user by the retrieval system in response to the query.[1]

Calculating the process of quality outcome using Query performance prediction of a retrieval system in response to a users query without any relevance information. Compared to the long survey of developing retrieval models to improve performance in IR, research on predicting query performance is still in its early stage. However, some associations are starting to realize the importance of this problem and a number of new methods have been proposed for prediction recently [2]. However, accurate performance prediction with zero-judgment is not an easy task. The major difficulty of performance prediction comes from the fact that many factors, such as the query, the ranking function and the collection, have an impact on retrieval performance. Each factor affects performance to a different degree and the overall effect is hard to predict accurately. The ability to predict query performance has the potential of a fundamental impact both on the user and the retrieval system.

Keyword query interfaces (KQIs) for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list. KQIs are evaluated over the data set that contains structured information. For certain results the investigation of predictions and the factors that allows a measurable and consistent degree of results. Particularly, two types of predictions in the context of information retrieval are set in focus. First, consider users attempts to express their information needs through queries, or search requests and try to predict whether those requests will be of high or low quality. Intuitively, the querys quality is determined by the outcome of the query, that is, whether the results meet the users expectations. Depending on the predicted outcome, action can be taken by the search system in view of improving overall user satisfaction. The second type of predictions under investigation is those which attempt to predict the quality of search systems themselves. So, given a number of search systems to consider, these predictive methods attempt to estimate how well or how poorly they will perform in comparison to each other [3].

The section 2 discusses literature survey related to methods of query prediction , section 3 contains the existing system associated with the a frame work to measure Structured Robustness , section 3 presents a new technique called as NASA used to reduce time complexity in proposed work and section 4 concludes paper.

## 2. Literature Survey

Researchers have projected methods to predict hard queries over structured text documents. It is broadly categorizing these methods are discussed here:

### 2.1 Pre-Retrieval Method

Pre-retrieval methods predict the difficulty of a query without computing its results. These methods usually use the statistical properties of the terms in the query to measure specificity, ambiguity, or term-relatedness of the query to predict its difficulty.[2] Query performance predictors in this category estimate the effectiveness of a query by the query terms specificity. Although pre-retrieval predictors do not consider the ranked list of results returned by the retrieval system for a given query, they can still rely on collection statistics to infer how difficult it will be for the system to

rank the documents according to the query.

### 2.2 Post-retrieval method

Post-retrieval methods are based on Clarity Score. Clarity Score is based on the intuition that the top ranked results of an unambiguous and thus well performing query are topically cohesive, whereas the result list of an ambiguous and thus poorly performing query will cover. This approach predicts the difficulty of a query more accurately than pre-retrieval based methods for text documents. Adaptive Clarity improves over Clarity Score and other state-of-the-art pre and postretrieval prediction approaches.[4] Adaptive Clarity proposed adaptations to Clarity Score. The approach to setting the number of feedback documents automatically is described, followed by the frequency dependent term selection. Adapted Clarity that is our adaptations on Clarity Score, on the TREC corpora and query sets already shown. Apart from Clarity Score, for reasons of comparison include a number of the best performing pre-retrieval predictor scores as already presented in the previous chapter. It is also implemented four postretrieval prediction methods, which base their predictions on different types of information:[7]

1) Ranking Robustness (based on document)
2) Query Feedback (based on query)
3) Normalized Query-Commitment (based on retrieval score) and
4) Autocorrelation (based on document)

Correctly identifying the ranking of retrieval systems can also be advantageous in a more practical setting when relying on different retrieval approaches (such as Okapi and Language Modeling) and a single body. Intuitively, different types of queries benefit from different retrieval approaches. If it is possible to predict which of the available retrieval approaches will perform well for a particular query, the best predicted retrieval strategy can then be selected. Overall, this would lead to an improvement in effectiveness[13].

