A Robust Document Proximity Based Approach and Location Aware Keyword Query Suggestion in Web

Y.S.Sai Priya, M.Tech 2nd Year,Dept.Of CSE, VRS & YRN College of Engineering and Technology , Chirala,India

Damarla Sree Latha,Associate professor,Dept of CSE, VRS & YRN College of Engineering and Technology , Chirala,India

ABSTRACT:
Keyword suggestion in web search helps user to access relevant information without having to known how to precisely express their queries Exiting keyword suggestion techniques do not consider the location of user and the query result the spatial proximity of user to the retrieved result is not taken as a factored in the recommendation. However the relevance's of search result in many application location based services is known to be correlated with proximity to the query issuer. Each query is related to one of topics identified in the conversion fragments preceding the recommendation and is submitted to a search engine over the English we propose in this paper an algorithm foe diverse merging of these lists using a sub modular reward function that reward the topical similar of documents to the conversation words as well as there diversity. We evaluates the proposed method through crowd sourcing the result superiority of the diverse merging technique over Several other which enforce the diversity of topics

KEYWORDS: Query suggestion, Spatial Databases, Document Proximity.

I.INTRODUCTION

Keyword suggestion (also known as query suggestion) has become one of the most fundamental features of commercial web search engines. After submitting a keyword query, the user may not be satisfied with the results, so the keyword suggestion module of the search engine recommends a z set of m keyword queries that are most likely to refine the user's search in the right direction. Effective keyword suggestion methods are based on click information from query logs and query session data, or query topic models new keyword suggestions can be determined according to their semantic relevance to the original keyword query. However, to our knowledge, none of the existing methods provide location-aware keyword query suggestion (LKS), such that the suggested queries retrieve documents not only related to the user information needs but also located near the user location. This requirement merges due to the popularity of spatial keyword search Google processed a daily average of 4.7 billion queries in 2011, a substantial fraction of which have local intent and target spatial web objects (i.e., points of interest with a web presence having locations as well as text Data mining is the information of domain we are mining like concept hierarchies, to organize attributes onto various levels of abstraction. A Spatial Keyword query is an approach of searching qualified spatial objects by considering both the query requester's location and user specified keywords. Taking both spatial and keyword requirements into account, the goal of a spatial keyword query is to efficiently find results that satisfy all the conditions of a search. Searching is a common activity happening in data mining. This motivated to develop methods to retrieve spatial objects. A spatial object consists of objects associated with spatial features. In other words, spatial objects involve spatial data along with longitude and latitude of location. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in acertain area, while nearest neighbor retrieval can discover the restaurant closest to a given address. However, existing keyword suggestion techniques do not consider the locations of the users and the query results. Users often have difficulties in expressing their web search needs they may not know the keywords. After submitting a keyword query, the user may not be satisfied with the results

II. LITERATURE SURVEY

The location aware keyword(LKS) query suggestion method provide the suggested queries retrieve documents which is related to user information and located near to users location.LKS framework, it construct and use keyword document bipartite graph(KD graph) that connect to keyword queries with their relevant document. LKS adjust weight on edges in KD graph to capture the semantics relevance between keyword queries and spatial distance between document location and user location. For distance calculation the Personalized PageRank(PPR) algorithm is used, it uses Random walk with restart(RWR) on KD graph, starting from user supplied query to find the set of keywords and spatial proximity to the user location. But RWR search has high computational cost on large graph to address this issue; a new portion based algorithm is used to reduce the cost of RWR search.

Authors in [1] propose a novel context-aware query suggestion approach which is in two steps. In the offline model-learning step, to address data sparseness, click-through bipartite is clustered in order to summarize queries into concepts. In this approach queries are suggested to the user in a context-aware manner.

Authors in [2] propose a novel query suggestion algorithm based on ranking queries with the hitting time on a large scale bipartite graph. This method captures the semantic consistency between the suggested query and the query given by user. Experiments show time is effective to generate semantically consistent query suggestions. The proposed algorithm and its variations can successfully execute huge queries, accommodating query suggestion.

Author [3] introduced novel, domain-independent and privacy preserving methods for enhancing MF models by expanding the user-item matrix and by imputation of the user-item matrix, using browsing logs and search query logs. They introduced two approaches to enhancing user modeling using these data. Authors show that CF systems can be enhanced using Internet browsing data and search engine query logs, both represent a rich profile of individuals' interests. They demonstrate the value of their approach on two real datasets each comprising of the activities of tens of thousands of individuals. The first dataset details the download of Windows Phone 8 mobile applications and the second - item views in an online retail store. Both datasets are enhanced using anonym zed Internet browsing logs.

