IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CONTEXT BASED SEMANTIC SEARCH DIVERSIFICATION MODEL

[1]Sijin P, [2]Champa H N

[1, 2] Department of Computer Science and Engineering,
[1, 2] University Visvesvaraya College of Engineering, Bangalore, India

Abstract: In keyword search it is important to identify the actual meaning of a query and the context of query terms at an earlier phase. Conceptualization is the process of identifying this contextual information about query terms and to produce meaningful segments which guides to achieve search diversification. Because of the advancements in Machine Learning (ML), Artificial Intelligence (AI) and the growth in overall Parallel computing applications the search diversification process is not at all a risky job now. The proposed Semantic Search Diversification Model (SSDM) identifies the context of the search query at the very beginning of the search. The concept of Smallest Lowest Common Ancestor (SLCA) nodes is used to lock the desired key and semantic keys over the XML tree branches. The Mutual Information value of the given query terms and their semi features are considered. The Probabilistic weight of the search query is calculated based on the relevancy and the novelty of the query. The proposed Anchor based Semantic Preservation (AESP) algorithm extracts the weighted SLCA nodes for the original and generated queries from the given XML data and determines the anchor nodes to initiate the tree pruning. The node lists are assigned to multiple processors in order to achieve parallelism. The fuzzy set of the popular terms in the given data set is used for creating semantic table for the query terms. The Normalized Discounted Cumulative Gain (nDCG) measures of the SSDM shows the usefulness of the query suggestions, and a point to apply search annotations if needed.

Index Terms **- Anchor node, Mutual information, Intention, Context, Diversification.**

I. INTRODUCTION

Usually search diversification processes incrementally compute the top-k qualified search results by balancing the relevance and novelty of the query results. In Information Retrieval(IR) approach of keyword search, the query intention can identify by various methods such as automated methods like Named Entity Recognizer (NER) [11], [15] Part-of-Speech tagging (POS), Conceptualization of Typed Terms of Query (CTTQ) [10], [19] and probabilistic approach like topic modeling, other advanced methods like conceptualization, Fuzzy Conceptual Model (FCM), Annotation, User interaction etc. The Database (DB) approaches bring value to IR style Query by specifying some contexts for the search to proceed. XML data representation can be used with all the above approaches in which XML documents are represented as nodes in labeled tree T for the given query Q = q1, q2,...qn where q1, q2,...qn are query terms.

The SLCA nodes of the original query keywords are extracted first from XML data using XQuery commands. The Mutual Information values (MI) of feature terms with the given keywords show the role of partial features in query diversification for generating relevant queries along with original queries. Top-k results can be filtered out for further processing. Fig. 1. shows the portions of DBLP XML tree for the query q="Binary Tree". The SSDM used Mutual information theory for extracting the query results, for example for the query "Binary Tree", a correlation table (for the extracted features and the query terms) is produced to maintain all the terms, their partial matches and their correlation values as shown in Table 1.

The effectiveness of a diversification system such as SSDM is measured in terms of Normalized Discounted Cumulative Gain (n-DCG). The n-DCG listed out queries should be normalized across queries. This is achieved by producing maximum possible DCG. The ratio of DCGp to IDCGp is called n-DCG. The DCGp calculation is given in (1). IDCGp is the DCGp through a position p.

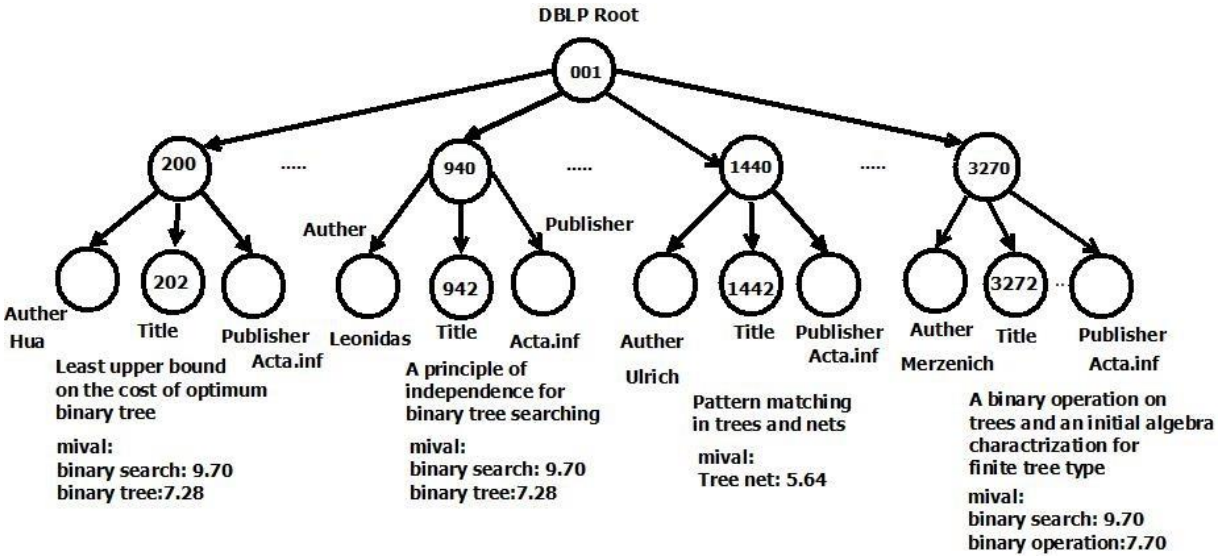

Figure 1: DBLP tree portions for query "Binary Tree".

$$DCG_{\mathrm{p}} = \sum_{\mathrm{i=1\ to\ n}} \frac{2^{\mathrm{rel_i}} - 1}{log(i+1)} \qquad (1)$$

The fuzzy set of popular query terms is created for mapping the query terms with semantically related alternatives from the given Tree. This approach will produce an additional set of queries which are similar to the original set of queries and is used for producing query suggestions.

The MI based SLCA node generation for query and it's semi features produces a huge set of SLCA nodes. The typical anchor based pruning algorithms are not enough to prune out all unwanted sub trees. The proposed semantic search diversification system, the SSDM reduced the response time considerably by using an anchor based semantic preservation algorithm. The AESP algorithm distributes the query and generated query's SLCA nodes to dedicated processors in round robin order. A workload based semantic table is maintained for candidate query terms in particular domains.

II. LITERATURE SURVEY

Keyword search is an essential service offered by Machine Learning (ML). ML is a method of data analysis, which enables a system to learn from data using algorithms, artificial intelligence tools, learned prototypes and efficient data representation methods [16], [26]. The ML evaluated data is used in the Industrial line to produce products and parts having high throughput, low cost and more reliability [14], [17], [18], [20]. Data mining is the process of formulating rules and associations to localize the desired pattern by applying ML and various data models [22], [23], [25] with data summarization methods [5], [8], [28]. In the SSDM design process the above methods have considerable role in the search query manipulation and diversification.

The works in [13] is also proposed a query relaxation framework in order to capture the approximate answers for a query over XML data by applying descendant clue. The Path Similarity (PS) algorithm is used to prepare the path algebras for the queries. The Query Approximation (QA) algorithm is used to produce the approximate query values. This approach produced an exclusive attribute set to the user which are measured over the data tree for a query, and could be used to effectively diversify the search in the context of attributes. The methods in [12] are also proposed a similar kind of work to measure the relevancy of a query. It pinpointed the context of the search query by the mutual evaluation of the relevance and novelty of the query. The SSDM achieves QA on the basis of MI values of query segments and it's semi-features.

The works in [3] says, the context or semantic information can easily extract by segmenting the query in batch mode. The paper considered the global and local context of a query for evaluating it. The global context represented the query as a globally recognized form such as English phrase. Local contexts of a query are represented with local linguistic features and term dependency. The pseudo feedback approach is used to identify confident segments. The SSDM is incorporated with a Fuzzy Set (FS) of query terms in order to impart semantic matching.

