e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Document Proximity: Keyword Query Suggestion Based On User Location

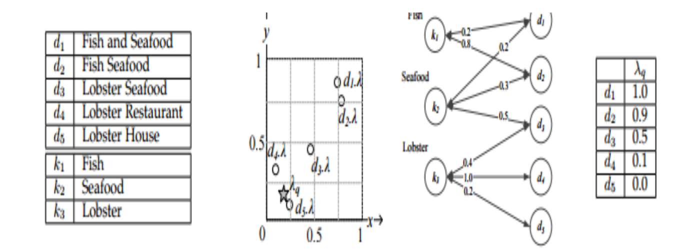
THOTA.MOUNIKA,MANDALA.SURESH KUMAR
PG Scholar, Department of CSE, Vaagdevi College of Engineering, Bollikunta
Warangal, Telangana, Mail id: mounika.thota2793@gmail.com,
Assiscant Professor Department of CSE, Vaagdevi College of Engineering, Bollikunta
Warangal, Telangana, Mail ID: mandala.suresh83@gmail.com,

ABSTRACT

The keyword suggestion in web search allows users to access relevant information without having to know how to express their queries accurately. Existing keyword suggestion techniques do not take into account user locations or query results. that is, the spatial proximity of a user to the results obtained is not taken into account in the recommendation. However, the relevance of search results in many applications (for example, location-based services) is known to correlate with their spatial proximity to the query sender. In this article, we designed a framework for keyword query suggestions that takes into account location. We propose a weighted keyword chart, which captures the semantic relevance between keyword queries and the spatial distance between the resulting documents and the user's location. The chart is scanned randomly, step by reset, to select keyword queries with the highest scores as suggestions. For our framework to be scalable, we propose a partitioning approach that goes beyond the basic algorithm up to an order of magnitude. The relevance of our framework and the performance of the algorithms are evaluated with real data.

1 INTRODUCTION

Users often have difficulty expressing their search needs on the Web; they may not know the keywords that can retrieve the information they need [1]. The keyword suggestion (also known as a query suggestion), which has become one of the most basic features of commercial web search engines, helps in this direction. After submitting a query by keyword, the user may not be satisfied with the results, so the search engine keyword suggestion module recommends a set of keyword queries that are most likely to refine the search of the user. Effective keyword suggestion methods are based on clicks information from query logs [2], [3], [4], [5], [6], [7], [8], and sessions. [9], [10], [11] or query subject models [12]. New keyword suggestions can be determined based on their semantic relevance to the original keyword query. The semantic relevance between two keyword queries can be determined (i) based on the overlap of their URLs clicked in a query log [2], [3], [4], (ii) by their proximity in a bipartite graph queries and their URLs clicked in the query log [5], [6], [7], [8], (iii) based on their co-occurrence in query sessions [13], and (iv) based on their similarity in the subject distribution space [12]. However, none of the existing methods provide a location keyword query suggestion, so that suggested keyword queries can retrieve documents not only related to the user's information needs, but also located near the location of the user. This requirement emerges because of the popularity of the spatial keyword search that takes a user location and a user-supplied keyword query as arguments and returns spatially close and textually relevant objects for those arguments. Google has processed an average of 4.7 billion queries per day in 20111, a substantial portion of which has targeted local targets and targeted space Web objects (ie, Points of Interest with a web-based presence having locations as well as textual descriptions). documents • S. Qi, D. Wu and N. Mamoulis work in the Computer Science Department of Hong Kong University, Hong Kong 1. http://www.statisticbrain.com/google-searches associated with geo-locations. In addition, 53% of Bing's mobile searches in 2011 were identified as having a local intent.2 To fill this gap, we propose a LKS framework (Suggestions for keyword queries based on location). We illustrate the advantage of LKS by using a toy example. Consider five d1-d5 geo-documents listed in



(a) Documents and Keyword Queries (b) Locations of Documents (c) KD-Graph (d) Euclidean Distances

Fig. 1. LKS Example

Figure 1(a). Each document di is associated with a location di .λ as shown in Figure 1(b). Assume that a user issues a keyword query kq = "seafood" at location λq, shown in Figure 1(b). Note that the relevant documents d1–d3 (containing "seafood") are far from λq. A locationaware suggestion is "lobster", which can retrieve nearby documents d4 and d5 that are also relevant

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

to the user's original search intention. Previous keyword query suggestion models (e.g., [6]) ignore the user location and would suggest "fish", which again fails to retrieve nearby relevant documents. Note that LKS has a different goal and therefore differs from other location-aware recommendation methods (e.g., auto-completion/instant search tag recommendation).