Author [4] proposed a new query suggestion paradigm, Query Suggestion with Diversification and Personalization that effectively integrate diversification and personalization into one unified framework. In the QS-DP, the suggested queries are successfully diversified to cover different facets of the input query and the ranking of the suggested queries are personalized to ensure that the top ones that align with a user's personal preferences. They propose a new representation for query log. The proposed multi-bipartite-graph representation comprehensively captures different kinds of relations between search queries in query log. Based on the multi-bipartite-graph representation, they design two strategies to identify the most relevant suggestion candidate.

Author [5] proposed a method that computes likeness among queries based on "Query- Clicked Sequence" model. This model counts weight of clicked document term by density of documents containing this term on clicked sequence, and filters content of unrelated documents during similarity computation. Based on the characteristics of different concentration on relevant and irrelevant documents occurring on clicked document sequence, this paper proposed a query similarity computing method based on irrelevant feedback analysis, and recommended queries based on this method. This method constructs a relevant term collection for each clicked sequence of one query, from relevant document and computes similarity among queries by relevant term collection offline with recommendation of online queries based on the computation result. Query recommendation based on their method can effectively decrease the negative effect on query similarity computation, and increase accuracy of query similarity computation, therefore increase accuracy of query recommendation, especially for informational queries.

Author [6] developed the QueRIE system for personalized query recommendations. QueRIE monitors the user's querying behavior and finds matching patterns in the system's query log, identifying same kind of users. These queries are used to recommend queries which user may find helpful. They explore the use of latent factor models when, instead of ratings, the input consists of database-query log data. And explored how latent factor models, and in particular matrix factorization using ALS, affect the quality of the recommendations and computational efficiency of their framework. Such techniques have become very popular in traditional rating-based recommender systems, and in this work authors verified that they capture latent similarities between users and "items" even when the input is not explicit.

Author [7] proposed time aware structured query suggestion which clustered query suggestion along timeline so the user can narrow down his search from a temporal point of view. When the suggested query is clicked the method presents web pages from query-URL bipartite graph. After ranking those according to click count within a particular time period this method helping user to access relevant web pages. It free the users from burden of entering a specific time constraint with query, this method can be used in the context of real user search tasks.

Author [8] Explained a web recommender approach based on learning from web logs it recommends user a list of pages that are relevant to the users proposed query by comparing with historic pattern and also rerank the result pages. This system proves to be efficient as the pages desired by the users are on the top in the result list and this method reduces the search time of the user. In this the recommendation is based on the feedback of users and web log analysis.

Author [9] proposed a snippet based method to facilitate users with query recommendations. The concepts related to the users information needs are suggested to the users to satisfy their current information need, extracted the concepts from the web snippet. Authors proposed two weight functions to measure the relevance between query and concept. Related concepts with different meaning are selected and recommended as query suggestions to the users.

Author [10] presented an approach based on the users search behaviour. Their suggested query recommendation framework follows the fact that if user clicks certain result returned by search engine then if does not necessarily mean that the user is interested in that result but if probably reflects that the user is instead interested in the snippets of the result. This is because that up to that time the user clicks certain result just by viewing the snippet, the resultant document has not shown to user by that time.

Author [11] has explained a dynamic knowledge based approach which gets updated by continuously as queries are issued, to keep record of possible variations of user interest. This model extensively guesses the real hidden intent of user behind a submitted query and proves its effectiveness by dropping the effect of aging by updating & rebuilding the query recommendation model incrementally. In this the update operation runs in parallel with the query processor. Thus this dynamic knowledge based approach is better than that of all static models based on query log.

Author [12] designed a location-aware keyword query suggestion framework. They propose a weighted keyword-document graph, which captures both the semantic significance between keyword queries and the spatial distance between the resulting documents and the user location. The graph is browsed in a random-walk-with-restart fashion, to select the keyword queries with the highest scores as suggestions. To make framework scalable, authors propose a partition-based approach that outperforms the baseline algorithm by up to an order of magnitude. And design the first ever Location-aware Keyword query Suggestion framework, for suggestions relevant to the user's information needs that also retrieve relevant documents close to the query issuer's location. Also extend the state-of-the-art Bookmark Colouring Algorithm (BCA) for RWR search to compute the location-aware suggestions.