Even though the proper context of a term in a search query can identify thorough concept labeling, that follows pairwise modeling of data and has considerably more computation time [6], [7], [27]. XML data modeling can be used to improve this cross computation problem. The context of the search can be determined by listing out all the nodes which contain the given query words and usually organized with an inverted list of nodes represented with Dewey label. These binary approaches are further

improved by applying programming methods like Lowest Common Ancestor (LCA)/ Smallest Lowest Common Ancestor(SLCA) / Exclusive Lowest Common Ancestor (ELCA) semantics so as to obtain the minimal number of meaningful results [21], [24]. This paper used SLCA approach for node list formation.

Deciding the context of search at an earlier stage is important in keyword search applications [4]. Even though the proper context of a term in a search query can identify only by thorough knowledge about the term in text segmentation, part-of-speech tagging, and concept labeling methods [9], if data are localized on a connected metadata model such as tree or graphs it is recommended to use methods like region encoding.

III. METHODOLOGY

Given a keyword query q and a dataset DT, feature analysis of query terms is achieved through MI(x,y) analysis of q over DT and it's trained FS where x and y are respective query terms. The semantic interpretation of q is represented with a variable weight(qnew) based on relevance and novelty of the generated query.

$$weight(q_{\text{new}}) \propto \frac{\bigcap qft_{\text{i}} \in q_{\text{new}} R(qft_{\text{i}}, DT)}{R(DT)}$$

Where $\bigcap \frac{R(qft_{\text{i}},DT)}{R(DT)}$ is the SLCA nodes for queries with semi features and qft_{i} is the query with semi features.

3.1 Problem statement

For the given query q(q1, q2,....qn) and the given dataset DT calculated MI(q1, q2,....qn) values for all query terms and their partial matches. The query weight(qnew) is proportional to the number of SLCA nodes generated for the entire query set of original and generated queries.

3.2 Conceptualization with semantic search diversification

Identifying the contextual meaning of a search query from it's candidate terms is the turning point of a search diversification system development process. For a query "Herman's works in Springer", the LCA nodes obtained are DBLP root node and node labeled Article 2 (1554). The SLCA node(s) obtained are Article 2(1554). If sibling support is activated in SSDM such as title "Efficient Worst-Case Binary Data Structures for Range Searching" the node labeled Article 2 should be the right answer. It is shown in Fig. 2.

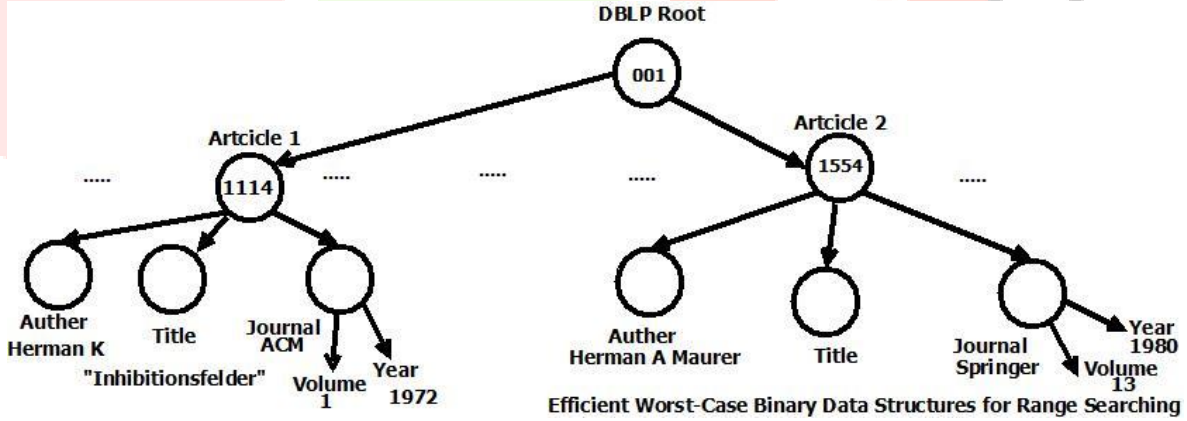

Figure 2: Sibling support in XML data tree.

The SLCA nodes of the original query keywords are extracted first from XML data using XQuery commands. The fuzzy set of popular query terms is created for mapping semantically similar terms in the given Tree. The feature extraction of the given query terms lists out the co-occurred terms for the query terms. The mutual information values of feature terms with the given keywords determine the contextual measure of the generated relevant queries and original queries. Top-k results can be filtered out for further processing.

3.3 Feature Extraction and Mutual Information

The Fig. 3. shows a typical semantic search diversification system based on MI. MI measures the mutual dependence between query terms and it's semi-features. The FS is a set of semantically related alternatives of contextual terms in the given corpus. It is usually prepared by applying FCM or MI of query terms over a word corpus. The SLCA nodes are identified and iterated for producing top-k results for the given query. A search query (e.g binary tree) usually contains keywords and acronyms, those are short in nature. The proposed method uses MI to quantify the discrepancy between the query keywords, and the associated words in the corpus to determine the context of the search [1]. A feature vector is used to hold the relevant features of the query

keywords and hence formed a reduced set of associated words. XQuery language is used to extract and manipulate features from the XML documents. The mutual information score is calculated according to (2) for the features and is listed in Table 2. These values reflect the mutual dependency between the feature vector and the query terms in their various contexts.

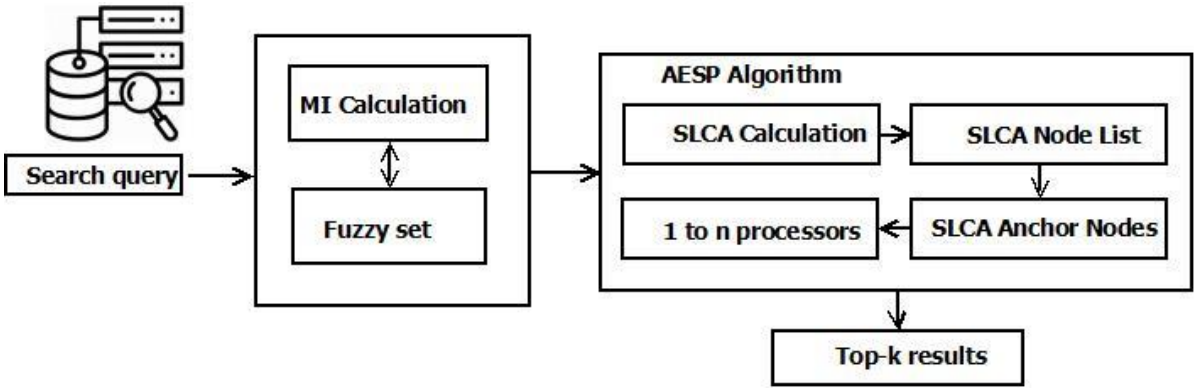

Figure 3: Semantic search diversification process.

Table 1: Feature extraction.

Keyword	Features
Binary	search, tree, testing, operation etc.
Tree	nets, search, Huffman, rational, Y-tree range,multi-way, parse, balanced, B+ tree, quad tree, problem, evaluation etc.

Where x and y are two discrete random variables. P(x,y) is the joint probability mass function of x and y and P(x) and P(y) are the marginal probability mass functions of x and y respectively. The MI has a bound of (-1) <= *mi*(*x; y*) <= *min*(*-logP* (*x*), *-logP* (*y*)).

$$MI(x,y) = P(x,y) + log\frac{P(x,y)}{P(x).P(y)} \qquad (2)$$

The MI valued features are listed in Table 2. it is notable that the term 'data structure'is not extracted by XQuery directly, it is a semantically related term of an ADT concept 'tree' and is mapped with the semantic table for tree and is returned by XQuery by a query command article=title[contains(;*DataStructure*)]=*text*().