A query predicted to perform poorly, may not necessarily be ambiguous but may just not be covered in the body to which it is submitted. Also, identifying difficult queries related to a particular topic can be a valuable asset for collection keepers who can determine what kind of documents are expected by users and missing in the collection. Another important factor for collection keepers is the find ability of documents, that is how easy is it for searchers to retrieve documents of interest Predictions are also important in the case of well-performing queries.[8] When deriving search results from different search engines and corpora, the predictions of the query with respect to each body can be used to select the best body or to merge the results across all corpora with weights according to the predicted query effectiveness score. Also, consider that the cost of searching can be decreased given a multiple partitioned body, as is common practice for very large corpora. If the documents are partitioned by, for instance, language or by topic, predicting to which partition to send the query saves time and bandwidth, as not all partitions need to be searched. Moreover, should the performance of a query appear to be sufficiently good, the query can be improved by some affirmative action such as automatic query expansion with pseudo-relevance feedback, In pseudo-relevance feedback it is assumed that the top K retrieved documents are relevant and so for a query with low effectiveness most or all of the top K documents would be irrelevant. Notably, expanding a poorly performing query leads to query drift and possibly to an even lower effectiveness while expanding queries with a reasonable performance and thus a number of relevant documents among the top K retrieved documents is more likely to lead to a gain in effectiveness. Another recently proposed application of prediction methods is to shorten long queries by filtering out predicted extraneous terms, in view of improving their effectiveness.

### 2.3 Motivation

The motivation for this work is to improve user satisfaction in retrieval, by enabling the automatic identification of well performing retrieval systems as well as allowing retrieval systems to identify queries as either performing well or poorly and reacting accordingly. It includes a thorough evaluation of existing prediction methods in the literature and proposes an honest appraisal of their effectiveness. Here carefully enumerate the limitations of contemporary work in this field, propose enhancements to existing proposals and clearly outline their scope of use. Ultimately, there is considerable scope for improvement in existing retrieval systems if predictive methods are evaluated in a consistent and objective manner.

## 3. Existing System

Some methods use machine learning techniques to learn the properties of difficult queries and predict their hardness . They have similar limitations as the other approaches when applied to structured data.

### 3.1 Mathematical Implementation Strategy

The Structured Robustness Algorithm (SR Algorithm), which computes the exact SR score, based on the top K result entities. Each ranking algorithm uses some statistics about query terms or attributes values over the whole content of DB. Some examples of such statistics are the number of occurrences of a query term in all attributes values of the DB or total number of attribute values in each attribute and entity set.[1] These global statistic stored in M (metadata) and I (inverted indexes) in the SR Algorithm pseudo code. SR Algorithm generates the noise in the DB on-the-fly during query processing. Since it corrupts only the top K entities, which are anyways returned by the ranking module, it does not perform any extra I/O access to the DB, except to lookup some statistics. Moreover, it uses the information which is already computed.

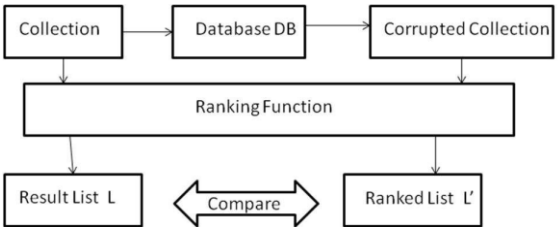


**Figure 1:** Execution flow of SR Algorithm.

### 3.2 Corruption of Structured Data

The first challenge in using the Ranking Robustness Principle for databases is to define data corruption for structured data. For that, a database DB using a generative probabilistic model based on its building blocks, which are terms, attribute values, attributes, and entity sets. A corrupted version of DB can be seen as a random sample of such a probabilistic model. Given a query Q and a retrieval function g, rank the answers in DB and its corrupted versions DB',DB",..... to get ranked lists L and L', L',....., respectively. The less similar L is to L', L", the more difficult Q will be.