Keyword query suggestion approaches can be classified into three main categories: random walk based approaches, learning to rank approaches, and clustering based approaches. We also briefly review alternative methods that do not belong to any of these categories. To the best of our knowledge, no previous work considers user location in query suggestion.

The methods in this category use a graph structure to model the information provided by query logs, and then apply a random walk process on the graph to compute the suggestions. Craswell and Szummer [13] apply such an approach on the query-click graph and suggest queries based on personalized PageRank scores.

Some query suggestion approaches [14] are based on learning models trained from co-occurrences of queries in search logs. Another learning-to-rank approach [15] is trained based on several types of query features, including query performance prediction. Beeferman and Berger [16] view the query log as a queryURL bipartite graph. By applying an agglomerative clustering algorithm on the vertices in the graph, query clusters can be identified.

Zhang and Nasraoui [17] create a graph with edges between consecutive queries in each session, weighted by the textual similarity between these queries. A candidate suggestion for a given query is given a score based on the length of the path between the two queries, aggregated across all sessions in a query log where the query and the suggestion cooccurred.

References [18] and [19] both study the problem of location aware type-ahead search, also known as instant search. LTAS finds documents near a user location, as the user types in a keyword query character by character.

Location-Aware Suggestions Based on User History Google [20] provides location-based query suggestions by simply selecting the user's past search queries that have results close to the user's current location. These suggestions may be insufficient if the user did not perform any historical searches near her current location. In addition, query suggestion based on location only may not match the user's search intent.

A relevant problem to query suggestion in relational databases is called query relaxation. The objective is to generalize an SQL query in case of too few or no results [21]. Query relaxation approaches cannot be applied for keyword query suggestion, because they require the relaxed query to contain the results of the original query, which is not essential in our case.

The basic structure of the KD-graph used in our model and other existing suggestion model is one type of heterogeneous graph that consists of multiple types of nodes and edges. There exist some research focus on the similarity search in heterogeneous graphs.

PathRank [22] extends the Personalized PageRank algorithm on heterogeneous graphs by discriminating different paths during the random walk process guided by predefined meta-paths

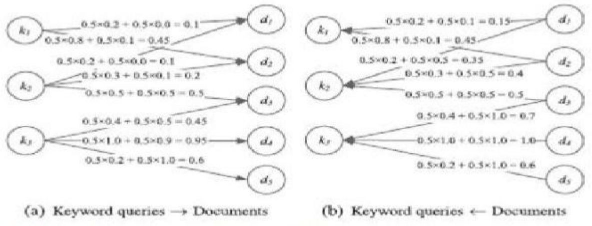
Random Walk Computation

Random walk with restart, also known as Personalized PageRank, has been widely used for node similarity measures in graph data, especially since its successful application by the Google search engine. Matrix-based methods [23], [24] solve PPR by precomputing the inversion matrix. Tong et al. [23] propose a matrix-based approach B_LIN that reduces the pre-computation cost of the full matrix inversion by partitioning the graph.

MC can also be applied online, without relying on pre-computations; a number of random walks are tried from the query node and the PPR score of other nodes are estimated from these samples [24]. However, as shown later in [25], a large number of (expensive) random walks are required in order to achieve acceptable precision.

3.LKS FRAMEWORK:

Consider a user-supplied query q with initial input kq; kq can be a single word or a phrase. Assuming that the query issuer is at location _q, two intuitive criteria for selecting good suggestions are: (i) the suggested keyword queries (words or phrases) should satisfy the user's information needs based on kq and (ii) the suggested queries can retrieve relevant documents spatially close to _q. The proposed LKS framework captures these two criteria. relevant documents spatially close to _q. The proposed LKS framework captures these two criteria.



(a) Keyword queries → Documents (b) Keyword queries ← Documents

Fig. 2. Location-aware edge weight adjustment.

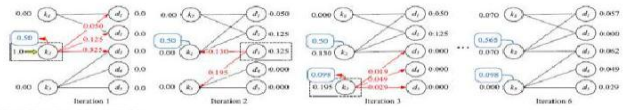


Fig. 3. Illustration of Algorithm BA.