3.4 Search diversification

The feature extraction process generates more keywords which are having a tendency to divert the query if they are on the same region of a sub tree because they collectively attributed some common concepts. The newly generated queries can be modeled by an aggregated scoring function as given in (3).

$$weight(q_{\text{new}}) = P(\frac{q_{\text{new}}}{q}, DT).NDIF(q_{\text{new}}, Q, DT) \qquad (3)$$

Where P(qnew/q;DT) is the probability of generating a new semantically related query for the given query q. NDIF(qnew,Q,DT) is the measure of difference between newly generated queries from previous queries.

Table 2: Mutual information values for various context terms w.r.t query terms.

Keyword	Features	MI value	Keyword	Features	MI value
Binary	search	9.28	Tree	Huffman	7.97
Binary	testing	7.97	Tree	search	7.64
Binary	tree	7.28	Tree	Y-Tree	7.97
Binary	data structure	8.7	Tree	tree net	5.64
Binary	operation	7.7	-	-	-

3.4.1 Evaluating the semantic similarity of the query terms w.r.t the application domain

Words with same meaning have similar representation is the working principle of Word embedding methods Mikolov et al. (2013), Kusner et al. (2015). In order to achieve such a representation the first part of the (3) is modeled as in (4). As a part of experimental analysis some important and popular index terms from DBLP dataset are selected and prepared semantically related terms for each index terms. Using this data, a semantic table is created for each index terms. For example for a query term "tree" the semantically related terms identified are data structure, depth, acyclic graph, root, leaf nodes. In order to reduce the complexity of prototype design this paper considers only data from DBLP domain.

$$P(\frac{q_{\text{new}}}{q}, DT) = \frac{P(\frac{q}{q_{\text{new}}}, DT) * P(\frac{q_{\text{new}}}{DT})}{P(\frac{q}{DT})} \tag{4}$$

where *P* (*q*/*q*new, *DT*) is the probability of getting query q as a coherent term for generated query qnew in the given XML data tree DT. *P* (*q*new/DT) is the query generation probability of the system. *P* (q/*DT*) is the probability of the system to search on the given query. It is possible to rewritten *P* (*q*/*q*new, *DT*) as in (5).

$$P(\frac{q}{q_{\text{new}}}, DT) = \pi_{q_i \in q, ft_{ij} \in q_{\text{new}}} P(\frac{q_i}{ft_{ij}}, DT) \tag{5}$$

Where *P* (q_i/ft_{ij},*DT*) is defined as the probability of query term correlated with the given feature and is represented as qft_i for example q='Binary' feature term is 'search'. It is interpreted as following and given in (6).

$$P(\frac{q_i}{ft_{ij}}, DT) = \frac{P(\frac{ft_{ij}}{q_i}, DT) * P(\frac{q_i}{DT})}{P(\frac{ft_{ij}}{DT})}$$

$$P(\frac{q_i}{ft_{ij}}, DT) = \frac{R(\frac{q_i, ft_{ij}, DT}{R(DT)})}{R(\frac{ft_{ij}, q_i, DT}{R(DT)})}$$

$$P(\frac{q_i}{ft_{ij}}, DT) = \frac{R(qft_i, DT)}{R(ft_{ij}, DT)} \tag{6}$$

The query generation probability *P* (*q*new/DT) can be calculated and given in (7)

$$P(\frac{q_{\text{new}}}{DT}) = \frac{R(q_{\text{new}}, DT)}{R(DT)} = \frac{\bigcap qft_i \in q_{\text{new}} R(qft_i, DT)}{R(DT)} \tag{7}$$

where $\bigcap qft_i \in q_{\text{new}} R(qft_i, DT)$ is the SLCA values of query term with it's features. From the above knowledge the (4) can be rewritten as in (8) where *1/P(q/DT)* has a value between (0*;* 1] and is denoted as *β*.

$$P(\frac{q_{\text{new}}}{q}, DT) = \beta * \frac{R(qft_i, DT)}{R(ft_{ij})} * \frac{\bigcap qft_i \in q_{\text{new}} R(qft_i, DT)}{R(DT)} \tag{8}$$

3.4.2 Evaluating the novelty measures of the generated queries w.r.t the search query

Novelty is the process of incrementally refine the diversified results into more specific one and is given in (9).

$$Novelty(q_{\text{new}}, Q, DT) = \frac{\{nx | nx \in R(q_{\text{new}}, DT) \bigwedge \nexists ny \in \{\bigcap q' \in QR(Q', DT)\} \bigwedge nx \leq ny\}}{R(q_{\text{new}}, DT) \cup \{\bigcap q' \in QR(Q', DT)\}} \quad (9)$$

Where $R(q\text{new}, DT)$ is the SLCA values generated by qnew. $\bigcap q' \in QR(Q', DT)$ is the SLCA results generated by queries in Q. $nx \leq ny$ gives vertex nx is a duplicate or ancestor of vertex ny. The denominator is the union of SLCA results generated by query in Q and qnew. In terms of relevance and novelty (3) can be rewritten as shown in (10).

$$weight(q_{\text{new}}) = \beta * \frac{R(qft_{\text{i}}, DT)}{R(ft_{\text{ij}})} * \frac{\bigcap qft_{\text{i}} \in q_{\text{new}} R(qft_{\text{i}}, DT)}{R(DT)} * \frac{\{nx | nx \in R(q_{\text{new}}, XTD) \bigwedge \nexists ny \in \{\bigcap q' \in QR(Q', DT)\} \bigwedge nx \leq ny\}}{R(q_{\text{new}}, DT) \cup \{\bigcap q' \in QR(Q', DT)\}} \quad (10)$$

$$= \frac{\beta}{R(DT)} * \frac{R(qft_{\text{i}}, DT)}{R(ft_{\text{ij}})} * \frac{\bigcap qft_{\text{i}} \in q_{\text{new}} R(qft_{\text{i}}, DT)}{} * \frac{\{nx | nx \in R(q_{\text{new}}, XTD) \bigwedge \nexists ny \in \{\bigcap q' \in QR(Q', DT)\} \bigwedge nx \leq ny\}}{R(q_{\text{new}}, DT) \cup \{\bigcap q' \in QR(Q', DT)\}}$$

$$= \frac{R(qft_{\text{i}}, DT)}{R(ft_{\text{ij}})} * \frac{\bigcap qft_{\text{i}} \in q_{\text{new}} R(qft_{\text{i}}, DT)}{} * \frac{\{nx | nx \in R(q_{\text{new}}, XTD) \bigwedge \nexists ny \in \{\bigcap q' \in QR(Q', DT)\} \bigwedge nx \leq ny\}}{R(q_{\text{new}}, DT) \cup \{\bigcap q' \in QR(Q', DT)\}}$$

IV. SEMANTIC SEARCH DIVERSIFICATION SYSTEM

4.1 Usability of anchor nodes in Tree

Anchor nodes have important roles in search diversification systems. If XQuery commands on platform such as BaseX is used for SLCA node extraction a list of nodes can be generated with regional information such as position, sibling nodes details, node levels. Using existing programming approaches the same results can be obtained with a little computation and traversing time on large XML data tree. Fig. 4. represents the positioning of anchor nodes and it's region based traversal to lock the specified keys including semantic keys. Let k1, k2 are the query terms and ft1, ft2 are features of q(k1,k2). Suppose on a particular context if the query k1, k2, ft1 is evaluated the node v11 be the first anchor nodes to partition the region in to four search spaces called Rpre, Rnext, Ranc, and Rdes in which Rpre is the set of previous or left of nodes of anchor node identified. Ranc is the set of ancestor or top level nodes of anchor node identified. Rdes is the set of descendant or bottom level nodes of anchor node identified. Rnext is the set of next of or right level nodes of anchor node identified. The identified Rpre nodes are v2, v3, v4, v5, v6, v7, v8, v9 and their post order numbers should be less than encoding range of v11. Rnext nodes are v15, v16, v17, v18, v19, v20 and their pre order numbers should be greater than upper encoding number of v11. Rpre nodes are v1, v10 and their pre order numbers should be less than lower encoding number of v11 and post order numbers should be greater than upper encoding number of anchor node v11. The rest nodes are Rdes v12, v13, v14. v10 also has ft1 but v11 is the anchor node due to exclusive property of SLCA semantics. Rdes set do not have ft1 so these nodes can skip. Rpre set has ft2 therefore the node v5 can set as SLCA node similarly Rnext set has ft2 therefore the node v19 can set as SLCA node.

