Browser Extension for Top-k Instance Retrieval

Mr. Mali S. R[#1], Ms. Prajakta Shaha[#2], Nilam Patil[#3], Shaheen Shikalgar[#4], Shinde Megha[#5]

[#1]Assistant Professor, [#2]Student, [#3]Student, [#4]Student, [#5]Student
Computer Science and Engineering, Adarsh Institute of Technology & Research Centre, Vita;
Shivaji University Kolhapur

Abstract — *Cloud computing has appearing as a pattern for data outsourcing with high quality data services. This concept represents one of the most important shifts in information technology many of us are likely to see in our daily life. Cloud computing has rising as a encouraging pattern for knowledge outsourcing and prime quality knowledge services. Using Cloud Storage, users can remotely store their data and enjoy on-demand high quality applications and services. Cloud computing gives the services for on demand users. On demand services distribution perform with outsourcing operations. This paper is deal with information extricate from top-k web pages, which are web pages that describe top k instances of a topic which is of general interest. Examples include "the 10 flowers in the world", "top 10 cricketers in India" etc. In this paper, we present a systematic method that extracts top-k lists from web pages with high performance. Cloud computing is the technology, in which storage and return of sensitive data information are increased in usage.*

Keywords —
Google Search Engine API, HTML Parser, Levenshtein Algorithm.

I.Introduction

Cloud computing, a critical pattern for advanced data service, has became a necessary feasibility for data users to outsource data. Cloud computing offer the services for on demand users. On demand services distribution perform with outsourcing operations. Before exchanges the info file should it write in code and store in cloud server. In cloud computing, data owners may share their outsourced data with a number of users, who might want to only retrieve the data files they are interested in. One of the most popular ways to do so is through keyword-based retrieval.

The main threat on data privacy roots in the cloud itself. When users outsource their private data onto the cloud, the cloud service providers are able to control and monitor the data and the communication between users and the cloud at will, lawfully or unlawfully. To ensure privacy, users usually encrypt the data before outsourcing it onto cloud, which brings great challenges to effective data utilization.

To get the particular data retrieval, the large amount of documents insists the cloud server to perform result significance ranking, rather than returning results which are not similar. Such ranked of the search system available the data users to find fastly the most accurate information, rather than sorting all the way through each and every match in the content set. On the other hand, to progress the search result exactness as well as to increase the experience of user searching. To provide more exactness to the end Privacy Conserving In cloud Documents in excess of uses result is done by searching, the unlabelled data keywords are incorporated in the index of the server and then searching is done this search results is then characterized and then they are sorted in their splitting up using Top k query algorithm. TOP-k selection queries will assist to sort the related data and provide the accurate data to the end user.

This paper is organized into four sections. Section I depicts about the introduction of top k instances and UML modeling language. In Section II, describes the literature about the searching features. In Section III, describes technique in the form of Levenshtein algorithm implementation on the top k instances is described with the help of UML class model. A small but an effective Class model is proposed and implementation of Levenshtein algorithm on the top k instances is represented. Al last, section IV conclude the research work

II.Literature Review

When client is requesting for the any required query to the google or any other search engine it gives bulk amount of data. Here we get number of urls and unnecessary data links. So there is large amount of time lost. They create a own data structure but don't database.

We get many identical urls which containing same type of data and that's why it required large amount of time for searching the

information. In existing method the retrieval result is based on user-click event or ranking based process for every search
Draw Backs of Existing System

- Accurate data we are not getting.
- Users will not get adequate required ranking functionality.

Data that can be sharing will not be safe.

III.Relevance of the Work

Now days when client is requesting for the any required query to the google or any other search engine it gives bulk amount of data. Most of the time we cannot reach at the proper result which we want. To get the appropriate result we remove dust for example multimedia url, canonical identical url. Then run distance calculation and data sort by the distance by using Levenshtein distance algorithm. And gives the information in the tabular form.

If there is no any information in table form on the crawler then download the closest page as per distance and display. This project is related to searching that's why there possibility of failure, here the accuracy of searching is 75%. It helps to reduce time for searching information from the crawler. The url extraction from raw data, remove dust, Parsing, browser extraction etc is the different process.

In this project we consume time for searching queries required to the client and reaches at accurate result which is in the tabular form. This search engine is user friendly. Fastest performance due to web minner.

IV.Proposed Work

We propose a method for Top-K retrieval, in this method user search query & keyword is same as already they searched query & keyword means the already selected content click will displayed in the top link and then all other ranking based links are display. We firstly create custome search engine (web crawler) by using google's APIs. We get large amount of raw data in the form of urls from google custome APIs. By using JSON parser we extract this urls and titles from raw data as per required search query. We got many useless urls where there is no any chances of getting tabular information as we required so that we remove this type of dust like multimedia urls and canonical identical urls.
After that we measure distance of urls by running Distance Calculation Algorithm and sort that urls according to maximum closest distance. By using Levenshtein Distance algorithm we compare search query entered by user and titles which we got from google custome search APIs. By running web minner we will find urls consist of tabular form of related search query. If urls containing tabular information the minner shows that result page else it will directly shows closest page related to that search query according to distance travelled by that urls. To give easy use of our browser to the end user we give browser extension on Google chrome. It is user friendly and easy as well as fastest to use.

