Ontology Based Prediction of Difficult Keyword Queries

Lubna.C*, Kasim K
Pursuing M.Tech (CSE)*, Associate Professor (CSE)
MEA Engineering College, Perinthalmanna
Kerala, India
lubna9990@gmail.com, kasim_mlp@gmail.com

Abstract— **Keyword queries are used to access data from databases. To improve the performance of querying system it would be useful to identify queries with low ranking quality. The degree of difficulty of a keyword query can be measured by analyzing the characteristics of difficult queries. The difficulty of a query over a database is predicted by considering the structure and the content of the database and the query results. An improved structured robustness algorithm based key word prediction is proposed by utilizing ontological mapping carried out using the Word Net tool.**

Keywords- Keyword query; database; difficult query; structured robustness; ontology

I. INTRODUCTION

Data mining is the process of extracting information from data sets or other data structures like databases. Keyword queries can be used to access data from databases. Databases are used to organize the data in a structured manner to model aspects of reality. Databases are managed by database management system which is software used to define, construct and manipulate database. [2]

Keyword search is a technique utilized for data mining. It provides access to stored information. Keyword queries are employed by means of keyword query interfaces which retrieve possible potential answers from datasets. The retrieved results are analyzed to measure the power of a query over a database in retrieving the desired results. Some queries exhibits low ranking quality. Queries with low ranking quality are termed as difficult or hard queries.

Difficult queries results in lower performance of data mining system. Identifying difficult queries will help in improving the performance by formulating methods to overcome the complexity involved in resolving the query. It is possible to optimize the query during query processing. Developing alternative queries or reformulating the query helps to overcome the difficulty involved with queries.

This work mainly focusses on efficient prediction of difficult queries and query results. Improved structured robustness algorithm is developed for performing the prediction. It relies on the ontology based Word Net tool for considering the semantic meaning of the query during query processing.

Algorithms and statistical measurements employed for implementing the underlying prediction procedure have been discussed in the following sections. Second section reviews the related concepts of the technique and discusses previous related works in the field. Proceeding sections gives details regarding the high level design architecture, module description and implementation of the proposed system.

II. BACKGROUND

A. Query Processing

Fig 1 shows the main steps involved in query processing. Information is retrieved from database based on queries which are being processed and optimized.

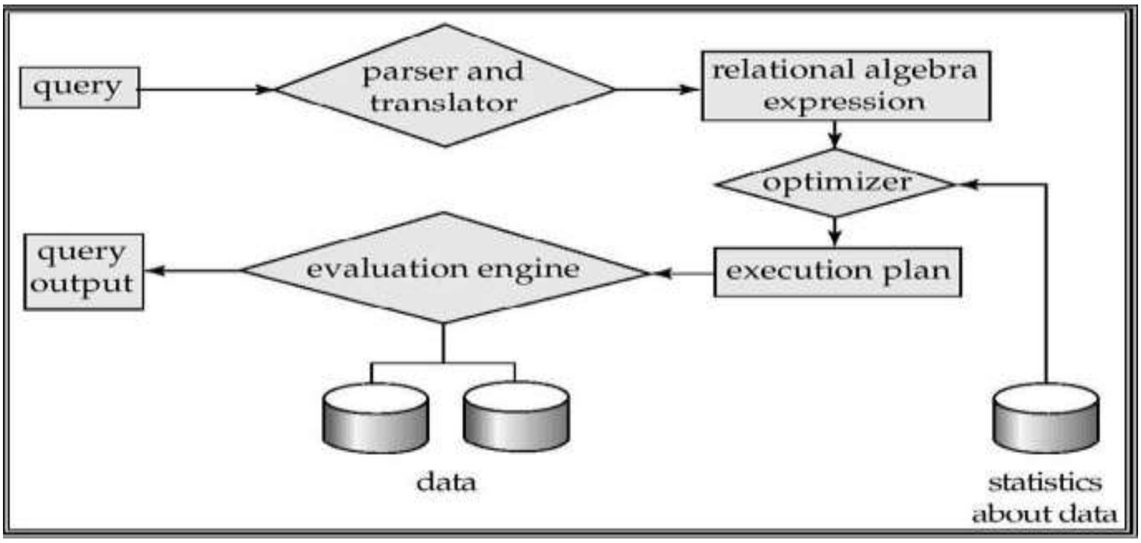


Figure 1: Query processing

B. Labelling semantic relations

Word Net is a lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, expressing a distinct concept. Synonyms are interlinked by means of conceptual-semantic and lexical relations [3].