4.2 Semantic search diversification algorithm

The semantic search diversification is achieved through Anchor based parallel sharing algorithm. The algorithm assign each processor a region for SLCA computation. The new queries qnew and their corresponding semi features are maintained in a vector V. When the new query is processed, the information about shared segments and the necessary communication details are maintained among processors. If the new query candidate contains shared parts (features) called ψ in V, for each shared part a processing status is set once it is processed. In another query search the obtained information can be passed to the required processor. The anchor pruning process avoids lot of nodes without computation and parallelism reduced computation time.

Algorithm 1: Anchor based parallel sharing semantic preservation algorithm-AESP

Data: A search query q with n keywords it's Fuzzy set FS and an XML data tree DT.

Result: Top-k query intentions Q_{top} and overall result set ϕ.

initialization;

1. FT_{mxn} =FeatureAnalysis(q,FS, DT);
2. while q_{new}=GenerateNewQuery(FT_{mxn}) $\neq$ null do
3. ϕ=null and P_qft=1;
4. $qins_{i,j}$= SLCANodeList(qft, TD) for $qft_{i,j} \in q_{new} \wedge 1 \leq i \leq m \wedge 1 \leq j \leq n$;
5. P-qft= π $ft_{ij} \in qft_{ij} \in q_{new} \frac{qins}{SLCANodeList(qft,XTD)}$;
6. If ϕ! = empty then
7. For all $node_{anchor} \in \phi$ do
8. Perform Partition($qins_{i,j}$,$node_{anchor}$) to $qins_{pre}$, $qins_{dec}$, $qins_{next}$
9. if $\forall$ $qins_{pre}$! =null then
10. $\phi^{'}$= SLCACalc($qins_{pre}$,$node_{anchor}$);
11. if $\forall$ $qins_{des}$! =null then
12. $\phi^{''}$= SLCACalc($qins_{des}$,$node_{anchor}$);
13. $\phi=\phi+\phi^{'}+\phi^{''}$;
14. if $\phi^{''}$!= null
15. ϕ.remove($node_{anchor}$);
16. if $\exists$ $qins_{next}$= null then
17. Break the for loop;
18. qins=$qins_{next}$ for $1 \leq i \leq m \wedge 1 \leq j \leq n$;
19. else
20. SLCACalc($qins_{ij}$);
21. weight(q_{new})=$P\_q\_new * \phi * \frac{\phi}{\phi+\phi'}$;
22. if $|Q| \leq$ k then
23. put q_{new}:weight(q_{new}) into Q;
24. put q_{new}:ϕ into ϕ;
25. else if weight(q_{new})>weight($q_{new'}$)
26. replace $q_{new'}$:weight($q_{new'}$) and q_{new}:weight(q_{new});
27. ϕ.remove($q_{new'}$);
28. for each q_{new^i} generate shared segment ψ
29. for pid =1 to n
30. LookupTable=LookUP(ψ);
31. set qft=$q^{'}ft^{'} \oplus$LookUP(ψ);
32. go to step 2
33. Return Q and ϕ;

It is possible to see from algorithm that the FeatureAnalysis() function formulates a 2D feature table consist of the query terms and their fuzzy counter parts. The features for all these items are identified and stored in feature table FTij. GenerateNewQuery function is invoked to obtain new queries qnew. The SLCA nodes are identified for each query term with its corresponding semi features (Binary Tree). P qft measures the probability of query instances over the selected SLCA nodes. SLCA nodes are identified for the threshold semi features (binary search, xml tree etc.). Anchor nodes nodeanchor are identified for each SLCA listed group. A tree partition function called Partition(qinsi,j,nodeanchor) is used to perform partition on various directions called qinspre, qinsdec, qinsnext. The SLCACalc(qinspre,nodeanchor) function is used to calculate SLCA nodes for ancestor region of anchor. The SLCACalc(qinsdes,nodeanchor) is used to calculate SLCA nodes for descendants. The lookup table LookUP( ) is used to hold partial matches where is the partial match. SLCA nodes are identified for partial matches and is added to φ. All the SLCA nodes are assigned to various processors with pid 1 to n in round. For each shared part a processing status is set once it is processed. In an another query search the obtained information can be passed to the required processor.

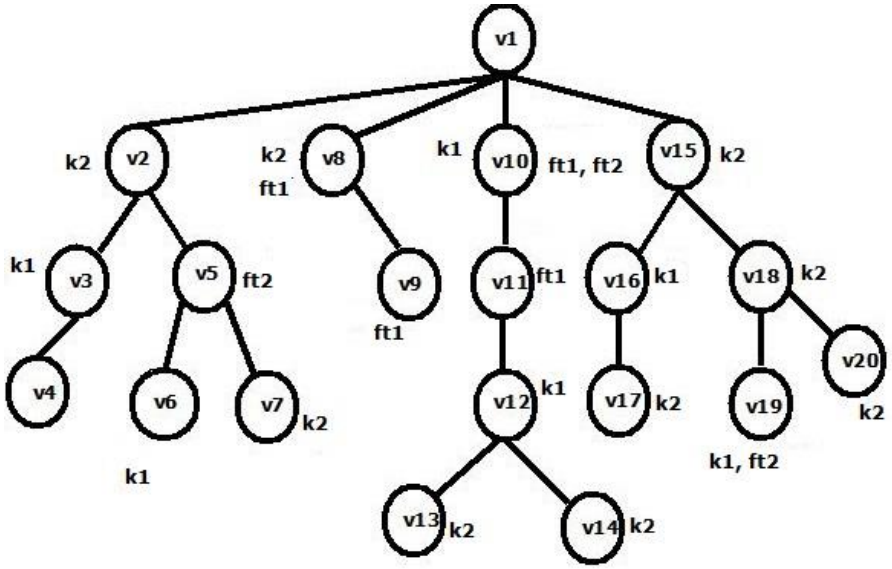

Figure 4: Anchor nodes on XML data tree.

V. PERFORMANCE EVALUATION

The proposed system runs on a computer system with processor Intel(R) Pentium(R) CPU with processing speed of 2.00 GHz and installed memory of 2 GB. All the program modules are developed in J2EE platform and deployed with windows compatible XAMPP server. XQuery programs run over BaseX is also used for fast SLCA node identification and other query operations over DBLP XML data tree during the processing of massive datasets. BaseX is a light-weight XML database management system and XQuery processor. It is specialized in querying, storing and visualizing large XML documents and collections. It serves as an excellent framework for building complex data-intensive web applications as shown in Fig. 5.

