

A Survey: Searching Techniques

¹Miss Mangala S. Teli, ²Asst. Prof. Priti S. Subramaniam

¹ PG Student (Computer Science & Engg.), S.S.G.B.C.O.E.T., Bhusawal

²Dept. of Computer Science & Engg., S.S.G.B.C.O.E.T., Bhusawal

Abstract: In day to day life as we are using internet mostly for to search every one want answer within short time. Any user who doesn't know how to search can get result by simply typing related words in search engine. They even know which technics are used to search similar data on web. There are many searching technics like Instant search, Type-Ahead Search, Text Proximity Search, Fuzzy Keyword Search Auto-Completion and many more are also available. This paper is about how all mentioned technics works to retrieves results quick and accurate.

Keywords: Instant search, Type-Ahead Search, Text Proximity Search, Fuzzy Keyword Search, Auto-Completion.

I. Introduction

Instant search means as a user type a key to search request is sent to server after each keystroke. This increases query log. This query log is analyzed by calculating success rate by absence of click-through data and user behaviors. But it is not accurate because many users can get information in search result link so doesn't click link. The new success rate calculating method is to calculate number of reformulations in the session, peak queries of the session, results provided by the system[1]. Type-Ahead Search means find match as user type's keyword not only results but top k best results should answer within few milliseconds this is main challenge. There are two popular techniques pruning and materialization. These two techniques are helpful for arranging results in sorted access which improves efficiency [2]. The term proximity has been frequently used to enhance retrieval quality. According to proximity scoring top k results are calculated using algorithms [3]. The fuzzy search can be classified into two categories, gram-based approaches and trie-based approaches. For fuzzy string matching sub-strings of the data are used The second class of approaches index the keywords as a trie, and rely on a traversal on the trie to find similar keywords Each query is a prefix and triet hat can support incremental computation efficiently hence Trie-based approach is appropriate for instant and fuzzy search [5]. In auto-completion, it suggest several possible queries the user can type in next. There have been many studies on predicting queries .Many systems do prediction by treating a query with multiple keywords as a single prefix string. If a related suggestion has the query keywords but not one after the other, then this suggestion cannot be found.

II. Literature Survey

Cetindil, J. Esmaelnezhad, C. Li, and D. Newman proposed how typing behaviors affects search results. Suggested three ways: 1. analyze the query log of an instant-search system, and classify sessions based on various user typing behaviors. 2. A decision-tree-based method in the absence of click data to measure the success rate of an instant-search system. Used the insights obtained from the user-behavior analysis and the returned results of a session to decide according to successful session. Also measures the accuracy of this method by a user study. 3. Proposed methods to estimate the statistics of missing information in a traditional search log to show the benefits of instant search compared to traditional search in terms of user effort, time required to fulfill an information need, and success rate. The comparison showed that instant search can increase the success rate of a search and save 2 seconds and. Query log is analyzed while using Instant search by session boundary detection and handling absence of click data.[1]G. Li, J. Wang, C. Li, and J. Feng type-ahead search on large amounts of data. That is, as a user types in a keyword query letter by letter, we want to on-the-fly find the most relevant (or "top-k") records. One approach is computes their ranking scores to find the most relevant ones after matching record found of those query keywords, and then. This approach is not efficient when there are a large number of candidate answers to compute and store. A list-pruning technique used to improve the performance of sorted access, and the techniques based on list materialization and forward lists for fuzzy search improves performance. These techniques can be easily extended to support large datasets through data partition.[2] R. Schenkel, A. Broschart, S. won Hwang, M. Theobald, and G. Weikum presented techniques that can speedily retrieves one or two orders of magnitude, trading in runtime for low disk space and maintaining the very high result quality of proximity scoring models. Author proposed materialized index structures and precomputed to boost performance. The multiple experiments are carried out over very large text