2 LKS FRAMEWORK

Consider a query provided by the user q with the initial entry kq; kq can be a single word or a sentence. Assuming that the sender of the query is at location λq, two intuitive criteria for selecting the right suggestions are: (i) suggested queries (words or sentences) must satisfy the user's information needs based on kq and (ii) suggested queries documents spatially close to λq. The proposed LKS framework takes these two criteria into account. 2.1 Chart Document-Keyword Without loss of generality, we consider a set of geodocuments D such that each document di ∈ D has a point localization di .λ. 3 Let K be a collection of keyword queries from a query log. We consider a weighted bipartite graph directed G = (D, K, E) between D and K and we refer to it as the graphical document keyword (or simply KD-graph). If a document di is clicked by a user who has sent the keyword query kj in the request log, E contains an edge e from kj to di and an edge e 0 from di to kj. Initially, the weights of the edges e and e 0 are identical and equal to the number of clicks on the document di, given the query by keyword kj [2]. As a result, the direct relevance between a keyword query and a clicked document is captured by the edge weight. In addition, the semantic relevance between two keyword queries is captured by their proximity in the G graph (for example, calculated as their RWR distance). All updates in the query log and / or the document database can be easily applied to the KD chart; for a new request / document, we add a new node to the graph; For new clicks, we only need 3. If a document is for multiple locations, we can model it as multiple documents, each referring to a single location. Location-independent documents can also be included in our infrastructure by disabling the location detection component to update the corresponding edge weights accordingly.

3 ALGORITHMS

ALGORITHM 1:

Baseline BA

Input : G(D, K, E), q = (kq, λq), m,

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Output: C

1 PriorityQueue Q ← ∅, C ← ∅

2 Add kq to Q with kq.aink ← 1

3 AINK ← 1

4 while Q 6= ∅ and Q.top.aink ≥1 do

5 Deheap the first entry top from Q

6 tm = the top-m entry from C

7 tm0 = the top-(m + 1) entry from C

8 if tm.rink > tm0 .rink + AINK then

9 break

10 distratio = 1

11 if top is a keyword query node then

12 distratio = 1 − α

13 top.rink ← top.rink + top.aink × α

14 AINK ← AINK − top.aink × α

15 if there exist a copy t of top in C then

16 Remove t from C

17 top.rink ← top.rink + t.rink

18 Add top to C

19 for each node v connected to top in G do

20 v.aink ← top.aink × distratio × w˜(top, v)

21 if there exists a copy v 0 of v in Q then

22 Remove v 0 from Q; v.aink ← v.aink + v 0 .aink

23 Add v to Q

24 return the top-m entries (excluding kq) in C involves retaining α portion of its active ink (line 13) and distributing 1 − α portion to each of its neighbor document nodes based on the adjusted edge weights (lines 19–23). The total active ink AINK is modified accordingly (line 14). As soon as a keyword query node has some retained ink, it enters C. Preparing an archive hub includes conveying the greater part of its dynamic ink to neighboring catchphrase question hubs in light of balanced edge weights (lines 19-23). The calculation restores the hopeful recommendations more prominent than m other than kq in C therefore (line 24 demonstrates the means of BA (for m = 1, = 0.1 and α = 0.5), when connected to the balanced KD chart of our case momentum (see Example 1 and Figures 1 and 2.) The number alongside every hub shows its measure of dynamic ink The numbers in the adjusted rectangles speak to the measure of ink held Initially, a unit

amount of ink is infused into the hub k2, ie the question watchword kq = "fish" gave by the client In the principal emphasis, the hub k2 holds 0.5 amount of ink and conveys 0.5 measure of ink at its neighboring d1-d3 archive hubs as per the balanced edge weights., d3 dispatches its dynamic ink in the measure of 0.325 to its neighbor inquiry ask for hubs k2 and k3 BA closes at the 6th cycle where the dynamic ink of every hub is littler than the proposition k2 "Lobster" , with the biggest measure of ink held (0.098). Segment based Algorithm (PA) The BA calculation can be moderate for a few reasons. To begin with, at every cycle, just a single hub is handled; in this way, the dynamic ink gradually diminishes and the end conditions are met. Also, given the expansive number of emphasess, the overhead of the line Q is critical. At long last, the hubs disperse their dynamic ink to every one of their neighbors, regardless of the possibility that some of them get just a little measure of ink. We take note of that current preprocessing systems that can accelerate the RWR and BCA look (eg, pre-choice of center point hubs) require full information of the chart before the calculation begins. In this way, they are not material to our concern in light of the fact that the edge weights in the Gq diagram rely upon the area of the question, which is obscure ahead of time. The use of a pre-figuring method for all conceivable question areas (i.e., all conceivable Gqs) has outrageous calculation and capacity necessities. To enhance BA's execution, we propose in this segment a segment based calculation (PA) that partitions catchphrase inquiries and KD-G archives into gatherings. Let P K = {P K i} the parcels of the catchphrase questions and P D = {P D i} the segments of the archive. The PA calculation takes after the fundamental routine of the BA calculation, however with the accompanying contrasts: (1) Node-Partition Graphs. Dad utilizes two GKP and GDP situated charts developed disconnected from KD-diagram G and parcels PK and P D. In the GKP chart, an inquiry question hub ki associates with a PD record segment if ki interfaces in G no less than one report in P D. Thus, in the GDP chart, an archive hub dj associates with a PK watchword segment if as of now interfaces in G to no less than one catchphrase inquiry hub ki . By method for instance, in FIG. 4, the record segments are PD 1 = {d 1, d 2} and PD 2 = {d 3, d 4, d 5} and the watchword inquiry segments are PK 1 = {k 1} and PK 2 = {k2, k3}. The edge weights are characterized by the chart Gq, figured amid the execution of PA. Each edge weight appeared in FIG. 4 shows the segment of the ink to be circulated to a segment P from a hub v which is the total of the balanced weights of the edges of the hub v to the hubs of P as indicated by Gq. (2) Ink conveyance. In PA, every hub disseminates its dynamic ink to its neighboring segments (not at all like BA, where every hub conveys its dynamic ink to each of its neighboring hubs). The need line utilized as a part of BA deals with the hubs that will convey the ink, however the need line utilized as a part of PA records the segments that will be handled. The ink got by a segment isn't stretched out to the hubs inside the parcel until the point when that segment achieves the highest point of the need line.