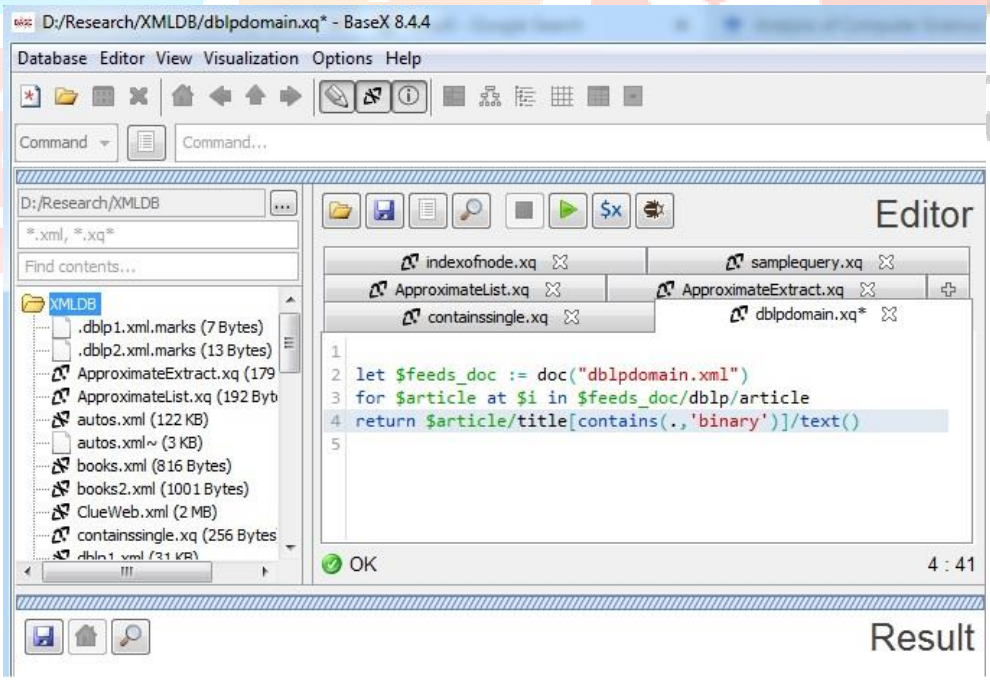

Figure 5: A typical BaseX window.

5.1 Information about dataset

The SSDM used DBLP dataset for it's experimental analysis. DBLP is a collection of open bibliographic information on major computer science journals and proceedings and is available in XML batch file. The database is publicly available at http://dblp.uni-trier.de/xml/ and contains more than 50000 data objects Biryukov and Dong (2010) and around 5634053 publication details.

5.2 Efficiency of search diversification algorithms based on query suggestions

SSDM shows a considerable reduction in response time because lot of nodes are skipped without any computations and used the least number of query suggestions to determine the context of the search. The proposed anchor based pruning with semantic preservation algorithm induced reduction in response time in average because the proposed search diversification model based on query suggestions is captured the user intentions at an earlier stage. The conceptualized mapping of keywords to their relevant fuzzy tables reduced the overall computations. Table 3 contains ten queries which are able to fetch answers from DBLP.

Average response time for calculating the SLCA results for a query from the give query suggestions using ASPE with XRank [7], [12] and proposed AESP is shown in Fig. 6a. The experimental results show that AESP shows considerably less response time i.e 20 percent time of baseline methods with five suggestions, 33 percent time of baseline methods with ten suggestions, 38 percent time of baseline methods with fifteen suggestions, 34 percent time of ASPE with twenty suggestions. When the query suggestions are more , there should be more SLCA computations with these suggestions and response time increases accordingly. In the case of short and vague queries, SSDM has to process comparatively less features and the SLCA nodes wants to process are further reduced. Which in turn shows considerable reduction in response time with less number of query suggestions in the case of AESP as shown in Fig. 6b. In the CTTQ evaluated query over the FCM trained corpus the response time is still more reduced.

Table 3: Ten frequent queries on DBLP domain.

No.	Query	Co-occurring terms
1	Binary search applications	Binary, search
2	Database query programming	database, query
3	Semantic network representation	semantic, network
4	Programming domain	software
5	Binary tree concepts	Binary, tree
6	Parallel programming	Parallel architecture
7	Dynamic programming	Dynamic programming
8	Binary data structure	data structure
9	Trees and nets data structure	tree
10	Pattern matching	image processing

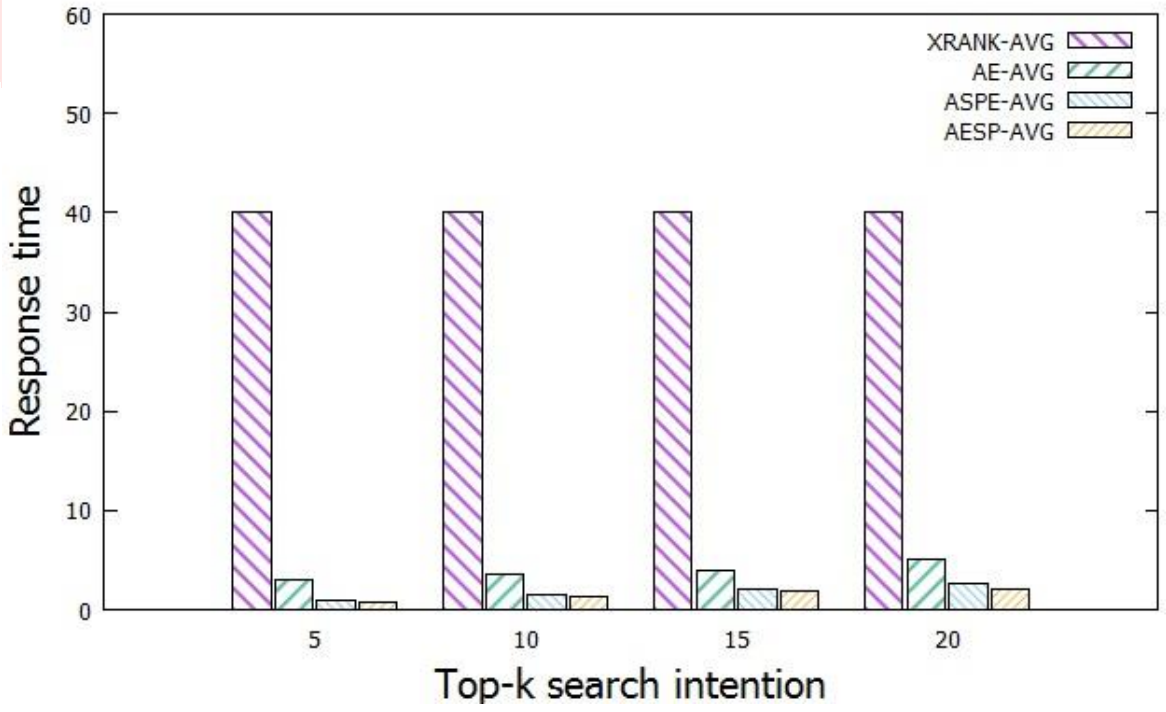

Figure 6a: Average response time with qualified query suggestions.

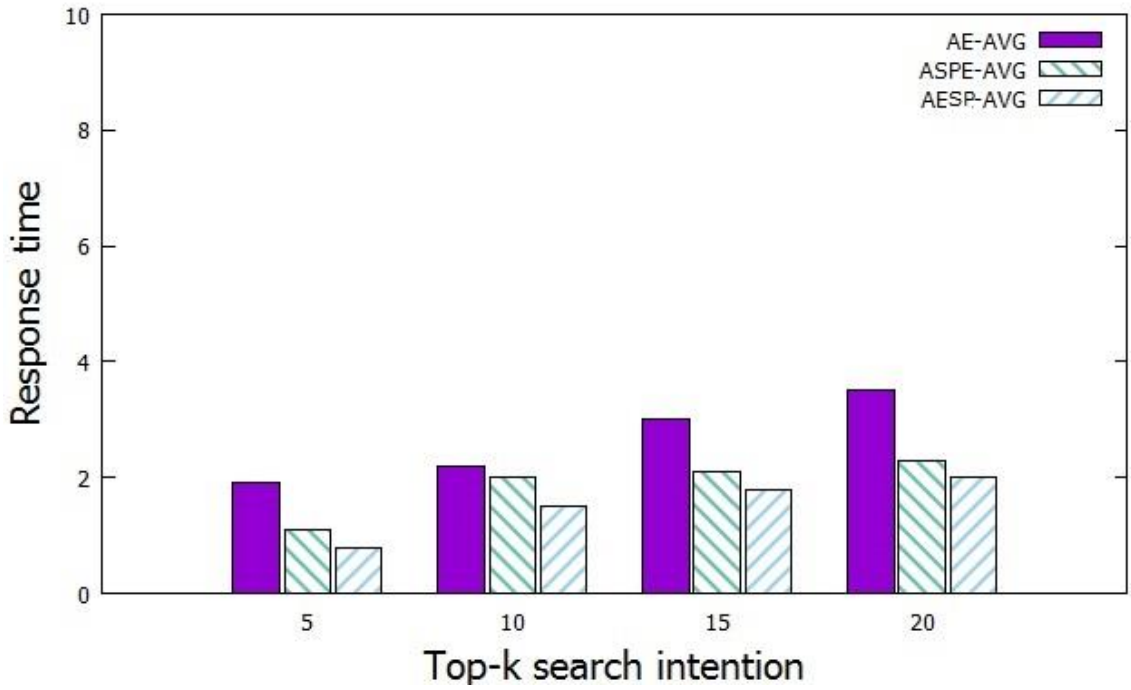

Figure 6b: Average response time with qualified query suggestions for short and vague queries.