The semantic relation within the query is labelled by using the Word Net module. It enhances the possible interpretation of query over all possible domains. Incorporating the ontology module along with structured robustness algorithm will help to improve the prediction.

III. RELATED WORKS

G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan [4] describe techniques for keyword searching and browsing on databases that developed as part of the BANKS system (Browsing ANd Keyword Searching). The BANKS system enables data and schema browsing together with keyword-based search for relational databases. BANKS enables a user to get information by typing a few keywords, following hyperlinks, and interacting with controls on the displayed results; absolutely no query language or programming is required. It presents a novel and efficient heuristic algorithms for executing keyword queries. Issue involved with this system is the computation overhead and lower accuracy. BANKS encompasses faster processing with minimum user effort.

V. Ganti, Y. He, and D. Xin [5] consider the entity database as a single relation, which involves joins over multiple base relations on a given baseline keyword search interface over an entity database, to map keywords in a query to predicates or ordering clauses. Techniques discussed in this work measure the correlation between a keyword and a predicate by analyzing two result sets, from the baseline search engine, of a differential query pair with respect to the keyword. This work addresses the incompleteness and impreciseness issues under the context of keyword search over entity search. The modified queries may be cast as either SQL queries to be processed by a typical database system or as keyword queries with additional predicates or ordering clauses to be processed by a typical IR engine. This work primarily focuses on translating keyword queries to SQL queries. This technique can be easily adopted by IR systems to improve the recall rate.

V. Hristidis, L. Gravano, and Y. Papakonstantinou [6] contributes for the incorporation of IR-style relevance ranking of tuple trees into our query processing framework. Keyword search exploits single attribute relevance ranking results if the RDBMS of choice has text indexing capabilities. This scheme relies on the IR engines of RDBMSs to perform such relevance ranking at the attribute level, and handles both AND and OR semantics. Existing query processing strategies for keyword search over RDBMSs are inherently inefficient, since they attempt to capture all tuple trees with all query keywords. Thus these strategies do not exploit a crucial characteristic of IR-style keyword search. The main contribution of this paper is the presentation of efficient query processing techniques for our IR-style queries over RDBMSs. This hybrid algorithm has the best overall performance.

Y. Zhou and W. B. Croft [7] developed a method for predicting query performance by computing ranking robustness which refers to a property of a ranked list of documents that indicates how stable the ranking is in the presence of uncertainty in the ranked documents. The idea of predicting retrieval performance by measuring ranking robustness is inspired by a general observation in noisy data retrieval that the degree of ranking robustness against noise is positively correlated with retrieval performance. Regular documents also contain “noise” if we interpret noise as uncertainty. The robustness score significantly and consistently correlates with query performance in a variety of TREC test collections.

V. Jain and S. Prasad [8] discusses about an ontology based information retrieval model. The major contribution of this paper is to provide semantic result and at the same time reducing the error rate significantly. In their research they explained the development of an ontology-based model for the generation of metadata for audio, and the selection of audio information in a user customized manner. Also conclude how the ontology they proposed can be used to generate information selection requests in database queries.

J. Alvez, J. Atserias, J. Carrera, S. Climent, E. Laparra, A. Oliver, and G. Rigau [9] discusses a work named Ontological annotations that would identify real-world entities alongside properties and relations that characterize the entities’ attributes and role in their textual context, with respect to reference ontology. Adding these annotations to unstructured or semi-structured data is a basic requirement to make Semantic Web technologies work.ropose techinque for developing an analytical platform that (1) provides a WordNet-based ontology offering a manageable and yet comprehensive set of concept classes, (2) leverages the lexical richness of WordNet to give an extensive characterization of concept class in terms of lexical instances, and (3) integrates a class recognition algorithm that automates the assignment of concept classes to words in naturally occurring text.

