ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Efficient Instant Search by Using Fuzzy Search with Proximity Ranking

Sonali A. Patil[1]
Assistant Professor, Computer Department
BSIOTR, Wagholi
Pune – India

Rupali A. Ingale[2]
M.E.Student, Computer Department
BSIOTR, Wagholi
Pune – India

Abstract: A Instant search is a system which find answer to a query immediately while user enter word character by character. In instant search and fuzzy search it improves the searching results of the system. A fuzzy search is improve the user search experience finding result to keyword like to query keywords. The mean thing in this that each query need to answered fast within millisecond to a achieve an instant response and high query throughput. The fuzzy search is a done by means of a fuzzy matching program system, which returns a list of results the based on a likely relevance even though a search argument words and keyword spellings may not exactly match.

Keywords: Fuzzy Search, Instant Search, Proximity Ranking, Search Engine. Convergent Encryption.

I. INTRODUCTION

Fuzzy matching programs can operate like a word spell checker and spelling-error one changing technique. For example, if a user types "Soory" keyword into Yahoo or Google both of use fuzzy matching technique, a list of hits is along this question "Did you mean sorry?" Alternative spellings and words information that sound the same but are spelled differently given. The fuzzy matching program can be detected for common input typing errors, as well as errors introduced by the optical character recognition (OCR) scanning of the printed documents.

A Search engine helps user to given position of information from large data storage media of content. So proposed the search engine to reduce the work load of the server. The User search keyword in internet its collected the all web pages from server i.e. the linking open data. All the open data are copied temporary into local system because of that the user does not depends on computer to view next web page so it get changed to other form the work amount of server. Local data search is an emerging information-retrieval example in which a system gets answers to a query instantly while a user types in the keywords character-by-character. In that the fuzzy search methods further improve the user search experiences by finding a expected answers in system and filtering keyword to the similar query keyword**.** A main challenge in this search is the high-efficiency requirement so that queries must be answered within short period i.e. in milliseconds to achieve an instant response and a high query. To overcome the space problem and the time limitation problem to fuzzy search method is to a improves the user fetch to get on the point the record in server side it temporarily stored the data in computer and further record search can be done without depending on computer system.

A. Problem Definition

How to integrate proximity information into ranking in instant-fuzzy search to compute relevant answers to the query while achieving efficient time and space complexities?

Solution:

The proximity of matching keywords in answers is an important function to determine the relevance of the answers. User queries typically contain correlated keywords, as well as some pattern based phrases and to answers to these keywords or phrases.

II. Background and Related Work

We are using AES algorithm for encryption/decryption of the data to be uploaded on the cloud storage. From it is clear that AES is more efficient than other algorithms. The time required to complete the process of encryption and decryption is less as compared to other algorithms. So as to achieve the better performance we have choose the AES algorithm. we can use any of the search engine API to implement project like : Bing , Google , yahoo etc. we store the words into database using his id keyword name and count fields, which will be later compare to the output of clusters use K-means clustering to create the similar words documents together. The Output to the clustering is frequent words in one clusters are compare to the previous maintained LOG , and update this log to TLB.

Cetindil∗, J. Esmaelnezhad,Taewoo Kim and Chen Li and Irvine(2014),Studied The integrate proximity information into ranking in instant-fuzzy search while achieving efficient time and space complexities[1].

G. Li, J. Wang, C. Li, and J. Feng(2009), Proposed the methods which is suitable for instant and fuzzy search as each query is a prefix and tries to support incremental computation effectively. A main challenge in this search is the high-efficiency requirement so that queries must be answered within short period i.e. in milliseconds. In this the paper we study how to answer top-k queries in this, i.e., as a user types in a keyword letter by letter, we need to efficiently find the k best answers. In place of inventing completely new algorithms from nothing, we study challenges when taking up having existence top-k algorithms in the literature that heavily be dependent on two basic list-access ways of doing: random access and sorted access[2].

G. Li, S. Ji, C. Li, and J. Feng(2009),Studied how to achieve a high interactive speed for large amount of data in multiple tables, so that a query can be answered efficiently within milliseconds and proposed efficient index structures and algorithms for finding relevant answers on-the-fly by joining tuples in the database.

In this conducted a thorough experimental evolution of the proposed techniques on real data sets to demonstrate the efficiency and practically of this new search paradigm[3].

A. Nandi and H. V. Jagadish(2007),Studied the problem of auto completion not just at the level of single “word” but at the level of a multiple word “phrase” .They have proposed system on predicting queries. Many systems do prediction by treating a query with multiple keywords as a single prefix string. Therefore, if a related suggestion has the query keywords but not consecutively, then this suggestion cannot be found[4].