5.3 nDCG Measures of Relevancy of a Search Framework

Normalized Discounted Cumulative Gain (nDCG)is used for measuring the usefulness of web search algorithms. It used logarithmic reduction method to list out the highly relevant documents to the top. A case study is considered in which some queries into the domain of DBLP dataset are considered and two groups of students group A and group B were asked to work on Google and Bing search engines to obtain diversified results over DBLP dataset for the selected queries. They analyzed thousands of search results in order to produce relevant queries and suitable query suggestions. Some useful query suggestions are prepared and given them a value between[0-10]. The top-n query suggestions are considered for DCG computations as given in (1) for the queries over DBLP domain. Fig. 7a. shows the query diversification results for query q5 and query q6 in which n-DCG values are not less than 0.8 when k≥2. It shows the proposed frame work has 80 percentage chance to approach the ground truth of ideal diversified query suggestion. Fig. 7b. shows the query diversification results for query q1 and query q2 for which SSDM has 84 percentage chance to approach the ground truth of ideal diversified query suggestions. In the case of q9 both google and bing shows depreciation for n-DCG values, but in the case of q8 it is normal, Fig. 7c. shows this. In such cases FCM can incorporate to increase the quality of search and more suggestions can apply.

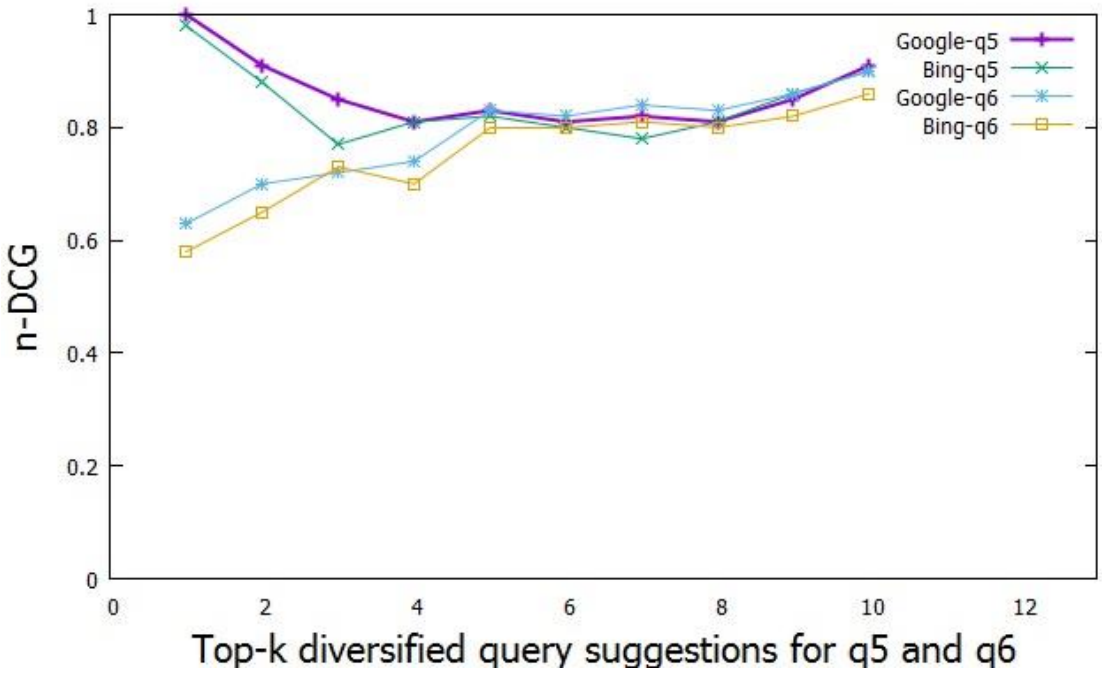

Figure 7a: n-DCG measure of Top-k query suggestions for querys q5 and q6 over Google and Bing search engines.

Only SSDM can answer the queries like q8 well, because for the given query "binary search" the term data structure is a feature with strong semantic correlation more than MI evaluated value, SSDM can intrinsically use it at the time of feature extraction by incorporating FCM, and can proceed for SLCA computation to boost up AESP evaluation, it is shown in Fig. 7d.

5.4 Measure of Diversification of queries

In order to measure the diversification ratio of queries, ten queries from Table 3 is considered over the DBLP dataset. The structural queries produced by DivQ can't exactly reflect the differentiation of the results. DivQ showed more than 75 percent differentiation ratio for query set 6 and 10. These queries contain specific keywords and the structural associations over the graph are differs in representations.

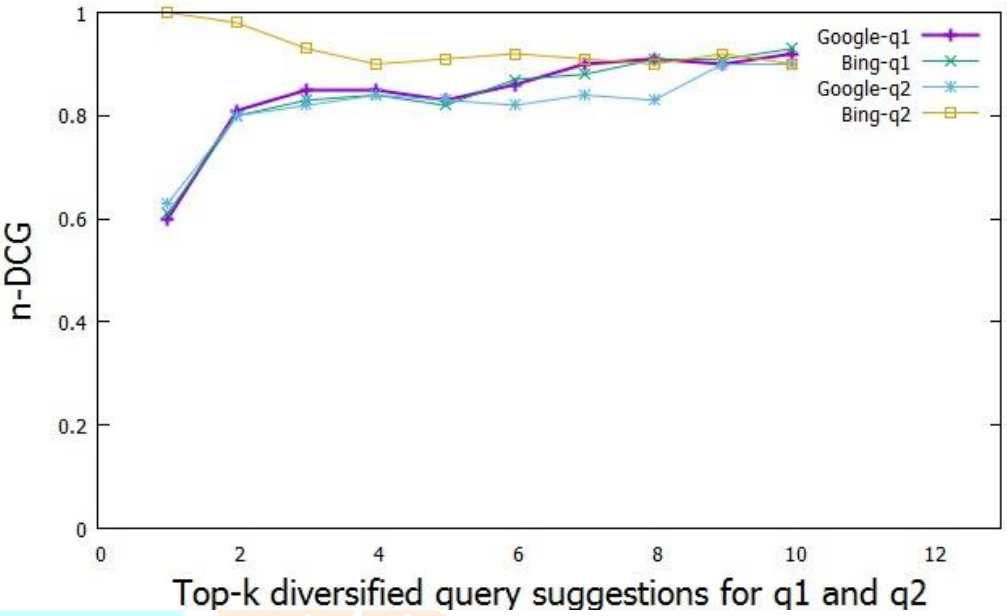

Figure 7b: n-DCG measure of Top-k query suggestions for querys q1 and q2 over Google and Bing search engines.