Database DB as a triplet (S, T ,A), where S, T , and A denote the sets of entity sets, attributes, and attribute values in DB, respectively. $|A|, |T|, |S|$ denote the number of attribute values, attributes, and entity sets in the database, respectively. Let V be the number of distinct terms in database DB. Each attribute value Aa $\epsilon$ A,1 $\leq$a, $\leq |A|$ can be modeled using a V-dimensional multivariate distribution Xa = (Xa,1, . . . ,Xa,V), where Xa,j $\epsilon$ Xa is a random variable that represents the frequency of term wj in Aa. The probability mass function of Xa is:

$$fxa(\bar{X}a = Pr(Xa; 1 = xa; 1,\ldots.., Xa; V = xa; V)$$

where $\bar{X}a$ = xa; 1,….., xa; V and xa; j $\epsilon$ $\bar{X}a$ are non-negative integers.
Random variable XA = (X1, . . . ,X$|A|$) models attribute value set A, where XA is a vector of size V that denotes the frequencies of terms in A. Hence, XA is a $|A|$ V matrix. The probability mass function for XA is:

$$fxA = \bar{X} = fxA(\bar{X}1,\ldots.., \bar{X}|A|)$$

where Xa $\epsilon$ X are vectors of size V that contain non-negativeintegers. The domain of X is all $|A|$ V matrices that contain non-negative integers, i.e. M($|A|$ V).

### 3.3 Structured Robustness calculation.

To compute the similarity of the answer lists using rank correlation. It ranges between 1 and -1, where 1, -1, and 0 indicate perfect positive correlation, perfect negative correlation, and almost no correlation, respectively. Equation computes the Structured Robustness score (SR score), for query Q over database DB given retrieval function g:

$$SR(Q; g;DB;XDB) = ESim(L(Q; g;DB);L(Q; g;X_{DB}))$$

### 3.4 Basic Estimation Techniques For SR using Top-K results

Generally, the basic information units in structured data sets, attribute values, are much shorter than text documents. Thus, a structured data set contains a larger number of information units than an unstructured data set of the same size. For instance, each XML document in the database data centric collection constitutes hundreds of elements with textual contents. Hence, computing Equation for a large DB is so inefficient as to be impractical. Hence, similar to, corrupt only the top-K entity results of the original data set. It will re-rank these results and shift them up to be the top-K answers for the corrupted versions of DB. In addition to the time savings, it show that relatively small values for K predict the difficulty of queries better than large values. For instance, we found that K = 20 delivers the best performance prediction quality in our datasets. We discuss the impact of different values of K in the query difficulty prediction quality [1].
Number of corruption iterations (N)
Computing the expectation in Equation for all possible values of x is very inefficient: Hence; estimating expectations using N ¿ 0 samples over M $|A|$ V). N is preferred for the sake of efficiency.

### 3.5 Efficiency of SR Algorithm

The time to compute the SR score only depends on the top-K results, since only the top-K results are corrupted and re-ranked. Increasing the data set size will only increase the query processing time; the complexity of data schema could have impact on the efficiency of our model. The propose approximation algorithms to improve the efficiency of SR Algorithm.

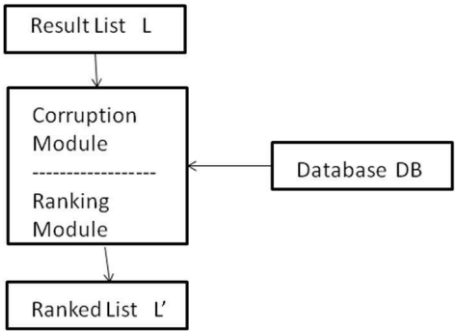


**Figure. 2:** Execution flow of SGS Approximation.

QAO Approximation called as Query-specific Attribute values Only Approximation. QAO-Approx corrupts only the attribute Values that match at least one query term. The number of attribute values that contain at least one query term is much smaller than the numbers of all attribute values in each entity It Can significantly decrease the time spent on corruption if result corrupt only the attribute values that contain query terms.