4 CONCLUSION

In this paper, we proposed a LKS structure giving watchword recommendations that are pertinent to the client data needs and in the meantime can recover significant records close to the client area. A gauge calculation reached out from calculation BCA is acquainted with take care of the issue. At that point, we proposed a parcel based calculation (PA) which processes the scores of the hopeful catchphrase inquiries at the segment level and uses a languid system to significantly decrease the computational cost. Experimental investigations are led to consider the adequacy of our LKS system and the execution of the proposed calculations. The outcome demonstrates that the system can offer valuable recommendations and that PA beats the pattern calculation essentially. Later on, we intend to additionally contemplate the viability of the LKS system by gathering more information and outlining a benchmark. What's more, subject to the accessibility of information, we will adjust and test LKS for the situation where the areas of the inquiry guarantors are accessible in the question log. Moreover, we trust that PA can likewise be connected to quicken RWR on general diagrams with dynamic edge weights and we will examine its general materialness later on. In addition, the present form of PA is by all accounts autonomous of the apportioning technique. It is intriguing to research whether elective dividing heuristics can additionally lessen the cost of the calculation.

REFERENCES

[1] M. P. Kato, T. Sakai, and K. Tanaka, "When do individuals utilize inquiry proposal? A question proposal log examination," Inf. Retr., vol. 16, no. 6, pp. 725– 746, 2013. Specialized REPORT,

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF HONG KONG 14

[2] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Question suggestion utilizing inquiry sign in web indexes," in EDBT, 2004, pp. 588– 596.

[3] D. Beeferman and A. Berger, "Agglomerative bunching of a web index question log," in KDD, 2000, pp. 407– 416.

[4] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, "Setting mindful question recommendation by mining navigate and session information," in KDD, 2008, pp. 875– 883.

[5] N. Craswell and M. Szummer, "Irregular strolls on the snap chart," in SIGIR, 2007, pp. 239– 246.

[6] Q. Mei, D. Zhou, and K. Church, "Question recommendation utilizing hitting time," in CIKM, 2008, pp. 469– 478.

[7] Y. Tune and L.- w. He, "Ideal uncommon inquiry proposal with understood client criticism," in WWW, 2010, pp. 901– 910.

[8] T. Miyanishi and T. Sakai, "Time-mindful organized question recommendation," in SIGIR, 2013, pp. 809–812.

[9] A. Anagnostopoulos, L. Becchetti, C. Castillo, and A. Gionis, "An enhancement system for inquiry suggestion," in WSDM, 2010, pp. 161– 170.

[10] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The question stream diagram: Model and applications," in CIKM, 2008, pp. 609– 618.

[11] Y. Tune, D. Zhou, and L.- w. He, "Question recommendation by building term-change charts," in WSDM, 2012, pp. 353– 362.

[12] L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, and M. Kitsuregawa, "A proficient way to deal with recommending topically related web inquiries utilizing shrouded subject model," WWW, pp. 273– 297, 2013.

[13] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu, "Figuring out how to recommend: A machine learning system for positioning inquiry proposals," in SIGIR, 2012, pp. 25– 34.

[14] D. Wu, M. L. Yiu, and C. S. Jensen, "Moving spatial watchword inquiries: Formulation, techniques, and examination,"

ACM Trans. Database Syst., vol. 38, no. 1, 2013.

AUTHOR'S PROFILE:

THOTA.MOUNIKA
PG Scholar, Department of CSE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana, Mail id: mounika.thota2793@gmail.com

MANDALA.SURESH KUMAR
Assiscant Professor Department of CSE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana, Mail ID: mandala.suresh83@gmail.com,