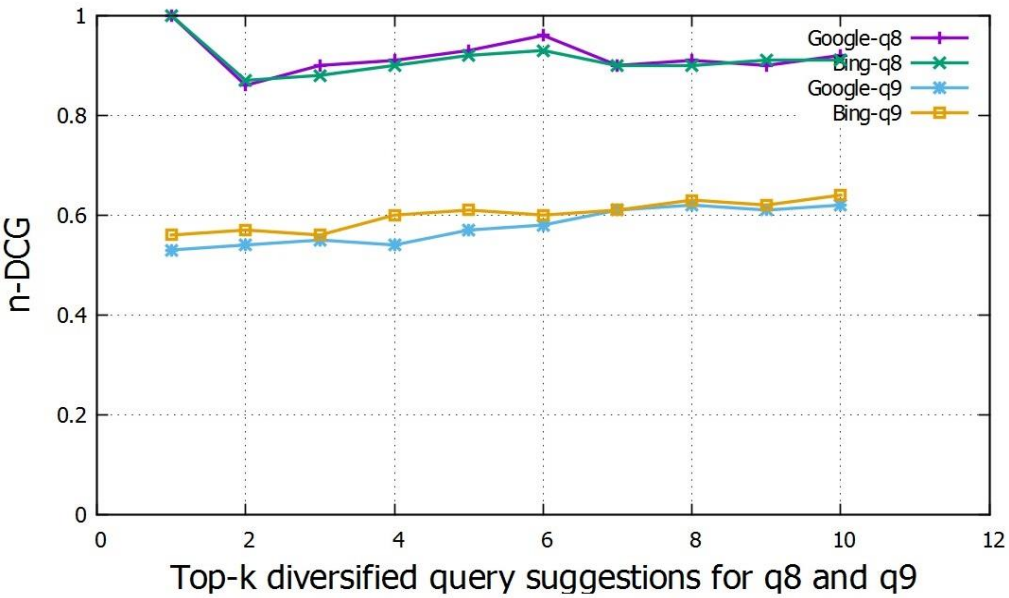

Figure 7c: n-DCG measure of Top-k query suggestions for querys q8 and q9 over Google and Bing search engines.

DivContest considers structure of a query over a graph and it also considers the word context of a query and hence able to produce more differed results [12]. The SSDM considers word context, structures, occurrence count for the words, and the semantic table inorder to produce more diversified results and is shown in Fig. 8.

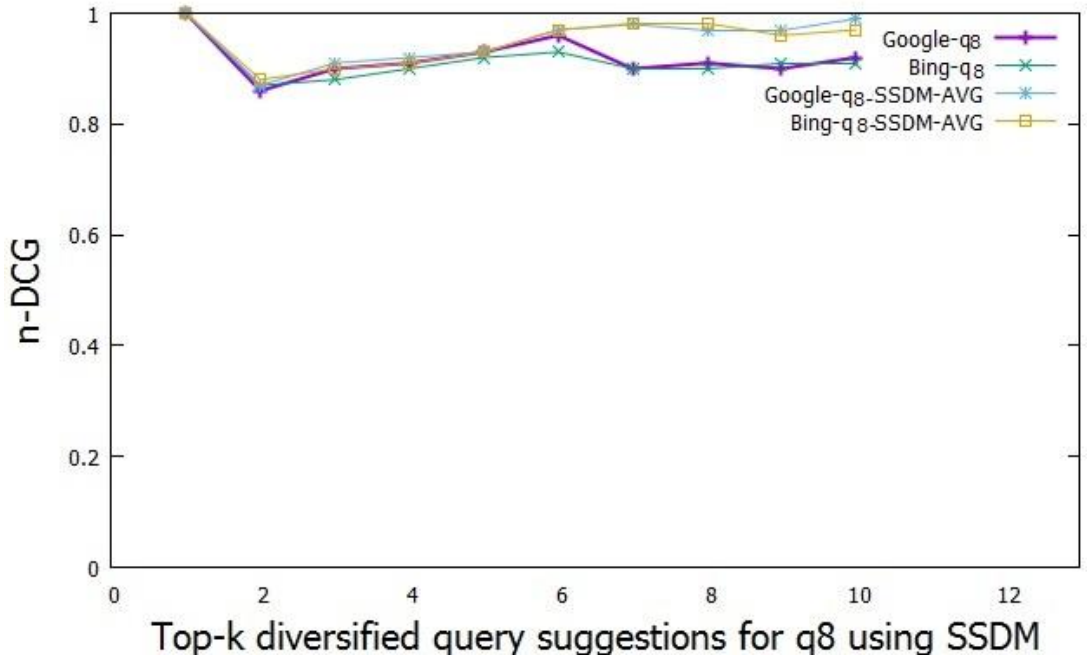

Figure 7d: n-DCG measure of Top-k query suggestions for query q8 with SSDM.

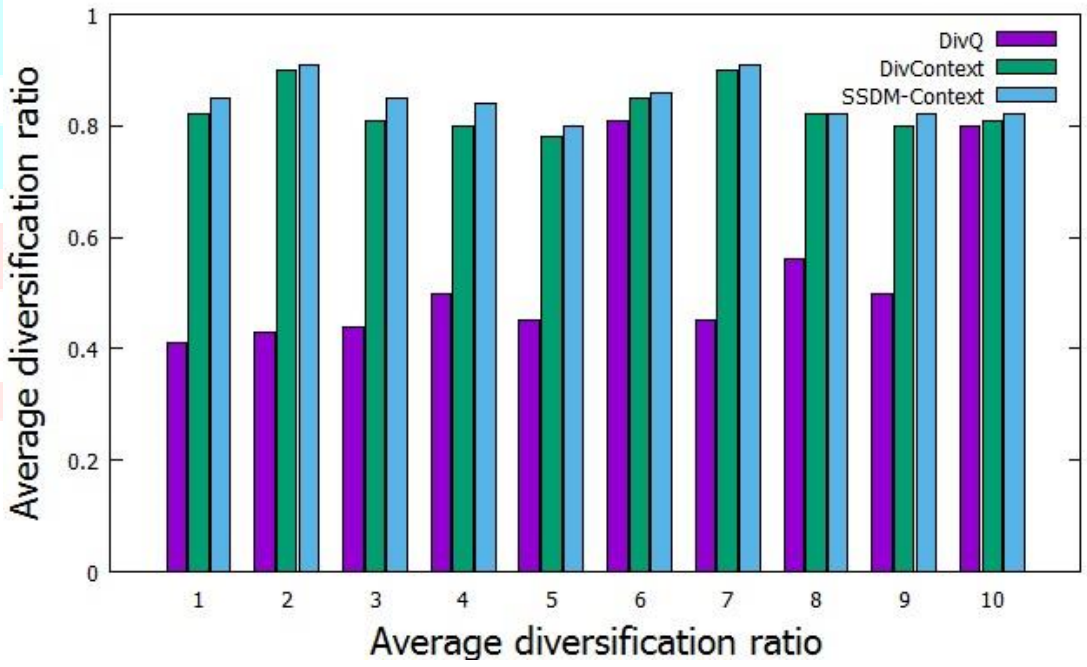

Figure 8: Differentiation of diversified query result.

5.5 Measure of Usefulness of query suggestions

The usefulness of a qualified suggestion model should consider the relevance of the query suggestions, as well as the novelty of the results to be generated by these query suggestions Fig. 9. shows this. Ten queries to the DBLP dataset is prepared. Ten vague queries for these ten queries are prepared. Query suggestions are prepared for the queries. Top-k query suggestions over DBLP-dataset are considered. If a query is obtained with the help of top-5 query suggestions then it is assigned with a score of 1. If it is matched to top (6-10) query suggestions then it is assigned with a score of 0.5, and for top (11-25) suggestions the score is 0.25 and more than 25th position the score is zero. The proposed AESP considers semantic results for the generated queries. Only AESP answers well about query q8 because only AESP preserves semantic queries by incorporating fuzzy set.