SGS Approximation called as Static Global Status Approximation. SGS-Approx needs to decrease N to with the

SR-time (Corruption time + Rerank time).It will combine the corruption and ranking module together. This way re-ranking is done on-the-fly during corruption. The Ranking algorithm proposed in this method is PRMS (Probabilistic Retrieval Model for Structured Data) to get the better results of approximation algorithm. It computes the language model of each attribute value smoothed by the language model of its attribute. It assigns each attribute a query keyword-specific weight, which specifies its contribution in the ranking score. It computes the keyword-specific weight j(q) for attribute values whose attributes are Tj and query q as,

$$\mu_j(q) = P(q/T_j) / (\sum T \epsilon DB * P(q/T))$$

### 3.6 Performance Study

In this paper the performance of SR score and SGS Approximation is studied.

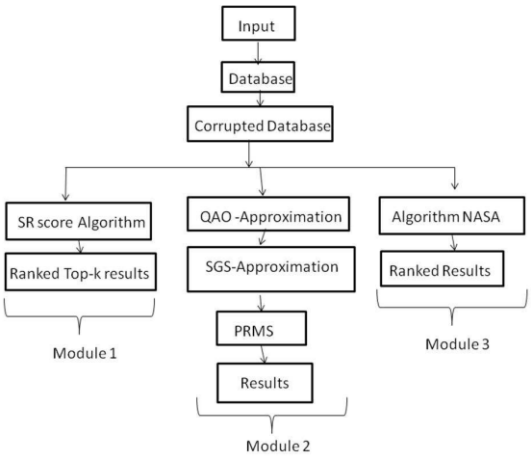


**Figure 3:** Execution flow of Modules.

### SR Algorithm

In first module SR Algorithm implementation is done as shown in Figure.3, which is the existing in our reference [1].SR-time mainly consists of two parts: the time spent on corrupting K results and the time to re-rank the K corrupted results. We have reported SR-time using (corruption time + re-rank time) format. SR Algorithm incurs a considerable time overhead on the query processing which is higher on query processing . Figure. will gives expected result of SR score.

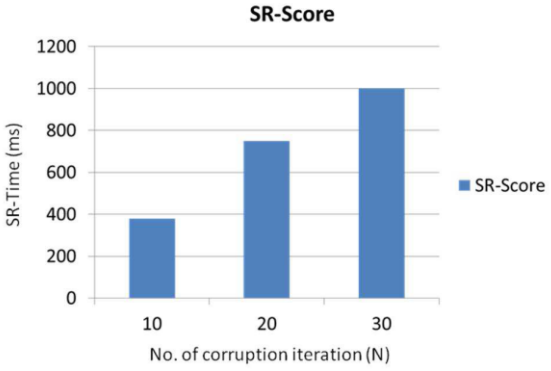


**Figure 4:** Result of SR-Score.

### SGS-Approximation

In next module SR Algorithm implementation is done as shown in Figure 2, which is the proposed work of our reference[1].QAO measure the prediction effectiveness for smaller values Of N.SGS Approximation on the database ,re-ranking is done during the corruption. SR time is mentioned as corruption time only. Figure 5 will gives expected results of SGSApproximation. According to our performance study of QAOApproximation, SGS-Approximation, and the combined result over database ,It delivers the best balance of improvement in efficiency and reduction in effectiveness for database. It will achieves high prediction accuracy as compared with SR score algorithm. To improve performance of SR score a new technology is used named as NASA. It will reduce the SR time of execution and gives high prediction accuracy.

