**Location Aware Keyword Query Suggestion Based on Document Proximity**

**ABSTRACT:**

Keyword suggestion in web search helps users to access relevant information without having to know how to precisely express their queries. Existing keyword suggestion techniques do not consider the locations of the users and the query results; i.e., the spatial proximity of a user to the retrieved results is not taken as a factor in the recommendation. However, the relevance of search results in many applications (e.g., location-based services) is known to be correlated with their spatial proximity to the query issuer. In this paper, we design a location-aware keyword query suggestion framework. We propose a weighted keyword-document graph, which captures both the semantic relevance 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 our framework scalable, we propose a partition-based approach that outperforms the baseline algorithm by up to an order of magnitude. The appropriateness of our framework and the performance of the algorithms are evaluated using real data.

**EXISTING SYSTEM:**

* In Existing system 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 m keyword queries that are most likely to refine the user’s search in the right direction.
* However, 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 emerges due to the popularity of spatial keyword search. Google processed a daily average of 4.7 billion queries in 2011,1 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 descriptions) or geo-documents (i.e., documents associated with geo-locations).

**DISADVANTAGES OF EXISTING SYSTEM:**

* Existing keyword suggestion techniques do not consider the locations of the users and the query results; i.e., the spatial proximity of a user to the retrieved results is not taken as a factor in the recommendation.
* However, the relevance of search results in many applications (e.g., location-based services) is known to be correlated with their spatial proximity to the query issuer

**PROPOSED SYSTEM:**

* We propose the first Location-aware Key word query Suggestion framework. We illustrate the benefit of LKS using a toy example. Consider five geo-documents d1-d5 as listed.
* Each document is associated with a location. Assume that a user issues keyword query seafood at location q. Note that the relevant documents d1–d3 (containing “seafood”) are far from q. A location-aware suggestion is “lobster”, which can retrieve nearby documents d4 and d5 that are also relevant to the user’s original search intention.
* That LKS has a different goal and therefore differs from other location-aware recommendation methods (e.g., auto-completion/instant search tag recommendation). Section 5 provides a detailed discussion about the differences between LKS and these models, while in Section 4 we experimentally show that an adaptation of the method is less effective than LKS.
* The first challenge of our LKS framework is how to effectively measure keyword query similarity while capturing the spatial distance factor. In accordance to previous query suggestion approaches LKS constructs and uses a keyword-document bipartite graph (KD-graph for short), which connects the keyword queries with their relevant documents.

**ADVANTAGES OF PROPOSED SYSTEM:**

* This 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.

**SYSTEM ARCHITECTURE:**

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

System                           :         Pentium Dual Core.

Hard Disk                      :         120 GB.

Monitor                         :         15’’ LED

Input Devices                 :         Keyboard, Mouse

* Ram                               :         1GB.

**SOFTWARE REQUIREMENTS:**\

Operating system                    :         Windows 7

.Coding Language           :         JAVA/J2EE

* Tool                                :        Eclips
* Database                         :         MYSQL