IV. PROPOSED WORK

A. System Architecture

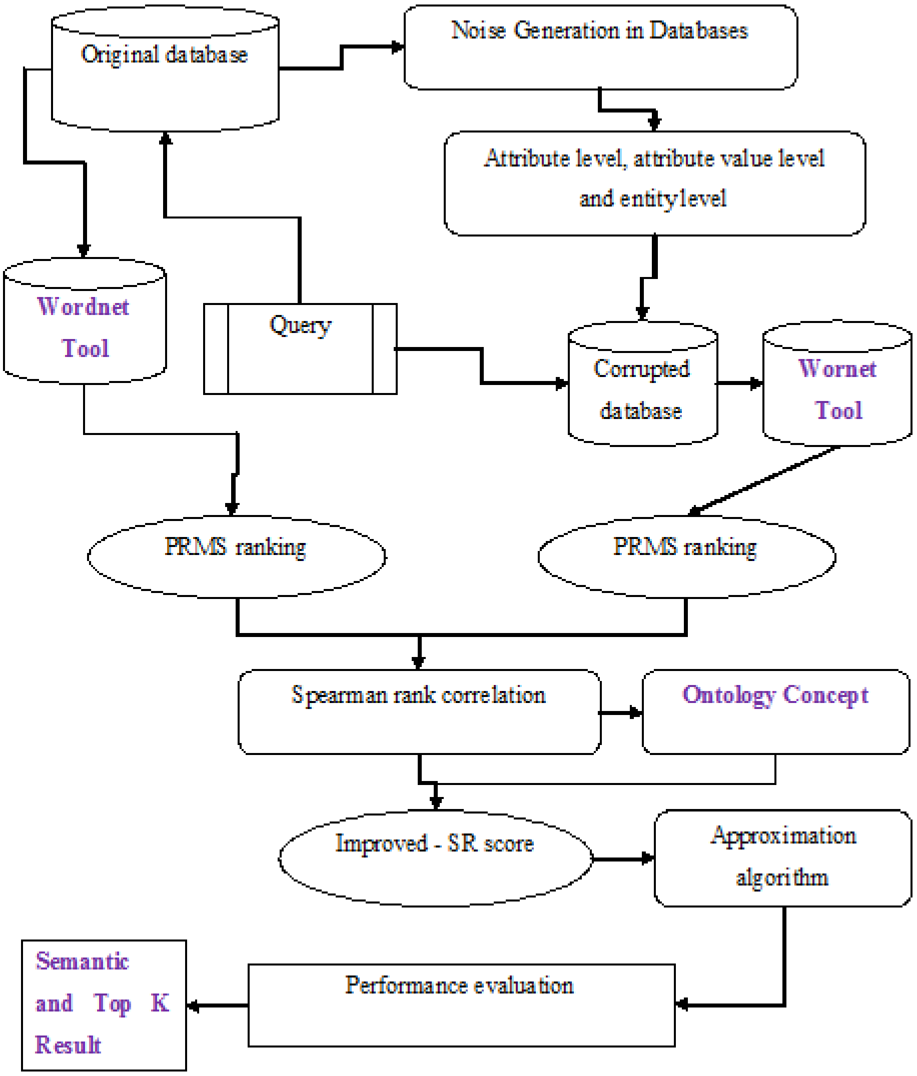


Figure 2: System Architecture

B. Module Description

- Noise Generation in Databases
- Ranking in original database with ontology
- Ranking in corrupted database with ontology
- Improved Structured Robustness Algorithm
- Approximation algorithms

Noise Generation in Databases

In order to compute SR, we need to define the noise generation model for database . We will show that each attribute value is corrupted by a combination of three corruption levels: on the value itself, its attribute and its entity set.

The corruption model must reflect the challenges about search on structured data, where we showed that it is important to capture the statistical properties of the query keywords in the attribute values, attributes and entity sets.

Ranking in original database with ontology

The probabilistic retrieval model for semi structured data (PRMS) can use them as weights for combining the score from each element into a document score, as follows:

$$P(Q|d) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_M(E_j|q_i) P_{QL}(q_i|e_j)$$

The rationale behind this weighting is that the mapping probability is the result of the inference procedure to decide which element the user may have meant for a given query term. Using Word Net tool search engine can find efficiently the meaning of query. Based on the page count it will rank using the I-SR algorithm.