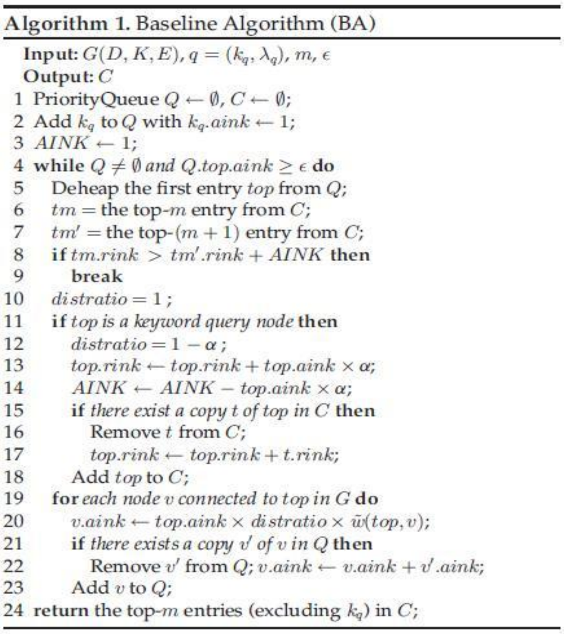
4. ALGORITHMS:

In this section, we introduce a baseline algorithm (BA) for location-aware suggestions (Section 3.1). Then, we propose our efficient partition-based algorithm (Section 3.2).

4.1 Baseline Algorithm (BA):

We extend the popular Bookmark-Coloring Algorithm

[25] to compute the RWR-based top-m query suggestions as a baseline algorithm. BCA models RWR as a bookmark coloring process. Starting with one unit of active ink injected into node kq, BA processes the nodes in the graph in descending order of their active ink. Different from typical personalized PageRank problems [27], [28] where the graph is homogeneous, our KD-graph Gq has two types of nodes: keyword query nodes and document nodes. As opposed to BCA, BA only ranks keyword query nodes; a keyword query node retains a portion of its active ink and distributes 1^a portion to its neighbor nodes based on its outgoing adjusted edge weights, while a document node distributes all its active ink to its neighbor nodes.



Algorithm 1. Baseline Algorithm (BA)

```
Input: $G(D,K,E), q = (k_q, \lambda_q), m, \epsilon$
Output: $C$
PriorityQueue $Q \leftarrow \emptyset, C \leftarrow \emptyset$;
Add $k_q$ to $Q$ with $k_q.aink \leftarrow 1$;
$AINK \leftarrow 1$;
while $Q \neq \emptyset$ and $Q.top.aink \geq \epsilon$ do
    Deheap the first entry $top$ from $Q$;
    $tm =$ the top-$m$ entry from $C$;
    $tm' =$ the top-$(m+1)$ entry from $C$;
    if $tm.rink > tm'.rink + AINK$ then
        break
    $distratio = 1$ ;
    if $top$ is a keyword query node then
        $distratio = 1 - \alpha$ ;
        $top.rink \leftarrow top.rink + top.aink \times \alpha$;
        $AINK \leftarrow AINK - top.aink \times \alpha$;
        if there exist a copy $t$ of $top$ in $C$ then
            Remove $t$ from $C$;
            $top.rink \leftarrow top.rink + t.rink$;
        Add $top$ to $C$;
    for each node $v$ connected to $top$ in $G$ do
        $v.aink \leftarrow top.aink \times distratio \times \tilde{w}(top, v)$;
        if there exists a copy $v'$ of $v$ in $Q$ then
            Remove $v'$ from $Q$; $v.aink \leftarrow v.aink + v'.aink$;
        Add $v$ to $Q$;
return the top-$m$ entries (excluding $k_q$) in $C$;
```

5.MATHEMATICAL MODEL

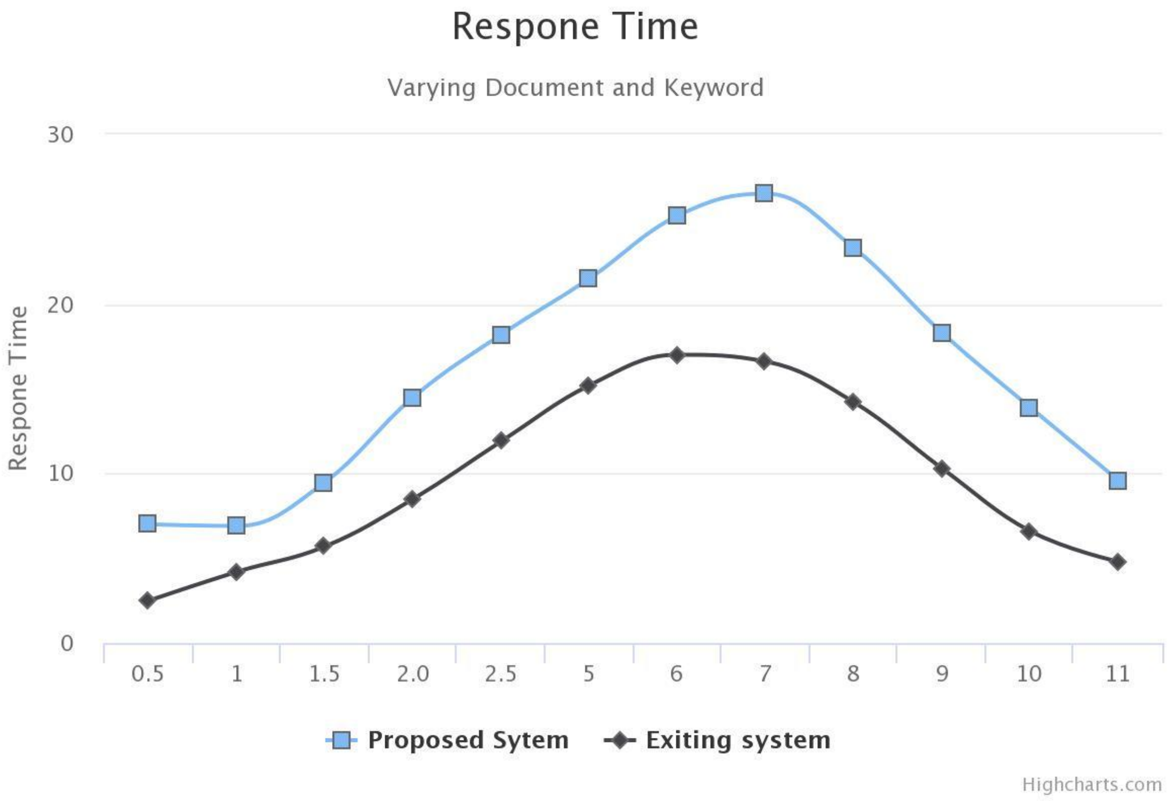
- Let S, be a system such that,
- S = {s,Subi,Pubi,Si,E}
- Where,
- S- Proposed System
- s- Initial state at
- Subi = the Registered Subscriber with attributes {Sid,pwd,Li} where
- Sid = subscriber_id
- Li = Location_id
- Pubi = the Registered Subscriber with attributes {Pid,pwd,Li} where
- Li = Location_id
- E – Event
- N – Node
- S – Subscription
- UB – Upper Bound
- Sid – Identifier an subscription

X- Input of System.

-L(Location),S(Searching) and R tree node

Y- Output of System. – Top K Subscription matching and show in Map

6.EXPERIMENTAL RESULT



7. CONCLUSION

In this paper, we proposed an LKS framework providing keyword suggestions that are relevant to the user information needs and at the same time can retrieve relevant documents near the user location. A baseline algorithm extended from algorithm BCA is introduced to solve the problem then, we proposed a partition-based algorithm which computes the scores of the candidate keyword queries at the partition level and utilizes a lazy mechanism to greatly reduce the computational cost.

Empirical studies are conducted to study the effectiveness of our LKS framework and the performance of the proposed algorithms. The result shows that the framework can offer useful suggestions and that PA outperforms the baseline algorithm significantly. In the future, we plan to further study the effectiveness of the LKS framework by collecting more data and designing a benchmark. In addition, subject to the availability of data, we will adapt and test LKS for the case where the locations of the query issuers are available in the query log. Finally,We believe that PA can also be applied to accelerate RWR on general graphs with dynamic edge weights; we will investigate this potential in the future.

8.REFERENCES

[1] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, *Context-aware query suggestion by mining click-through and session data,* in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2008, pp. 875–883.

[2] Mei, Qiaozhu, Dengyong Zhou, and Kenneth Church. *Query suggestion using hitting time.* Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

[3] Royi Ronen and et.al, *Recommendations Meet Web Browsing: Enhancing Collaborative Filtering using Internet Browsing Logs*, IEEE 32nd International Conference on Data Engineering (ICDE), pp 1230 – 1238, 2016.

[4] Di Jiang, Kenneth Wai-Ting Leung, Lingxiao Yang, Wilfred Ng, *Query suggestion with diversification and personalization*, Knowledge-Based Systems. Volume 89, Pages 553–568, 2015.