V.Proposed Methodology

In existing method the retrieval result is based on user-click event or ranking based process for every search. We propose a method for Top-K retrieval, in this method user search query & keyword in which query is same as already they searched query and keyword means the already selected content click will displayed in the top link and then all other ranking based links are display.

We firstly create custom search engine (web crawler) by using google's Custom search APIs. We get large amount of unstructured raw data including webpage metadata. By using JSON parser we extract this urls and titles from raw data as per required search query.
We got many useless urls where there is no any chances of getting tabular information as we required so that we remove this type of dust like multimedia urls and canonical identical urls.

After that we measure distance of urls by running Distance Calculation Algorithm and sort that urls according to maximum closest distance. By using Levenshtein Distance algorithm we compare search query entered by user and titles which we got from google custom APIs.

By running web minner we will find urls consist of tabular form of related search query. If urls containing tabular information the minner shows that result page else it will directly shows closest page related to that search query according to distance travelled by that urls. To give easy use of our browser to the end user we give browser extension on Google chrome. It is user friendly and easy as well as fastest to use.

VI.HW /SW Requirement

H/W Requirement:

Processor	: core i5 (And Above)
RAM	: 512 MB (min)
Hard Disk	: 4 GB
Key Board	: Standard Windows Key Board
Mouse	: Two or Three Button Mouse

S/W Requirement:

Windows 7/8/10.

Visual Studio 2012.
C# 4.0.
LINQ.
JQuery.
HTML 5 and CSS 3.
Google Custom Search API.

VII. System Architecture

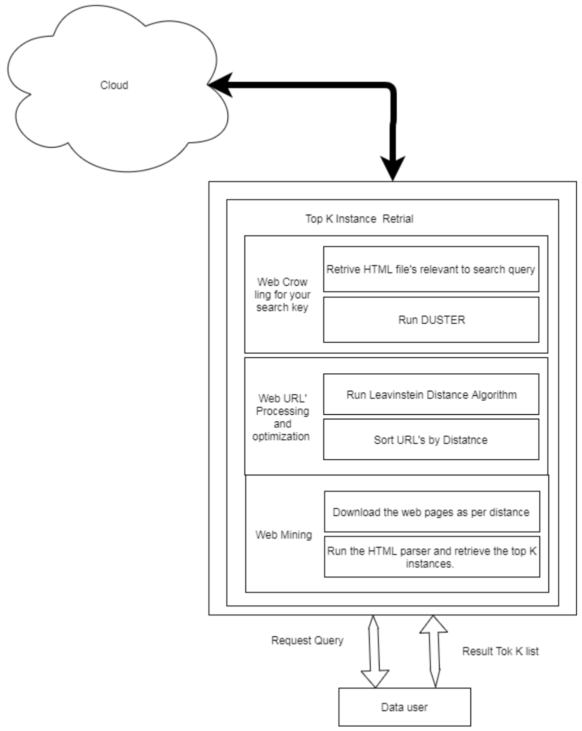
Cloud
Top K Instance Retrial
Web Crow ling for your search key
Retrive HTML file's relevant to search query
Run DUSTER
Web URL' Processing and optimization
Run Leavinstein Distance Algorithm
Sort URL's by Distatnce
Web Mining
Download the web pages as per distance
Run the HTML parser and retrieve the top K instances.
Request Query
Result Tok K list
Data user

Fig: System Architecture.

VIII. Future Scope

To improve the accuracy upto 100% we can modify the algorithm for giving accurate result. We parse for specific search next we searching for dew. Search engines are not only for searching text. At their heart, search engines are all about quickly and efficiently filtering and then ranking data according to some notion of similarity. By using this project we can fastest retrieve data and search the information which is in top list. It is user friendly and easy as well as fastest to use.

IX.References

1. Lakshmi k pillai, P.Saranya ,“Towards Secure Top-k Multi keyword Retrieval over encrypted cloud data”.
2. C.Rajeshkumar, Dr.K.Rubasoundar ,“Retrieval of Encrypted cloud data using multikeyword.”
3. Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou, “Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data”
4. Raghavendra S, Geeta C M, Rajkumar Buyya, Venugopal K R, S S Iyengar, MSIGT: Most Significant Index Generation Technique for Cloud Environment”
5. Jaikishan Tindwani, Aruna Gupta, “A Survey on Multi-Keyword Ranked Query Search over Encrypted Cloud Storage”
6. Priyanka D. Deshmukh, Prof. Prashant Jawalkar, “A Survey on Privacy-Preserving Multi Keyword Ranked Search Over Encrypted Cloud Data Using Hybrid Encryption”
7. Prof. C. R. Barde, Pooja Katkade, Deepali Shewale, Rohit Khatale, “Secured Multiple-keyword Search over Encrypted Cloud Data.”
8. “Secure and Efficient Retrieval of Data in Cloud”, Akshay D. Kapse, Piyush K. Ingole.