Ranking in corrupted database with ontology

Improved Structured Robustness Algorithm

We compute the similarity of the answer lists using Spearman 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 3 computes the Structured Robustness score (*SR* score), for query *Q* over database *DB* given retrieval function *g*:

SR(Q, g,DB,XDB) = E{*Sim(L(Q, g,DB), L(Q, g,XDB))*}

where *Sim* denotes the Spearman rank correlation between the ranked answer lists.

Algorithm:

```
1. Consider the input query Q, Inverted Index I, Number of relations exist in the ontology R, Finite Set O (Ontology), Similarity word, Top k result List L of Q by ranking function g, Number of corruption iteration N.

2. ISR ← 0 ; C ← {};

3. For i=1 → N do
```

4. For (int i=0; i<word length; i++);

5. WordInformation[i]=Find WordInnformation for Words(i) by Wordnet

6. For (j=1; j<R;j++)

7. R=R_{ij} // no.of relations exist in ontology concept.

8. For each concept of ontology

9. IfType(wordType.word) is a noun then

10. wordDistance= wordType.wordGetSimilarity (concept of ontology), return

11. Build similarity matrix

12. Improve the relevance score value based on the web page counts.

13. Then do the same process for corrupted database.

14. For each result R in L do

15. For each attribute value A in R do

16. Obtain corrupted version of A.

17. For each keyword w in Q do

18. Calculate number of w in corrupted A.

19. Perform for all words in a given query.

20. Read the character after " +", " -" and hyphen using I-SR algorithm.

21. Update all the metadata values.

22. Compute the ranking using function g and correlation method.

23. Obtain ISR result with semantic result using ontology concept.

24. Return ISR+=Semantic and Similarity top k result (L, L')

Algorithm shows the Improved Structured Robustness Algorithm (SR Algorithm), which computes the exact I-SR score based on the top *K* result entities.

Approximation algorithms

In this section, we propose approximation algorithms to improve the efficiency of ISR Algorithm. Our methods are independent of the underlying ranking algorithm.

Query-specific Attribute values Only Approximation (QAO-Approx): QAO-Approx corrupts only the attribute values that match at least one query term.[1]

Static Global Stats Approximation (SGS-Approx): SGS Approx uses the following observation: Given that only the top-K result entities are corrupted, the global DB statistics do not change much.[1]

V. RESULTS AND DISCUSSION

Recall

Recall value is calculated is based on the retrieval of information at true positive prediction, false negative. Recall is the fraction of relevant instances that are retrieved,

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

- **TP (True positive)**

If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP);

- **TN (True negative)**

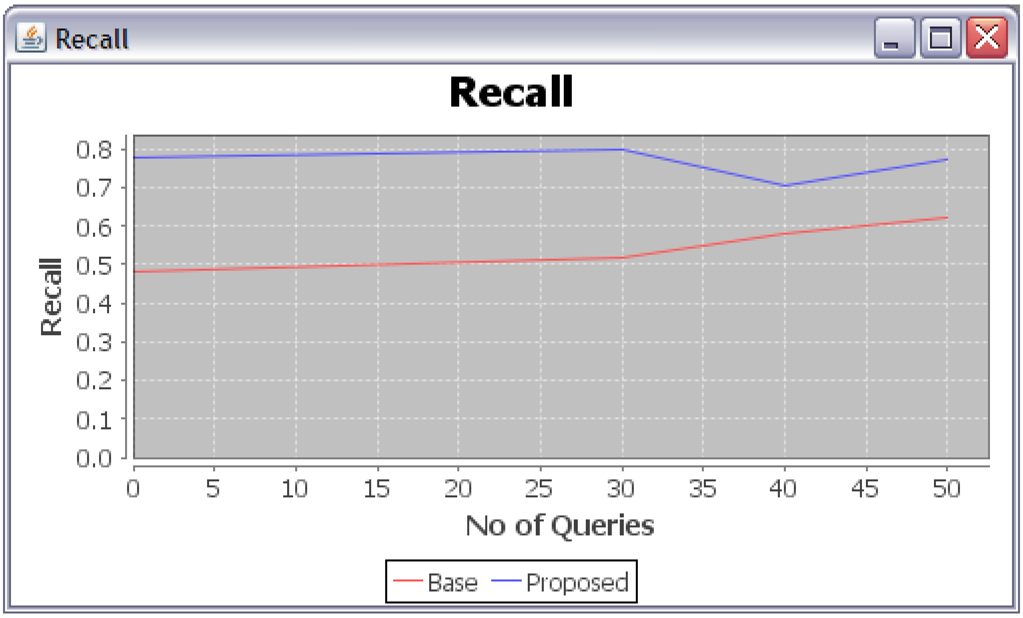
A true negative (TN) has occurred when both the prediction outcome and the actual value are n in the number of input data.