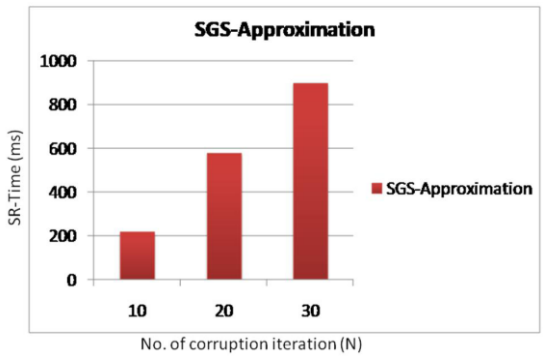


**Figure 6:** Result of SGS Approximation.

### 3.7 Disadvantage of Existing System

The average computation time of SR score by means of SR Algorithm and compare it to the average query processing time using PRMS for the queries in our query workloads. SR Algorithm incur a considerable time overhead on the query processing. This operating cost is higher for queries over the database. QAO measure the prediction effectiveness for smaller values of N using average correlation score. The results of applying SGS-Approx on the database. Since re-ranking is done on-the-fly during the corruption, SR-time is reported as corruption time only.

### 3.8 Problem Definition

The time to compute the SR score only depends on the top-K results, since only the top-K results are corrupted and reranked. Increasing the data set size will only increase the query processing time; the complexity of data schema could have impact on the efficiency of our model. To overcome this problem a new method named as NASA is using to reduce time complexity with the help of k- Nearest Neighbor algorithm (k- NN)

## 4. Proposed Work.

A computational problem instance has an input and an objective it wants to compute on that input. An algorithm is a

procedure to compute the objective function. [13]. SR Algorithm incur a considerable time overhead on the query processing. The time to compute the SR score only depends on the top-K results, since only the top-K results are corrupted and re-ranked. To overcome this problem of time overhead new algorithm is going to launched called as NASA. NASA means K-Nearest Neighbor algorithm Algorithm implementing in structured Robustness Algorithm. Algorithm 1 will explain the execution flow of NASA.In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a nonparametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. k-NN has some strong consistency results. As the amount of data approaches infinity, the algorithm is guaranteed to yield an error rate. k-NN uses the previous instances as a model for future instances and prediction for the current instance is chosen as the classification of the most similar previously observed instance. Instances with correct classifications (predictions) (xi, f (xi)) are stored in memory. Given, a new instance is xq, and the prediction is most similar to is instance xk.

### 4.1 The execution flow of K-Nearest Neighbors is algorithm as follows

First Determine the parameter (K ) number of nearest neighbors. then Calculate the distance between the query instance and samples. With use any distance algorithm. Sort the distances for all the samples and determine the nearest neighbor based on the K-th minimum distance Use the majority of nearest neighbors as the prediction value

---

**Algorithm 1. NASA**

---

**Input:** Query Q, Top-k, result L of Q ranking function g,Metadata M, Inverted index I, Number of Corruption iteration n, Process X, Number of Neighbour Kn(1,2,.....,n-1), Database DB

---

1. SR ←0;C ← 0; // C Caches Keywors in Q
2. FOR i = 1 → N DO
3. I'←I;M' ←M;L' ←L; // Corrupted copies of I , M and L
4. For each result R in L Do
5. For each query Xl → N-1 DO
6. Find Kn Nearest Neighbor
7. Use Euclidean distance between two instances d(Xm,Xn) = Sqrt(Sumeforr = 1toN(ar(Xm) - ar(Xn))2)
8. A' ←A;// Corrupted version of A
9. For each keywords w in Q Do
10. Compute = of w in A' for keywords in Q as needed but not in C ,calculate and cache them
11. IF w varies in A' and A THEN
12. Update A',M' and entry of w in I'
13. Add R' to L'
14. Rank L' using g which returns L based on I',M'
15. SR ← Sim(L;L')
16. RETURN SR←SR/N; // AVG score over N rounds

---

### 4.1 Advantages of k-NN

1. K-Nearest Neighbors algorithm just store instances so the utilization of memory space can be less.
2. K-Nearest Neighbors algorithm can handle complex target functions and it can improve the loss of information.