collection which results in increased in retrieval effectively and efficiently. As compared to a term-based top-k evaluation algorithm achieves an improvement of two orders of magnitude, with a improved result quality with static index pruning for the proximity lists,[3]A. Nandi and H. V. Jagadish solved two main problems one is Fussy Tree structure is created for larger phrases which are larger than word. Second is that a “phrase”, not similar to “word” and does not have a well-defined boundary, also that it is decided by auto completion system not only what to predict, but also how significant phrase that is possible. Also proposed a probabilistically driven multiple completion choice models, to improve the quality of suffix completion exploit some features for example frequency distributions. A new evaluation metric is introduced, TPM, that measures the net benefit provided by an auto completion system much better than the traditional measures of precision and recall. [4]S. Ji, G. Li, C. Li, and J. Feng proposed an efficient incremental algorithm to answer single-keyword fuzzy queries. In “interactive, fuzzy search” system searches the underlying data “on the fly” as the user types in query keywords. Author develop multiple incremental search algorithms using previously cached and computed results to achieve an interactive speed and deployed UC Irvine people directory using these techniques. [5]

III. Related Work

Instant search query log describes how typing behaviors affects search results. There are two methods used to measure success rate that are success measurement and user study .First method works on reformulation of query, peak queries and results by system. The second method works on the feedback given by user for each session. Instant search saves 2 seconds per search and shows result with fewer characters entered to increase success rate. These techniques also reduce typing efforts.

In Type Ahead search while finding top best (top-k) answers there are two algorithms based on access methods- random access and sorted access. In Type ahead while finding top-k answers using sorted access methods list pruning and materialization techniques are used. This technique can be expanded to support fuzzy type-ahead search which allows minor errors between answers and query keywords. Text proximity search make use of top-k query search engine having proximity scoring function for text retrieval. Materialized and precomputed index structures boost performance. This technique can speedily up evaluate by one or two orders of trading in runtime for cheap disk space, magnitude and maintains the very high result quality of proximity-aware scoring models.

Phrase prediction technique decides what to predict and how far to predict so proposed an effective technique for multi-word auto completion. The fussy tree algorithm is introduced to decide prediction quality by considering precision, recall and TPM quality metrics. Auto completion decreases the number of keystrokes typed by up to 20% for email composition and also for developing an encyclopedia entry. In interactive-fuzzy search system searches the underlying data “on the fly” as the user types in query keywords even if user doesn’t know proper spellings. Interactive search algorithm is used to speed up searching. Computes answers on large dataset by using cached results of previous queries and achieves an interactive speed.

IV. Conclusion

We have studied multiple keyword searching techniques such as instant search, Type-head search, Text Proximity Search, Effective Phrase Prediction, and Fuzzy Keyword Search. All these techniques have different processing and make used different algorithms to retrieve top-k results to meet time and space requirements. We have studied here some drawbacks and advantages of searching techniques. Using different dataset number of experiments carried to retrieve result to meet time and space requirement.

References

- [1]. Cetindil, J. Esmaelnezhad, C. Li, and D. Newman, “Analysis of instant search query logs,” in WebDB, 2012, pp. 7–12.
- [2]. G. Li, J. Wang, C. Li, and J. Feng, “Supporting efficient top-k queries in type-ahead search,” in SIGIR, 2012, pp. 355–364.
- [3]. R. Schenkel, A. Broschart, S. won Hwang, M. Theobald, and G. Weikum, “Efficient text proximity search,” in SPIRE, 2007, pp. 287–299.
- [4]. A. Nandi and H. V. Jagadish, “Effective phrase prediction,” in VLDB, 2007, pp. 219–230.
- [5]. S. Ji, G. Li, C. Li, and J. Feng, “Efficient interactive fuzzy keyword search,” in WWW, 2009, pp. 371–380.
- [6]. H. Yan, S. Shi, F. Zhang, T. Suel, and J.-R. Wen, “Efficient term proximity search with term-pair indexes,” in CIKM, 2010, pp. 1229–1238.
- [7]. H. Bast and I. Weber, “Type less find more: fast auto completion search with a succinct index”, in SIGIR, 2006, pp. 671-678.
- [8]. Kaushik Chakrabarti, Surajit Chaudhuri ,Venkatesh Ganti and Dong Xin, “An Efficient Filter for Approximate Membership Checking,” in SIGMOD Conference, 2008, pp. 805-818.
- [9]. Tao Tao, Cheng Xiang Zhai “An Exploration of Proximity Measures in Information Retrieval,” in SIGIR, 2007, pp. 295-302.
- [10]. A. Arampatzis and J. Kamps, “A study of query length,” in SIGIR, 2008, pp. 811–812.