- **FP (False positive)**

If the outcome from a prediction is p and the actual value is n then it is said to be a false positive (FP).

- **FN (False negative)**

False negative (FN) is when the prediction outcome is n while the actual value is p.

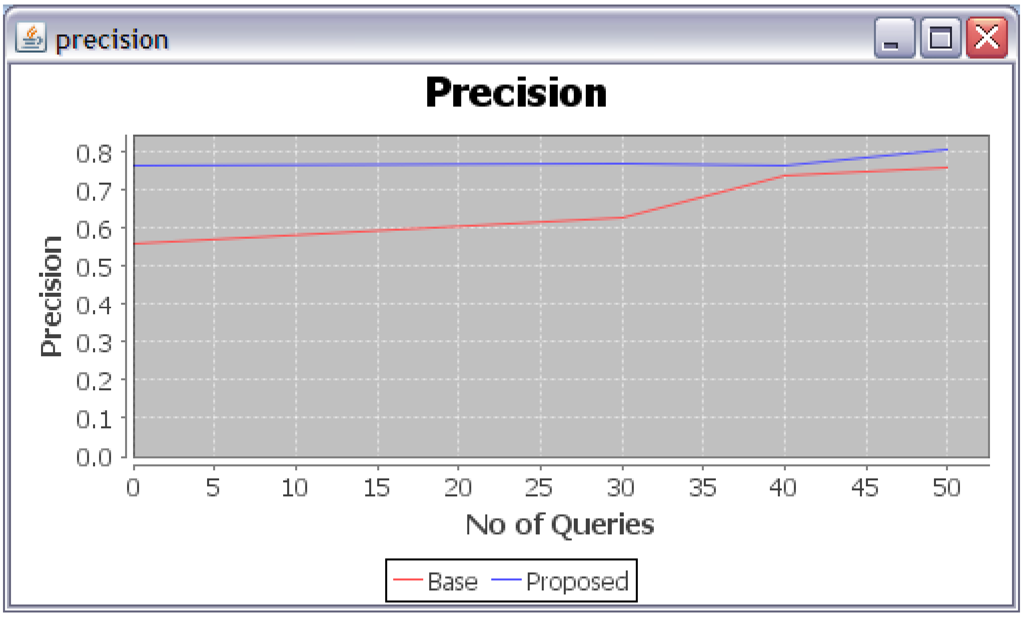


From this graph value we can say that our proposed scenario gives maximum recall values than our existing system. We use Improved Structured Robustness algorithm in proposed system to provide accurate results. It takes number of queries in x-axis and recall values in y-axis.

Precision

Precision value is calculated based on the retrieval of information at true positive prediction, false positive . Precision is calculated as the percentage of positive results returned that are relevant.