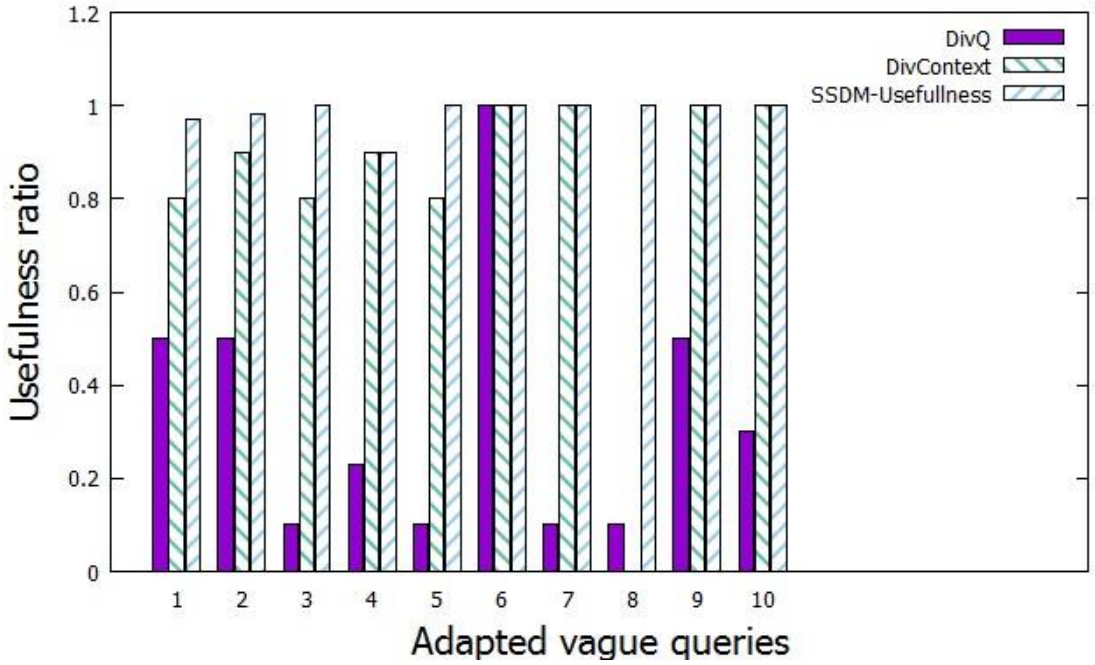

Figure 9: Usefulness of diversification model.

VI. CONCLUSION

The proposed Semantic Search Diversification method identifies the contexts of the input query by balancing the relevance and novelty of the queries. The system used a fuzzy set of keywords in a semantic table as a candidate term alias to achieve semantic preservation. In SSDM search diversification is achieved by evaluating the mutual information score for query keywords and keywords with semi features. The anchor nodes are determined initially by extracting the SLCA nodes for the original queries and their semantic variants from the XML Data. The Anchor based parallel semantic preservation algorithm proceeds to the ancestor, descendant and next levels of branches of the tree to list out top-k SLCA nodes for each assigned region. The experimental analysis over the DBLP dataset shows good results in terms of efficiency, usefulness and effectiveness of diversification. The Ndcg measures of the proposed work show high relevancy i.e. more than 75 percent of ideal query suggestions. High usefulness is achieved for vague queries with less number of relevant and novel query suggestions.

REFERENCES

[1] Herve Abdi and Lynne J Williams. Principal Component Analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433-459, 2010.

[2] Maria Biryukov and Cailing Dong. Analysis of Computer Science Communities based on DBLP. International Conference on Theory and Practice of Digital Libraries, pages 228-235, 2010.

[3] Chenliang, Aixin Sun, Li, Jianshu Weng, and Qi He. Tweet Segmentation and Its Application to Named Entity Recognition. IEEE Transactions on Knowledge and Data Engineering, 27(2):558-570, 2015.

[4] I Jen Chiang, Charles Chih Ho Liu, Yi Hsin Tsai, and Ajit Kumar. Discovering Latent Semantics in Web Documents using Fuzzy Clustering. IEEE Transactions on Fuzzy Systems, 23(6):2122-2134, 2015.

[5]Zhicheng Dou, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song. Automatically Mining Facets for Queries from Their Search Results. IEEE Transactions on Knowledge and Data Engineering, 28(2):385-397, 2016.

[6]Binbin Gu, Zhixu Li, Xiangliang Zhang, An Liu, Guanfeng Liu, Kai Zheng, Lei Zhao, and Xiaofang Zhou. The Interaction between Schema Matching and Record Matching in Data Integration. IEEE Transactions on Knowledge and Data Engineering, 29(1): 186-199, 2017.

[7] Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram. XRANK: Ranked Keyword Search over XML Documents. Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pages 16-27, 2003.

[8] Wang Haoxiang and S Smys. Big Data Analysis and Perturbation using Data Mining Algorithm. Journal of Soft Computing Paradigm (JSCP), 3(01):19-28, 2021.

[9] Dongwoo Kim, Haixun Wang, and Alice H Oh. Context-Dependent Conceptualization. *IJCAI*, pages 2654-2661, 2013.

[10] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From Word Embeddings to Document Distances. International Conference on Machine Learning, pages 957-966, 2015.

[11] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: Named Entity Recognition in Targeted Twitter Stream. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721-730, 2012.

[12] Jianxin Li, Chengfei Liu, and Jeffrey Xu Yu. Context-based Diversification for Keyword Queries over XML Data. IEEE Transactions on Knowledge and Data Engineering, 27 (3):660-672, 2015.

[13] Jian Liu and DL Yan. Answering Approximate Queries over XML Data. IEEE Transactions on Fuzzy Systems, 24(2):288-305, 2016.

[14] Junqiang Liu, Ke Wang, and Benjamin CM Fung. Mining High Utility Patterns in One Phase without Generating Candidates. IEEE Transactions on Knowledge and Data Engineering, 28(5):1245-1257, 2016.

[15] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. Recognizing Named Entities in Tweets. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1:359-367, 2011.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 2013.

[17] Pankesh Patel, Muhammad Intizar Ali, and Amit Sheth. From Raw Data to Smart Manufacturing: AI and Semantic Web of Things for Industry 4.0. IEEE Intelligent Systems, 33(4):79-86, 2018.

[18] Jennifer S Raj. Machine Learning Implementation in Cognitive Radio Networks with Game Theory Technique. Journal: IRO Journal on Sustainable Wireless Systems, 2020(2): 68-75, 2020.

[19] Francois Role and Mohamed Nadif. Handling the Impact of Low Frequency Events on Co-occurrence based Measures of Word Similarity. Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Scitepress, pages 218-223, 2011.

[20] V Suma and Shavige Malleshwara Hills. Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics. Journal of Soft Computing Paradigm (JSCP), 2(03): 153-159, 2020.

[21] Chong Sun, Chee-Yong Chan, and Amit K Goenka. Multiway SLCA-based Keyword Search in XML Data. Proceedings of the 16th international conference on World Wide Web, pages 1043-1052, 2007.

[22] KR Venugopal, KG Srinivasa, and Lalit M Patnaik. Soft Computing for Data Mining Applications. Springer, 2009.

[23] Eliska vestakov and Jan Janouek. Automata Approach to XML Data Indexing. *Information*, 9(1):12, 2018.

[24] Marcos R Vieira, Humberto L Razente, Maria CN Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina, and Vassilis J Tsotras. On Query Result Diversification. Data Engineering (ICDE), 2011 IEEE 27th International Conference on, pages 1163-1174, 2011.

[25] Lidong Wang. Heterogeneous Data and Big Data Analytics. Automatic Control and Information Sciences, 3(1):8-15, 2017.

[26] Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A Survey on Evolutionary Computation Approaches to Feature Selection. IEEE Transactions on Evolutionary Computation, 20(4):606-626, 2016.

[27] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Understanding Short Texts through Semantic Enrichment and Hashing. IEEE Transactions on Knowledge and Data Engineering, 28(2):566-579, 2016.

[28] Rui Zhao and Kezhi Mao. Fuzzy Bag-of-Words Model for Document Representation. IEEE Transactions on Fuzzy Systems, 26(2):794-804, 2017.