H. Yan, S. Shi, F. Zhang, T. Suel, and J.-R. Wen(2010),Studied the new index structures based on a term-pair index, and study new document retrieval strategies on the resulting indexes[5].

III. Motivation

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. To improve the efficiency of an instant-fuzzy search system by considering the proximity information system when we need to the compute top-k answers. To adapt existing solutions to solve this problem, including computing the all answers, doing early termination. A technique to index important phrases to avoid the large space overhead of indexing all word grams. The presented by fuzzy search algorithm for finding to the records in a query efficiently, and studied how to compute and rank the segmentations consisting of the indexed phrases. Through observation by giving thought to as the space, time, and relevancy trade-offs of these moves near. In particular, the experiments on true facts showed the doing work well of the proposed technique for 2-keyword and 3-keyword queries that are common in search applications. All the answers for the other queries would give the best performance and satisfy the high-efficiency requirement of instant search. A main challenge in this search is the high-efficiency requirement so that queries must be answered within short period i.e. in milliseconds to achieve an instant response and a high query throughput.

IV. Objectives and Scopes

- To design the front end and store the cumulative results in the back end.
- To develop and design code for various Keyword search.
- To test the system and implement the algorithm for search the data.
- **Faster Searches:** By predicting your search and showing results without opening multiple search pages.
- **Smarter Predictions:** Even when you don't know exactly what you're looking for, the application helps the user to get the actual data which the user need.
- **Instant Results:** The application stores the retrieved records locally and then it searches consecutive times locally without searching the website.

V. Methodology

- The proposed methodology includes the following steps:

1. FindSimilarWord: The FindSimilarWord algorithm is used to find a list of similar words for the given query word with threshold distance "tr". The during processing step of all unique words in the file are represented using data structure system. If the remove distance is within the threshold distance then the word will be added into the result list of similar words.
2. Ranking: The Ranking is used to rank documents based on distance between query keywords. In the entire set of documents which have the all query keywords or words similar to query keywords are the considered for ranking process. It is the important concept.
3. Search Algorithm: This algorithm search, is used to look for on the point documents. As first started preprocessing of query keywords is done, this has to do with taken away of stop words from the keyword list. For each query keyword stemming is done. To get list of similar words to each query keyword threshold distance is worked out based on length of query keyword. Similar words are found using get ready distance. Inverted lists are went across to come to a decision about documents which have within all query keywords. Ranking is applied to find documents with phrases. Documents without groups of words i.e. Documents having in it few query keywords but they do not form a group of words will be identified. The outcome will be put on view as coming together of documents with groups of words and documents without groups of words.

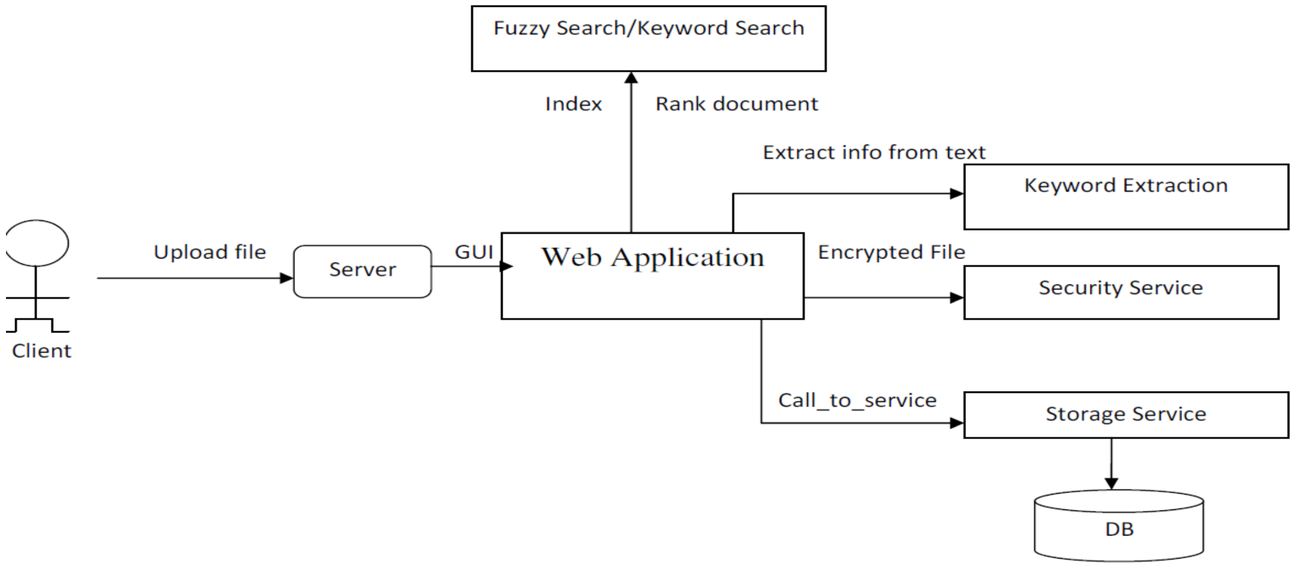

Fig.1.Architecture of Proposed System

The above Fig.1. Represents the architecture of proposed system. The flow and description of each the modules present in the system discussed as below:

1] Web Application This module is responsible for interaction with the web application. This is a client and server software application in which the client system is runs on the web browser. Suppose user is uploading xyz.txt file containing some data pre-processing applying on the file like find the synonyms of words and store it into the list. In this way all the files were stored into dbm_serviceadmin and whenever user type keyword in search box dbm_service called all the time as shown in above fig.1.

2] Security Service This module will handle encryption and decryption of the data. This module is also responsible for hash code generation. In the security service using the AES algorithm for encryption/decryption of the data to be uploaded on the cloud storage

3] Storage Service It simply stores the data. This is the actual physical storage system where only single copy of the data is stored. It also handles all DB operations.

4] Keyword Extraction This module filter words is read textual content word by word and remove the unnecessary words. In the filtering state, delete all commonly using English words. The perform stemming is words like search or searched or searching which all means search. This process is reducing the word to root is called the stemming system.

5] Database In this module store the data on the searching keyword information such as similar word, opposite words and definition related query keyword. By returning the matching files when user searching inputs exactly match the predefined keyword. This type of system enhance the usability of fuzzy me word search. The solution when exact match failure is we exploit and edit distance to qualify keyword similarity.

6] Fuzzy Search/Keyword Search In this module finding the similar word or information. A fuzzy matching programs can operate like a word spell checker and spelling-error one changing technique.

VI. Algorithm Used

1) AES (Advanced Encryption Standard) Algorithm

Steps of AES Encryption Process

Step 1: Byte Substitution (SubBytes)

Step 2: Shiftrows

Step 3:MixColumn

Step 4: Addroundkey

Steps of AES Decryption Process

The process of decryption of an AES cipher text is similar to the encryption process step in the reverse order. Each round consists of the four processes conducted in the reverse order .There are above processes

2) Search Algorithm

Input:

QueryWord // List of query keywords

Output:

Search Result

Steps:

1 Extract stop words from QueryWord

2 Do stemming to each applied QueryWord

3 for each keyword in QueryWord

3 find threshold edit distance

4 similarWord = findSimilarWords(keyword, node, threshold)

5 documents with phrases = Ranking (similar-Word, wordList) .

6 for each keyword

7 find documents without phrases

8 Result = (documents with phrases) union (documents without phrases).

3) Convergent Encryption

Convergent encryption provides data confidentiality in de-duplication. A user (or data owner) derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user derives a tag for the data copy, such that the tag will be used to detect duplicates. A convergent encryption scheme can be defined with four primitive functions:

1. KeyGen(M) -> K is the key generation algorithm that maps a data copy M to a convergent key K .

2. Encrypt(K,M) -> C is the symmetric encryption algorithm that takes both the con-vergent key K and the data copy M as inputs and then outputs a cipher text C .

3. Decrypt(K,C)-> M is the decryption algorithm that takes both the cipher text C and the convergent key K as inputs and then outputs the original data copy M .

4. TagGen(M) -> T(M) is the tag generation algorithm that maps the original data copyM and outputs a tag T(M). We allow TagGen to generate a tag from the corresponding cipher text by using T(M)=TagGen(C), where C=Encrypt(K,M).

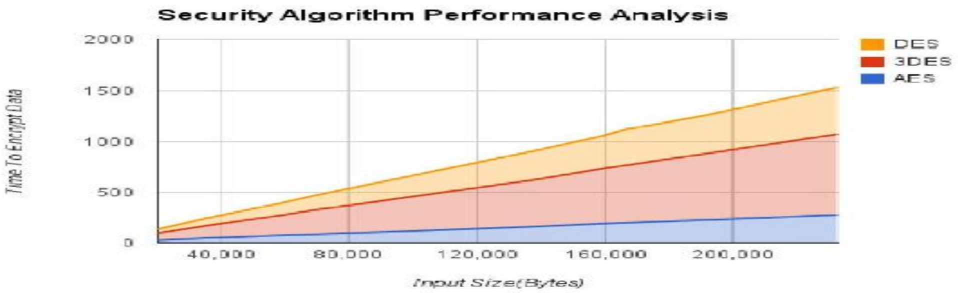

Fig.2.Performance Analysis for Security Algorithm

VII. Result

In this section we represent the result of practical work. The above diagram shows the performance analysis for the In this section we represent the result of practical work. The above diagram shows the performance analysis for the security algorithms. We are using AES algorithm for encryption/decryption of the data to be uploaded on the database storage system. From Fig.2 it is clear that AES is more efficient than other algorithms. The time required completing the process of encryption and decryption is less as compared to other algorithms. So as to achieve the better performance we have choose the AES algorithm.

The search query is the high-efficiency requirement so that queries must be answered within short period i.e. in milliseconds to a achieve an instant response and a high query throughput we have choose the search algorithm fig 3,fig 4 and fig 5 is a graphical form of the data encryption, decryption and search query and In search query engine enter query get result file name, download key no, query and rank count display.

VIII. Conclusion

We studied how to improve ranking of an instant-fuzzy search system by considering proximity information. We studied how to adapt existing solutions to solve this problem, including computing all answers, doing early termination, and indexing term pairs. We proposed a technique to index important phrases to avoid the large space overhead of indexing all word grams. We presented a security algorithm to get the keyword set. We compared our techniques to the instant-fuzzy adaptations of basic approaches. We conducted a very thorough analysis by considering space, time, and relevancy tradeoffs of these approaches. The time required completing the process of encryption and decryption is less as compared to other algorithms. So as to achieve the better performance we have choose the AES algorithm.

Acknowledgement

I take this special opportunity to express my sincere gratitude towards professor and all the people who supported me during my project work. I would like to express my gratitude to the P.G. Coordinator Prof. A. C. Lomte for providing special guidance. I would also like to thank our HOD Prof. G. M. Bhandari and our Principal Prof. Dr. T. K. Nagaraja who always has enough time to solve everyone’s problems at hour of the day. Finally thanks to all teachers who are always supportive at us.

References

1. Cetindil∗, J. Esmaelnezhad,Taewoo Kim and Chen Li and Irvine, "Efficient Instant-Fuzzy Search with Proximity Ranking",Available: http://www.imdb.com, as of July 2013.
2. S. Ji, G. Li, C. Li, and J. Feng, “Efficient interactive fuzzy keyword search,” in WWW, 2009, pp. 371–380.
3. MS.Rupali A.Ingale,Prof.Sonali A.Patil, "User Based Instant Search Mechanism By Using Fuzzy Search And Proximity Ranking" Fifth Post Graduate Conference Of Computer Engineering cPG-CON 2016, 25 -26 March 2016.
4. Ms.Rupali A.Inagale,Prof Sonali A.Patil," User Based Instant Search Mechanism By Using Fuzzy Search And Proximity Ranking "International Journal Of Innovative Research In Computer And Communications Engineering(IJIRCCE),Volume 4,Issue 5,May 2016,ISSN(Online):2320 9801.

5. G. Li, S. Ji, C. Li, and J. Feng, "Efficient Type-Ahead Search on Relational Data: A Tastier Approach," Proc. ACM SIGMOD Int‟ l Conf. Management of Data, pp. 695-706, 2009.
6. A. Nandi and H. V. Jagadish, "Effective phrase prediction," in VLDB,2007, pp. 219–230.
7. 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.
8. Cetindil, J. Esmaelnezhad, C. Li, and D. Newman, "Analysis of instant search query logs," in WebDB, 2012, pp.7–12.
9. C. Silverstein, M. R. Henzinger, H. Marais, and M. Moricz, "Analysis of a very large web search engine query log," SIGIR Forum, vol. 33, no. 1, pp. 6–12, 1999.
10. Vaishali D.Salunkhe Singh and Deepak Gupta,"Survey on Efficient Instant-Fuzzy Search with Proximity Ranking" Available:http://www.ijarcsse.com,as December 2014.
11. H. Bast and I. Weber, "Type Less, Find More: Fast Auto completion Search with a Succinct Index," Proc Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 364-371, 2006.
12. S. Chaudhuri, V. Ganti, and R. Motwani, "Robust Identification of Fuzzy Duplicates," Proc. 21st IEEE Int'l Conf. Data Eng., pp. 865- 876, 2005.

AUTHOR(S) PROFILE

Prof. Sonali Appasaheb Patil, received the degree of M.Tech in CSE from JNTU University in 2013 and currently pursing Phd from BSAU,Chennai and Presently working as an Assistant Professor in JSPM'S BSIOTR,Savitribai Phule Pune University.

Ms. Rupali Annasaheb Ingale, received the degree of B.E.Computer Engineering from BVCOE and RI Savitribai Phule Pune University in 2013 and currently pursuing Master of Engineering from JSPM'S BSIOTR,Savitribai Phule Pune University.