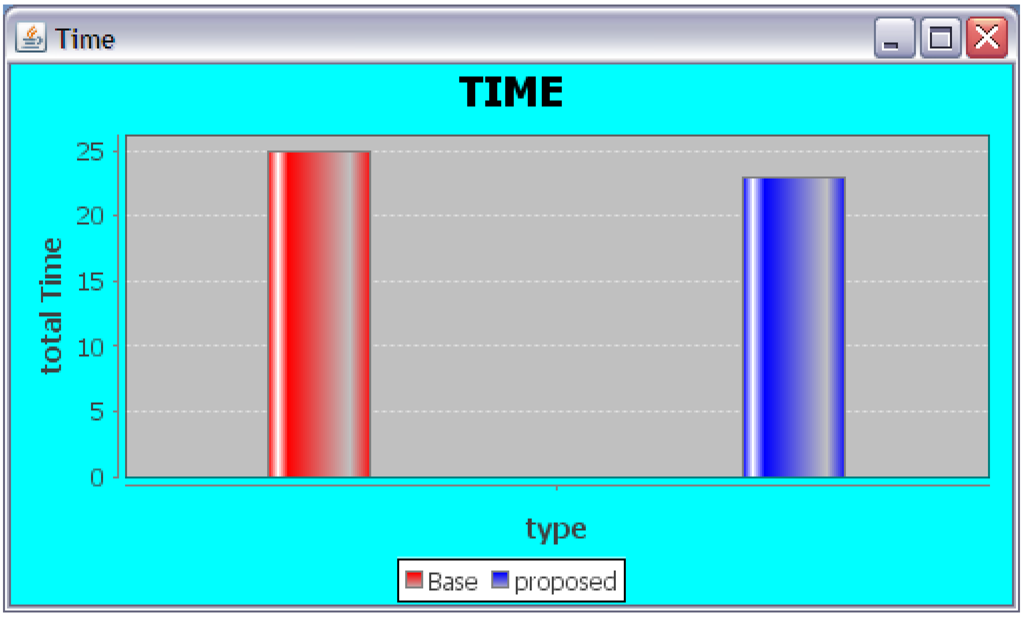
$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$



From this figure we can conclude that our proposed system produce more precise values than our existing system. It takes number of queries in x-axis and precision values in y-axis.

Time Complexity

We use ontology with Word Net tool to extract the answer set firstly and efficiently with relevant top k results. We consider the methodologies in x axis and total time factor in y axis. By using proposed I-SR with ontology concept it takes minimum amount of time for computation in proposed system



VI. CONCLUSION AND FUTURE WORK

A framework has been developed by employing algorithms to measure the degree of the difficulty of a query over a database, using the ranking robustness principle and ontology based Word Net tool. Based on our framework, the algorithm employing the structure robustness score calculation, spearman's correlation .the approximation algorithms and ontology based Word Net tool would efficiently predict the effectiveness of a keyword query.

From the experimental results, we can say that the proposed system is more effective than the existing system in terms of accuracy rate, quality of result and short threshold time. Proposed system ensures significant impacts in reducing the error rate and incurred time overhead involved in the prediction process. The main complexity involved when compared with unstructured databases was the semantic relations existed within the databases among different schema components.

Many relational databases contain text columns in addition to numeric and categorical columns. It would be motivating to observe whether correlations between text and non-text data can be expected in a meaningful way for ranking. Finally, comprehensive quality benchmarks for database ranking want to be established. This would provide future researchers with a more combined and efficient basis for evaluating their retrieval algorithms.

VII. ACKNOWLEDGMENT

An exertion over a long period may be successful only with advice and guidance of many well-wishers. We take this opportunity to express our gratitude to all who encouraged us to complete this work. We would like to express our deep sense of gratitude to our respected Principal for his inspiration and for creating an atmosphere in the college to do the work. We would also like to thank Head of the department, Computer Science and Engineering.

REFERENCES

[1] A S. Cheng, A. Termehchy, and V. Hristidis, “Efficient prediction of difficult keyword queries over databases,” Knowledge and Data Engineering, IEEE Transactions on, vol. 26, no. 6, pp. 1507–1520, 2014.

[2] R. Elmasri, Fundamentals of database systems. Pearson Education India, 2007, vol. 2.

[3] George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, “Keyword searching and browsing in databases using banks,” in Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE, 2002, pp. 431–440.

[5] V. Ganti, Y. He, and D. Xin, “Keyword++: A framework to improve keyword search over entity databases,” Proceedings of the VLDB Endowment, vol. 3, no. 1-2, pp. 711–722, 2010.

[6] V. Hristidis, L. Gravano, and Y. Papakonstantinou, “Efficient ir-style keyword search over relational databases,” in Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003, pp. 850–861.

[7] Y. Zhou and W. B. Croft, “Ranking robustness: a novel framework to predict query performance,” in Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 2006, pp. 567–574.

[8] V. Jain and S. Prasad, “Ontology based information retrieval model in semantic web: A review,” International Journal, vol. 4, no. 8, 2014.

[9] J. Alvez, J. Atserias, J. Carrera, S. Climent, E. Laparra, A. Oliver, and G. Rigau, “Complete and consistent annotation of wordnet using the top concept ontology.” in LREC, 2008.