## 5. Conclusion

The conclusion of the frame work shows Post retrieval methods predict the difficulty of a query with computing its results. The computation the query prediction based on SR algorithm gives the outcomes of corrupted database in re-ran ked form. Execution time depends on Increasing the data set size will only increase the query processing time The complexity of data schema could have impact on the efficiency of our model. k- NN algorithm helps to reduce time complexity and outcomes are in the form of ranking. So NASA will help to give a more effective result in ranking order with respect to less time of execution.

## References

[1] Shiwen Cheng, Arash Termehchy, And Vagelis Hristidis, Efficient Prediction Of Difficult Keyword Queriesover Databases IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014.
[2] JoaquIn PErez-Iglesias And Lourdes Araujo, Evaluation Of Query Performance Prediction Methods By Range, E. Chavez And S. Lonardi (Eds.): Spire 2010, Lncs 6393, Pp. 225236, 2010.
[3] Joaqun Perez-Iglesias and Lourdes Araujo, Evaluation of Query Performance Prediction Methods by Range E. Chavez and S. Lonardi (Eds.): SPIRE 2010, LNCS 6393, pp.225236, 2010.
[4] Ben He and Iadh Ounis ," Inferring Query Performance Using Pre-retrieval Predictors", Department of Computing Science University of Glasgow.
[5] Oren Kurland1, Anna Shtok1, David Carmel2, And Shay Hummel, A Unified Framework For Post-Retrieval Query- Performance Prediction
[6] Thanh Tran And Lei Zhang Keyword Query Routing, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 2, February 2014 363 J. Clerk Maxwell, A Treatise On Electricity And Magnetism, 3rd Ed., Vol. 2. Oxford: Clarendon, 1892, Pp.68-73.
[7] Luca Di Angelo1 And Luigi Giaccari,," An Efficient Algorithm For The Nearest Neighbourhood Search For Point Clouds," Ijcsi International Journal Of Computer Science Issues, Vol. 8, Issue 5, No 1, September 2011.
[8] Claudia Hauff,"Predicting The Effectiveness Of queries And Retrieval Systems", January 29, 2010.
[9] Xiaogang Wang, Member, IEEE , Shi Qiu, Ke Liu, and Xiaoou Tang, Fellow, IEEE,"Web Image Re-Ranking

Using Query-Specific Semantic Signatures",VOL. 36, NO. 4, APRIL 2014.

[10] Yang Kehua, Abdoullahi Diasse Hunan University, A Dynamic Materialized View Selection In A Cloud-Based Data Warehouse, Ijcsi International Journal Of Computer Science Issues, Vol. 11, Issue 2, No 1, March 2014.

[11] Yun Zhou And W. Bruce Croft, Measuring Ranked List Robustness For Query Performance Prediction, Jun 10,2007.

[12] Polynomial Time Algorithms , Computational Complexity , June 4th, 2009.

[13] Jitao Sang And Changsheng Xu, Browse By Chunks: Topic Mining And Organizing On Web-Scale Social Media, Acm Transactions On Multimedia Computing, Communications And Applications, Vol. 7s, No. 1, Article 30, Publication Date: October 2011.

[14] Leong Hou U, Hong Jun Zhao, Man Lung Yiu, Yuhong Li, and Zhiguo Gong,"Towards Online Shortest Path Computation", VOL. 26, NO. 4, APRIL 2014.

[15] Jagbeer Singh, Bichitrananda Patra, Satyendra Prasad Singh, An Algorithm To Reduce The Time Complexity Of Earliest Deadline First Scheduling Algorithm In Real-Time System, (Ijacsa) International Journal Of Advanced Computer Science And Applications,Vol. 2, No.2, February 2011.

[16] S. Cheng, A. Termehchy, and V. Hristidis,2012 , Predicting the effectiveness of keyword queries on database..

[17] Chihung Chi;Y eZhou; andXiaojun; Performance Prediction For Performance Sensitive Queries; Issnll1007-0214ll08/10llpp618-628 V ol:18;No6;December2013: