IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Keyword Interrogation Implication on Document Vicinity Based on Location and Rating

A.A. Bhujugade[1*], D.V. Kodavade[2]

[1] Dept. of CSE, DKTE Society's Textile & Engineering Institute (An Autonomous Institute), Ichalkaranji, India
[2] Dept. of CSE, DKTE Society's Textile & Engineering Institute (An Autonomous Institute), Ichalkaranji, India

*Corresponding Author: akshaybhujugade65@gmail.com

Available online at: www.ijcseonline.org

Accepted: 18/Jul/2018, Published: 31/Jul/2018

***Abstract*—**One of the fundamental feature of web search engine is keyword suggestion. 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 set of alternative keyword queries that are most likely to refine the user's need. The suggested keywords are semantic relevance to keyword query. Spatial vicinity of user can be also consider to get suggestion in effective manner. In this paper, we develop location-aware keyword query suggestion framework considering the document distance and rating. The system uses keyword document graph for capturing semantic relevance between keyword queries and spatial distance of document and query issuers' location. The keyword document graph is browsed in random walk with restart fashion, for calculating the highest score for better keyword query suggestion. The baseline algorithm and partition-based algorithm uses RWR to compute top-m suggestions and based upon users selected keyword query the documents are ranked using bayesian ranking method.

***Keywords*—**Keyword query suggestion, Spatial objects, Document proximity.

I. Introduction

Search engines are the software programs that searches for the sites over the web according to user keyword query. While user enters the keyword, search engine suggest the keyword list which are semantic relevance to the query issuers keyword [2]. If users don't know how to express their queries he can use keyword suggestion which is used in web search so that, it can help the user to access relevant information. So the aim of keyword suggestion module is to satisfy the user with actual information needed. Some of the users may also have a local intent while searching. As per business 2 community survey 93 % of Google searches have local intent which motivated to develop the methods which retrieve spatial objects.[1]

A spatial object has a spatial data with longitude and latitude of the location. Spatial keyword query is a way of searching for the qualified spatial objects [3]. The spatial-keyword query considers the location of the query issuer and the keyword specified by the user. Considering the both spatial and keyword requirements, the goal of spatial keyword query is to find effectively search results that satisfy the search criteria. A keyword suggestion module can be developed that considers user location and accordingly suggestion will be suggested referring document rating with its distance.

A LKS Framework can be constructed considering document rating. As resulted keyword has ability to retrieve document which is near to query issuer's location referring document rating. A LKS framework has two criteria while suggesting keywords, the keyword suggested should be semantic relevance to original keyword query and the suggested keyword should have ability to retrieve nearby document. For satisfying the first criteria LKS users the Keyword Document graph. The KD graph has two types of nodes keyword nodes and document nodes. This two nodes are connected to each other by edges with weights. Second criteria is satisfied by location aware edge weight adjustment. The KD graph maps keyword queries with their relevance documents.

Random walk with restart gives the proximity score between two nodes in graph [4], [5]. RWR is used by many applications like recommendation system, automatic image captioning, etc. The goal of RWR is to find top-k highest proximities for a given node. To compute top-m suggestions LKS uses RWR search on KD graph. The BA and PA algorithms uses RWR to compute suggestions.

The paper is organized as follows Section I contains the introduction, Section II contain the related work of keyword suggestion techniques, Section III contain proposed work with architecture, Section IV contain implementation part of LKS framework with bayesian ranking, and in Section V the results are discussed, Section VI concludes research work and future scope.

II. RELATED WORK

Keyword query suggestion approaches can be classified into various types of categories like learning to rank approaches, clustering based approaches and random walk based approaches.

A. Learning to Rank Approaches

The methods proposed by Y. Liu et al. [6] is trained based on several types of query features, including query performance prediction. L. Li et al. [7] train a hidden topic model. For each candidate query, its posterior distribution over the hidden topic space is determined. Given a user query q, a list of suggestions is produced based on their similarity to q in the topic distribution space.

B. Clustering Based Approaches

R. Baeza-Yates et al. [8] discusses a methodology for a during query firing process search engine suggest m related queries, this related queries are based on queries issued previously, and can issued by user to redirect the search process. This will further improve the notion of interest of the suggested queries and to develop other notions of interest for the query recommender system. Query clustering is done to achieve the semantically similar queries. During clustering process, the use of the content of historical preferences of users is checked. The method also ranks related queries to relevance criteria.

H. Cao et al. [9] devised a context-aware query suggestion approach follows two-step offline model learning step and online query suggestion step. Offline model learning step, address data sparseness, queries are summarized into concepts by forming the cluster of click-through bipartite, Then sequence suffix tree is generated from session data for query suggestion model. In online query suggestion step, mapping of query sequence is achieved by capturing search context. By looking up the context in the concept sequence suffix tree this approach suggests queries to the users in context-aware manner.

C. Random Walk Based Approaches

P. Berkhin [10] presented BCA Computes authority weights over the web pages utilizing the web hyperlink structure. In the original BCA, a node distributes its ink aggressively and care only about the nodes with ink greater than ϵ. BCA can be optimized by using lazy updating Mechanism and spatial proximity caching. BCA results in a Bookmark coloring vector. BCA models RWR as a bookmark coloring process. N. Craswell et al. [11] discussed a search engine which has the ability to record the documents which were clicked for which query. In a weighted graph, RWR (Random Walk with Restart) gives the relevance score of two nodes. RWR specify how closely related the two nodes are in graph. RWR do not scale for large graphs. The Markov random walk is applied to a large click log. The advantage is it will retrieve relevant documents that are not yet been clicked for that query and rank effectively.

Q. Mei et al. [12] proposed algorithm for query ranking which was based on hitting time, it reflects the probability that a random walker arrives a node within certain steps. The proposed method controls the semantic consistency of the suggestions to the original query. The advantages of this methods are the generated suggestions are semantically consistent to the original query, the method boosts long tail queries as suggestion, etc. P. Boldi et al. [13] Query flow graphs are used than RWR is applied. The query-flow graph summarizes a query log in a compact representation. This representation can be obtained efficiently from the source data and enables several key search and mining operations. The query-flow graph is supports two key applications in usage mining.

Y. Song et al. [14] mine a term-transition graph from search engine logs and apply a topic-based unsupervised Page rank model that suggests queries based on the topic distribution and term-transition probability within each topic. M. P. Kato et al. [15] mentioned when there is rare or single-term queries input, the search engines should provide better assistance and according to searcher's current state they should dynamically provide query suggestions. It will further investigate the usage of query suggestion with datasets including user information to propose a query reformulation taxonomy specifically designed for query suggestion classification, and to improve query suggestion functionality based on our insights. T. Miyanishi et al. [16] devised Time-aware Structured Query Suggestion, it first clusters query suggestions from a temporal point of view and then presents web pages from query-URL bipartite graphs after ranking them according to their popularity within a specific time period.

Shuyao Qi et al. [1] devised bookmark coloring Algorithm that computes the RWR based on the top-m query suggestion as a baseline algorithm. BA processes the nodes in the graph in descending order of their active ink. BA only ranks keyword query nodes. The Baseline algorithm has drawbacks such as the no of iterations are more it is time-consuming process. BA is slow for several reasons. Only one node is processed at each iteration. If the number of iteration is more, there is an overhead of maintaining the queue. To improve the performance of BA, the partition based algorithm is proposed PA divides the documents and keyword queries in KD graph g into a number of groups. PA adopts a lazy mechanism that accelerates RWR search. As partitions are created the number of iteration are less, it is also time-saving process.

As LKS Framework captures two criteria for selecting good suggestion that is the suggested keyword should be semantic relevance to original keyword query, and it should have ability to retrieve nearby document. It doesn't check for document rating. So LKS Framework with Bayesian ranking method can be developed that captures two criteria that the keyword suggested should be semantic relevance to original keyword query and should have ability to retrieve document with minimum geo distance and well document rating.

III. PROPOSED WORK

The proposed work contains the location-based service during the query transaction. After calculating top-m keyword suggestion the suggested keyword his ability to retrieve those document which is near-by user location with good rating. Using Bayesian ranking method the selected keyword captures one criteria that is it has ability to rank documents that are near to user location with well rating. With Bayesian ranking we will like to rank good documents, the documents with lots of votes with minimum distance, at the top, the reason behind it is, the people are more likely to look at the top of the ranking than at the bottom, and we want to show people the documents that they will like. This can be done using the utility function[2],

$$E\ [\ n \times p \times X + n \times (1 - p) \times Y\ ]\ldots\ldots(1)$$

Where n is count of initial K-D Graph from selected keyword to particular document, p is the Euclidean distance from user location to document, x is the document rating and y is the five minus document rating.

So using the utility function the document ranking is done based on high utility value.

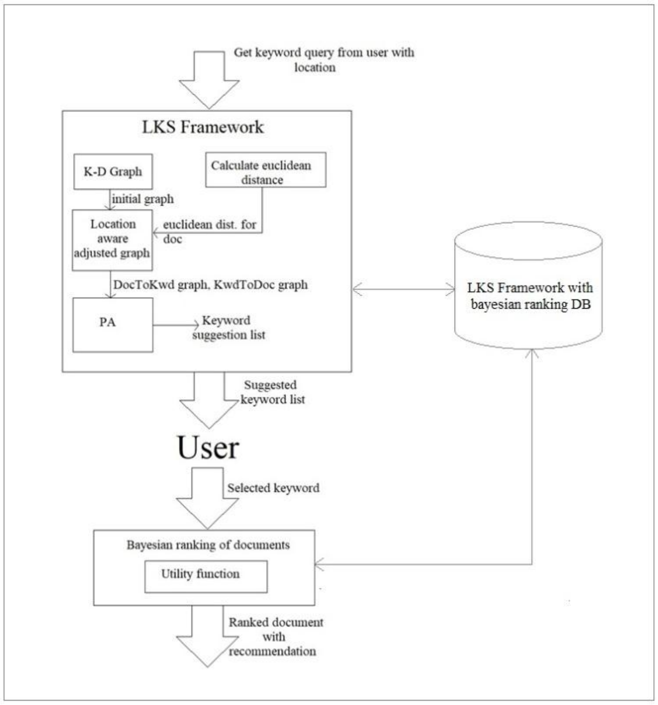
Get keyword query from user with location
LKS Framework
K-D Graph
Calculate euclidean distance
initial graph
Location aware adjusted graph
euclidean dist. for doc
DocToKwd graph, KwdToDoc graph
PA
Keyword suggestion list
LKS Framework with bayesian ranking DB
Suggested keyword list
User
Selected keyword
Bayesian ranking of documents
Utility function
Ranked document with recommendation

Figure 1. System architecture of LKS Framework with Bayesian ranking of documents

The system starts with taking user keyword query as input with location. The LKS framework computes top-m suggestions. The suggested keyword has ability to retrieve nearby document considering the document rating using bayesian ranking method. As LKS framework takes input as user location, the longitude and latitude of a user location is detected and euclidean distance is measured for all the geo documents in the system. The initial K-D graph is constructed from a query log file of search engine. The two graph are constructed that is, keyword to document adjusted graph and document to keyword adjusted graph. So finally to compute top-m suggestion the PA uses the keyword query, adjusted graphs as input for computation. PA gives suggestion list to user, based on user selected keyword the bayesian ranking method uses the initial graph and identifies the documents connected to selected keyword and applies the utility function to rank the documents. The documents are ranked based on high utility value. The first two documents who is having highest utility value is recommended by the system.

IV. Implementation

A. LKS Framework

LKS Framework captures two criteria for good suggestion that is the suggested keyword should be semantic relevance to original keyword query, and it should have ability to retrieve nearby document. As the dataset contains the keyword with URL clicked the initial K-D Graph is generated. Here the edge weights are defined by the number of count with same keyword query with same URL clicked. There will be no edge between keyword node to document node in graph, if there is no entry in dataset with that keyword and URL clicked. The whole graph is represented in matrix. The rows represents the keyword and the columns represents the documents i.e URL. If there is edge from keyword node to document node the edge weight are defined in specific matrix cell, and if there is no edge from keyword node to document node the matrix cell value is set to zero. This initial K-D graph needs to be constructed only once. When user enters in system the longitude and latitude of location is detected. After detecting the Euclidean distance is measured for all the documents in the system and table is maintained. After this the location aware edge weight adjustment is done, and by doing it the weights of edges are increased of those documents that are near to user's location. The edge weight adjustment is done by maintaining two different graphs that are keyword to document adjustment graph and document to keyword adjustment graph using following equations [1],

$$\tilde{\omega}(e) = \beta \times \omega(e) + (1 - \beta) \times \left(1 - dist\left(\lambda_q, d_j . \lambda\right)\right) \ldots \ldots (2)$$

Where $\omega(e)$ is the initial weight of edge from initial KD graph, $dist\left(\lambda_q, d_j . \lambda\right)$ is the euclidean distance for document d_j and β is set to 0.5 to consider the location of query issuer. By using $\tilde{\omega}(e)$ equation (2) the Keyword to Document adjustment graph is generated.

$$\tilde{\omega}(e') = \beta \times \omega(e') + (1 - \beta) \times (1 - mindist(\lambda_q, D(k_i))) \ldots (3)$$

Where $\tilde{\omega}(e')$ is the adjusted weight from keyword to document graph, $\omega(e')$ is the initial weight of edge from initial KD graph, $mindist(\lambda_q, D(k_i))$ is the minimum euclidean distance for documents and β is set to 0.5 to consider the location. When β is set to 1 the query issuers location is ignored and when it set to zero many documents are retrieved that are not relevant to the initial input. To compute top-m suggestions the PA or BA can be used. Baseline algorithm starts with injecting 1 amount of ink to users provided keyword query from KD Graph. The inputs to BA are KD graph, Adjusted KD graphs, user query with location, m etc. Ɛ is the termination condition. The two queue Q, C are used. Q stores the nodes that will be processed. In C we store Candidate suggestions. When 1 amount of ink is injected to users keyword query than it's retain ink is calculated. Only keyword node will calculate retain ink. The keyword node will distribute the calculated ink based on edge weight adjustment from keyword node to document node adjustment. After distributing ink towards document node the maximum active ink node is selected and total available ink is distributed towards keyword node according to edge weight adjustment from document node to keyword node adjustment. This process will continue up to termination criteria. Finally C has the Suggestions.

As initial KD graph contains keyword nodes and document nodes the partitions are created of keyword nodes and document nodes by random partitioning method for calculating top-m suggestion using partition based algorithm. As of baseline algorithm the ink is distributed from a node to partition. The overall working of PA is same as of BA with some difference like lazy distribution mechanism means a node has a buffer where it maintains the ink of a partition that is less than Ɛ. If again the buffer value is changed and greater than Ɛ after some iteration than it has ability to distribute ink to that node. As of BA as soon as keyword node calculate it's retain ink, it enters in C and same finally C has Suggestions.

B. Bayesian Ranking of documents

The Utility function (1) is used for ranking of documents which are connected to keyword node so based on the selected keyword the document are ranked which are connected to selected keyword.

Following is the pseudo code for ranking documents

Pseudo code: Bayesian ranking of documents

```
Input: kq
Output: Ranked documents
Get kq;
for each node v connected to kq do
    E [ n × p × X + n × (1 - p) × Y ]
return v by highest utility E
```

Where n is count of initial KD Graph from selected keyword to particular document, p is the Euclidean distance from user location to document, x is the document rating and y is the five minus document rating.

V. Results and Discussions

The LKS framework with bayesian ranking were implemented using java. The experiments were run on machine with Intel core i5-6200U 2.30 GHz and 4 GB main memory. MySQL database server is used at backend to store, retrieve and perform operations on tables stored in database server.

The dataset used for evaluation is query log file of search engine. The each record in the log contains a keyword query, the time when query was submitted, a URL clicked, and rating. The KD graph has keyword nodes and document nodes or URL nodes. The edge between a keyword query node ki and a URL node dj, if there exist records containing both keyword and URL in log record. The edge weights of KD graph are defined by the count of same ki and di in query log file. The edge weights are normalized by dividing the maximum number of clicks in the log for any query-document pair. The graph is represented in the form of matrix. Where the rows represents the keywords and the columns represents the documents, if there is no edge from keyword node to document node the matrix cell value is set to zero. The initial KD graph is constructed only once.

The evaluation metrics verify whether the keyword suggestions are semantically relevance to the original keyword query and able to find documents that are close to query issuer's location with well document rating. The PA was experimented with giving many different input query and some of them are "breakfast" and "clinic". The user location was latitude = 16.704076, longitude = 74.444444, according to this location the suggestions where computed. PA computes suggestions "Nasta center" and "Udapi nasta" for user supplied keyword query "breakfast". So by selecting nasta center suggestion the bayesian ranking method ranks the documents which are connected to keyword query node nasta center from KD graph. Following is the graph which shows user location λq and documents ranked according user selected keyword "Nasta center". The documents are ranked according to one criteria that the documents should be near to user location with well rating.

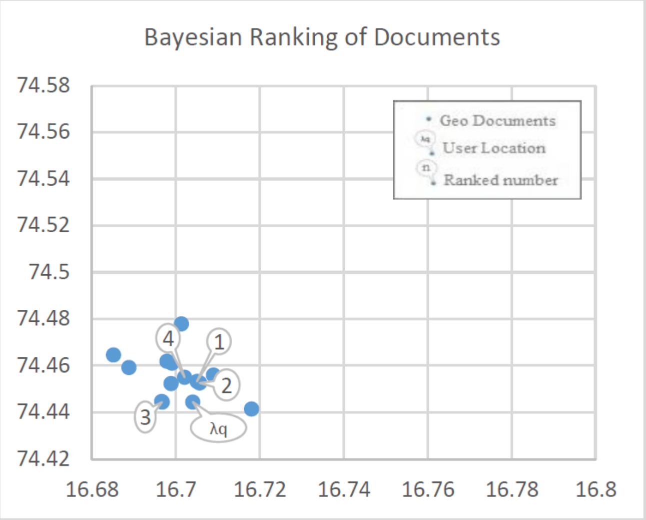

Figure 2. Graph showing bayesian ranking of documents by selecting keyword "Nasta center"

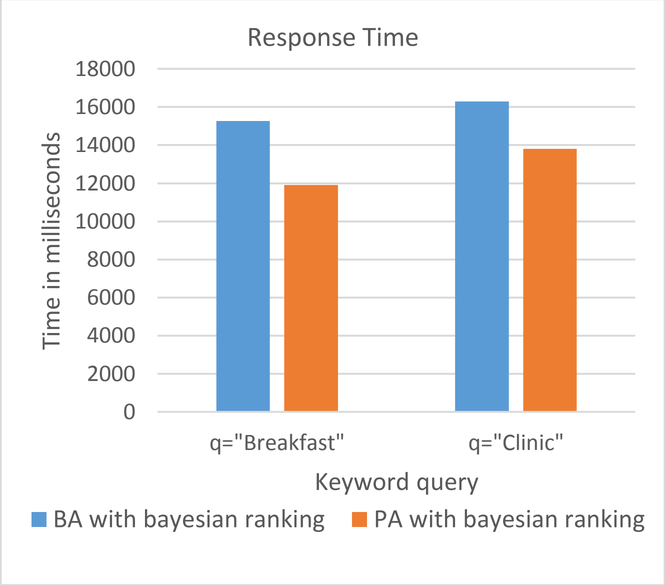

Figure 3. Graph comparing the response time of both BA with bayesian ranking and PA with bayesian ranking

Above graph shows the response time of both the PA with bayesian ranking and BA with bayesian ranking, we can notice that the response time of PA with bayesian ranking is less than BA with bayesian ranking. In baseline algorithm at each iteration we noticed that only one node is processed. As the number of nodes are more the termination conditions are meet after to many iterations. So BA takes time to compute suggestions than PA. In PA the partition of keyword query nodes and document nodes are done using random partitioning method. And also we notice that the number of iterations of PA are less than BA.

Following is the table showing results of BA with Bayesian ranking and PA with Bayesian ranking. We tried same keyword query with same dataset on two different algorithms BA and PA.

Keyword	**BA with Bayesian ranking**		**PA with Bayesian ranking**	
	BA (Time in ms)	**Ranking** (Time in ms)	**PA** (Time in ms)	**Ranking** (Time in ms)
Breakfast	13945	1318	10614	1303
	Total = 15263 ms		Total = 11917 ms	
	Iterations = 7	-	**Iterations** = 6	-
Clinic	14865	1422	12404	1397
	Total = 16287 ms		Total = 13801 ms	
	Iterations = 8	-	**Iterations** = 7	-

Table 1. Result Analysis

The both PA and BA are used to compute top-m suggestions. We conclude that the performance of PA with Bayesian ranking is better than BA with Bayesian ranking in both time (in milliseconds) and iterations. As PA terminates with minimum number of iterations the time taken is also minimum.

Based on selected keyword bayesian ranking rank the documents by capturing one criteria that is the document should be near to user location and also with good rating.

VI. CONCLUSION AND FUTURE SCOPE

The LKS Framework uses baseline algorithm and partition based algorithm to compute top-m keywords suggestion. The performance of partition based algorithm is effective than baseline algorithm. By using Bayesian ranking method the document are ranked based on maximum utility value that captures one criteria that the document should be near with good rating.

In the future, we plan to test LKS framework for the case where the locations of the query issuers are available in the query log file.

REFERENCES

[1] Shuyao Qi, Dingming Wu, and Nikos Mamoulis *"Location Aware Keyword Query Suggestion Based on Document Proximity,"* IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 28, NO. 1, JANUARY 2016.

[2] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu, *"SEAL: Spatio-textual similarity search,"* Proc. VLDB Endowment, vol. 5, no. 9, pp. 824– 835, 2012.

[3] D. Wu, G. Cong, and C. S. Jensen, *"A framework for efficient spatial web object retrieval,"* VLDB J., vol. 21, no. 6, pp. 797–822, 2012.

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

[5] Y. Fujiwara, M. Nakatsuji, M. Onizuka, and M. Kitsuregawa, *"Fast and exact top-k search for random walk with restart,"* Proc. VLDB Endowment, vol. 5, no. 5, pp. 442–453, Jan. 2012.

[6] 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, pp. 15–24, 2012.

[7] L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, and M. Kitsuregawa, *"An efficient approach to suggesting topically related web queries using hidden topic model,"* World Wide Web, vol. 16, pp. 273–297, 2013.

[8] R. Baeza-Yates, C. Hurtado, and M. Mendoza, *"Query recommendation using query logs in search engines,"* in Extending Database Technology, pp.588–596, 2004.

[9] 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, pp. 875–883, 2008.

[10] P. Berkhin, *"Bookmark-coloring algorithm for personalized pagerank computing,"* Internet Math., vol. 3, pp. 41–62, 2006.

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

[12] Q. Mei, D. Zhou, and K. Church, *"Query suggestion using hitting time,"* in Proc. 17th ACM Conf. Inf. Knowl. Manage., pp. 469–478, 2008.

[13] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, *"The query-flow graph: Model and applications,"* in Proc. 17th ACM Conf. Inf. Knowl. Manage., pp. 609–618, 2008.

[14] Y. Song, D. Zhou, and L.-w. He, *"Query suggestion by constructing term-transition graphs,"* in Proc. 5th ACM Int. Conf. Web Search Data Mining, pp. 353–362, 2012.

[15] M. P. Kato, T. Sakai, and K. Tanaka, *"When do people use query suggestion Inf. Retr.,"* vol. 16, no. 6, pp. 725–746, 2013.

[16] 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.

[17] Akshay A. Bhujugade, Dattatraya V. Kodavade *"A Survey on Keyword Interrogation Implication on Document Vicinity Based on Location,"* International Journal of Computer Engineering In Research Trends, Volume 4, Issue 11, pp. 514-518, November - 2017.

Websites

1. https://www.business2community.com/infographics/local-seo-statistics-must-know-infographics-01557523
2. http://julesjacobs.github.io/2015/08/17/bayesian-scoring-of-ratings.html

Authors Profile

Mr. Akshay A. Bhujugade pursed Bachelor of Engineering from Savitribai Phule Pune University, Pune in year 2016, He is currently pursuing Master of Technology from DKTE‘s TEI, (An Autonomous Institute), Ichalkaranji, India. His research work focuses on data mining, recommendation systems, network security.

Dr .D. V. Kodavade, the Head of Department of Computer Science & Engineering, at DKTE Society's Textile & Engineering Institute, Ichalkaranji, India. He is a member of the ACM, CSI, IEEE Computer Society. His current research interests includes Artificial Intelligence & Knowledge Based Systems, IoT, Neural Networks, Hybrid Intelligence.