Continuous Top-k Monitoring on Document Streams

Abstract

- The efficient processing of document streams plays an important role in many information filtering systems. Emerging applications, such as news update filtering and social network notifications, demand presenting end-users with the most relevant content to their preferences.
- In this work, user preferences are indicated by a set of keywords. A central server monitors the document stream and continuously reports to each user the top-k documents that are most relevant to her keywords.

- Our objective is to support large numbers of users and high stream rates, while refreshing the top-k results almost instantaneously.
- Our solution abandons the traditional frequency-ordered indexing approach. Instead, it follows an identifier-ordering paradigm that suits better the nature of the problem.
- When complemented with a novel, locally adaptive technique, our method offers (i) proven optimality w.r.t. the number of considered queries per stream event, and (ii) an order of magnitude shorter response time (i.e., time to refresh the query results) than the current state-of-the-art.

EXISTING SYSTEM

- In traditional text search, there are snapshot (i.e., one-off) top-k queries over static document collections. The inverted file is the standard index to organize docu-ments .
- It comprises a list for every term in the dictionary; the list for a term holds an entry for each document that contains the term.
- By sorting the lists in decreasing term frequency, and with appropriate use of thresholding a snapshot query can be an-swered by processing only the top parts of the relevant lists.

- Due to the said sorting, we refer to that paradigm as frequency-ordering. This common practice for snapshot queries has been followed by most approaches for continuous top-k search, albeit adapted to the “standing” nature of the continuous queries and the highly dynamic characteristics of the document stream

DISADVANTAGES

- The amount of information made available to users far exceeds their capacity to discover and understand it.

Proposed System

- Our advanced approach (MRIO) outperforms the current state-of-the-art by one order of magnitude.
- MRIO employs novel bounds that offer proven op-timality w.r.t. the number of considered queries per stream event.
- MRIO is more than two times faster than RIO, demonstrating that a skillful adaptation of ID-ordering to CTQDs alone (as in RIO) is not enough to derive the improvements achieved in this work.

- We further improve the performance of MRIO by restructuring its query index (i.e., rearranging the queries inside) to better exploit locality and strengthen the pruning effectiveness of its bounds.

Advantages

- The efficient filtering and monitoring of rapid streams is key to many emerging applications.
- The efficient filtering and monitoring of rapid streams is key to many emerging applications.

HARDWARE REQUIREMENTS

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Floppy Drive	: 1.44 Mb.
Monitor	: 15 VGA Colour.
Mouse	: Logitech.
Ram	: 512 Mb.

SOFTWARE REQUIREMENTS

Operating system	: Windows XP/7.
Coding Language	: ASP.net, C#.net /java

Conclusion

- In this paper, we propose a scalable framework for the processing of continuous top-k queries on document streams (CTQDs).
- A CTQD continuously reports the k most relevant documents to a set of keywords. CTQDs find application in many emerging applications, such as email and news filtering.
- Our preliminary approach, RIO, adapts the ID-ordering paradigm to the CTQD set-ting. An analysis on RIO reveals that the key factor that determines its performance is the number of iterations it executes..

References

- [1] P. Haghani, S. Michel, and K. Aberer, “The gist of everything new: personalized top-k processing over web 2.0 streams.” in CIKM, 2010, pp. 489–498.
- [2] K. Mouratidis and H. Pang, “Efficient evaluation of continuous text search queries,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 10, pp. 1469–1482, 2011.
- [3] N. Vouzoukidou, B. Amann, and V Christophides, “Processing continuous text queries featuring non-homogeneous scoring func-tions.” in CIKM, 2012, pp. 1065–1074.