[5] Bo Zhang, Bin Zhang, Shubo Zhang, Chao Ma, “Query Recommendation Based on Irrelevant Feedback Analysis”, 8th International Conference on Bio Medical Engineering and Informatics, pp 644-648, 2015.

[6] Magdalini Eirinaki , *QueRIE reloaded: Using matrix factorization to improve database query recommendations*, IEEE international conference on Big Data, pp 1500 – 1508, 2015

[7] T. Miyanishi and T. Sakai, *Time-aware structured query suggestion,* in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval,pp. 809–812, 2013.

[8] R. Bhushan and R. Nath. 2012. *Recommendation of optimized web pages to users using Web Log mining techniques*. Advance Computing Conference (IACC), 2013 IEEE 3rd International. IEEE, 2012.

[9] Goyal, Poonam, and N. Mehala. 2011. *Concept based query recommendation*. Proceedings of the Ninth Australasian Data Mining Conference-Volume 121. Australian Computer Society. Inc.

[10] Y. Liu, Junwei Miao, Min Zhang, Shaoping Ma, and Liyun Ru. 2011. *How do users describe their information need: Query recommendation based on snippet click model*. Expert Systems with Applications 38, no. 11 (2011): 13847-13856.

[11] D. Broccolo, O. Frieder, F. Nardini, R. Perego and F. Silvestri.2010. *Incremental Algorithms for Effective and Efficient Query Recommendation*. SPIRE 2010, LNCS 6393. pp. 13-24. Springer-Verlag Berlin Heidelberg.

[12] Shuyao Qi,Dingming Wu and Nikos Mamoulis, *Location aware keyword Query suggestion based on document proximity*, IEEE transaction paper on knowledge and data engineering, Vol 28, No 1, pp 82-97, 2016.

[13] N. Craswell and M. Szummer, *Random walks on the click graph*, in Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2007, pp. 239–246.

[14] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu, *Learning to suggest: A machine learning framework for ranking query suggestions*, in Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2012, pp. 25– 34.

[15] Y. Liu, R. Song, Y. Chen, J.-Y. Nie, and J.-R. Wen, *Adaptive query suggestion for difficult queries*, in Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2012, pp. 15–24.

[16] D. Beeferman and A. Berger, *Agglomerative clustering of a search engine query log*, in Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2000, pp. 407–416.

[17] Z. Zhang and O. Nasraoui, *Mining search engine query logs for query recommendation*, in Proc. 15th Int. Conf. World Wide Web, 2006, pp. 1039–1040

[18] S. Basu Roy and K. Chakrabarti, *Location-aware type ahead search on spatial databases: Semantics and efficiency*, in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 361–372.

[19] R. Zhong, J. Fan, G. Li, K.-L. Tan, and L. Zhou, *Location-aware instant search*, in Proc. 21st ACM Conf. Inf. Knowl. Manage., 2012, pp. 385–394.

[20] J. Myllymaki, D. Singleton, A. Cutter, M. Lewis, and S. Eblen, *Location based query suggestion*, U.S. Patent 8 301 639, Oct. 30, 2012

[21] T. Gaasterland, *Cooperative answering through controlled query relaxation*, IEEE Expert, vol. 12, no. 5, pp. 48–59, Sep. 1997.

[22] S. Lee, S. Park, M. Kahng, and S.-G. Lee, *PathRank: A novel node ranking measure on a heterogeneous graph for recommender systems*, in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., 2012, pp. 1637–1641

[23] H. Tong, C. Faloutsos, and J.-Y. Pan, *Fast random walk with restart and its applications*, in Proc. 6th Int. Conf. Data Mining, 2006, pp. 613–622.

[24] K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova, and M. Sokol, *Quick detection of top-k personalized PageRank lists*, in Proc. 8th Int. Workshop Algorithms Models Web Graph, 2011, vol. 6732, pp. 50–61.

[25] Y. Fujiwara, M. Nakatsuji, H. Shiokawa, T. Mishima, and M. Onizuka, *Efficient ad-hoc search for personalized PageRank*, in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 445–456.

Author's Profile:

Y.S. Sai Priya received B.Tech degree in Computer Science and Engineering and pursuing M.Tech in Computer Science and Engineering from VRS & YRN College of Engineering and Technology ,Dept of CSE,Chirala,Prakasam,India.

Damarla Sree Latha working as Associate professor in VRS & YRN College of Engineering and Technology ,Dept of CSE,Chirala,Prakasam,India.