International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Efficient Top-k List Extraction from the Web

Jyoti H. Yadav[1], M. M. Deshpande[2]

A. C. Patil College of Engineering, Kharghar, Navi Mumbai, India.
Department of Computer Engineering, University of Mumbai

Abstract: ***In today's technological world the largest source of getting the information is World Wide Web. But most information i.e. present on web is of different format for e.g. it may be structured, non-structured. So when user enters any query or top-k items query on search engine he gets many URL linking as result which is time consuming to fetch every URL. So to avoid above problems we present our proposed new technique for extraction of top-k list. It will give exact result which user wants in less time. It describes top-k instance of general interest.***

Keywords: Top-k list, Title extract, Dust removal, Parsing.

1. Introduction

Today, huge source of information is WWW. To extract useful information on web is known as mining web. It divides into two formats: i) Structured formatted information extraction ii) Natural language text extraction. The first form is available mainly in hypertext markup language or extensible mar up language. Web tables are presented in large numbers in entire corpus, but only less amount of information is exact as per users' expectation.

Top-k list data is so much valuable and accurate.It is cleaner and having high quality.According to criteria the ranking is done on it.So ranked data produces as a result.

Top-k page title gives the clear context.So page can be easily interacted and extracted.Some common examples of top-k title are:
10 most influential researchers alive today are.
10 most famous android games.
10 blockbuster movies in 2015 in India.
Top 10 wild animals in India.

To obtain our goal of top-k list, when user enters query in search engine, raw data extracted from web. From that top-10 titles are extracted. From those titles we remove dust contents such as facebook posts, videos, linked in posts etc. After that operation by using edit distance algorithm we calculate score for every title matches with query title and arranges that score list in descending order. From remaining pages parsing is done by using HTML parser; extract data and displays expected result if presented in tabular form. If those pages does not contain any tabular expected result, then parser display closest page result according to search query and score calculation.

2. Existing System

Many methods have been reported in the literature to extract lists or tables from the web. None of them considers top-k list extraction. Some of the methods extracts data from web tables or list based on very specific tags <TABLE>, <OL>, <UL>,<DL> which are list-related [1]. Some methods include extraction of data records which are of same type based on the DOM trees similarity. These approaches are inflexible because of the inconsistent and dynamic nature of web pages. More recently, several techniques have attempted to use visual information in HTML in information extraction. Miao et al [4] introduces the research performed for mining contiguous as well as non-contiguous information records. Main concept used- tag path, also visual signals. First visually repeating information detected from that data records fetched. However, these techniques promiscuously extract all elements of all tables or lists from a web page, therefore the goal is different from that of this work which is to get one specific list from a page while filtering all other lists as noise. Due to above reasons our proposed system focuses on specified important data on web which we can get with the help of top-k list. In this all similar manner information within top-k list are grouped together using proper context. The main features of top-k list are:

- Top-k data on web is large and rich.
- As compared to other forms of data it is cleaner means having maximum quality.
- With respect to conditions described in title of top-k page, attributes are ranked.
- This data is important because each list contains context that can interpreted by user.

3. Problem Statement

In today's technological world the largest source of getting the information is World Wide Web. But most information present on web is of different format for e.g. it may be structured, unstructured. Extracting specific information from such data is very difficult. Therefore, we present here our proposed method about information extraction from top-k web pages, which describes the top-k instance of which is of general interest.

4. Proposed System

The proposed scheme meets all the requirements listed below

4.1 Design Requirements

1) **Public verifiability:** To allow proposed scheme to verify the correctness of data on demand without

retrieving the entire data or without introducing additional contents to the users.

2) **Correctness:** To obtain result which is correct.
3) **Time factor:** to obtain result in as minimal time as possible.

4.2 Working of System

The proposed scheme meets all the requirements listed above. To achieve this target, we have used two basic entities; they are data server and data user. We use the concept of Google Custom Search. Google Custom Search enables you to create a search engine for your website, your blog, or a collection of websites.

4.3 Implementation Details

The proposed scheme is designed to extract top-k list data which is rich and ranked. Figure shows the architecture for the proposed extraction system.

1. **Role of data server**: Data server provides requested query data in the form of html raw files. Here, the concept of Google custom search is used. For data retrieval we use Google server.

Algorithm 1: **Working of data server**
Input : Top-k query
Output: HTML raw data

2. Role of data user: User request top-k data from server by inserting top-k query in search engine. As a result, from server user will get raw html files on which user has to perform following steps-

1) Extraction of web URLs and its titles: From raw data user will fetch first ten URLs with its titles.
2) Remove dust from web URLs: From extracted URLs user will remove dusting contents which includes you tube videos, Facebook posts, linked in posts, other video sites etc. This DUST information can be created for number of reasons. For removing this DUST we created one dust list that list contains Facebook comments, twitter comments, audios, videos, YouTube, urls. By using this Dust algorithm, when user searches query then they get exact result removing the dust urls. To use this algorithm user can get its proper result within less time.

3. Run levenshtein distance algorithm: After removing dust user implement edit distance algorithm on remaining titles and calculates score on the basis of best matching title with query title and according to that score sorting is performed in descending order.

Algorithm 2: Levenshtein edit distance algorithm
Input: Page title and Query title
Output: Matching Score List in Descending Order
Steps-
(a) Page titles are matches with query title one by one
(b) Matching score is calculated one by one
(C) According to score calculated list is arranged in descending order
(d) Final score resulted in descending order displays

4. Run Html parser: After sorting, user will create folder in C drive as temphtml.Html parsing will be performed on listed links. Parsing performed to check whether any table is available inside the pages. If yes then it will be displays as output to the query. If there is no any table then user will perform final step.

5. Get closest result: If table is unavailable in previous step then user will directly fetch and extract closest result present in top URL according to score calculated.
So, finally user will get his top-k query result.

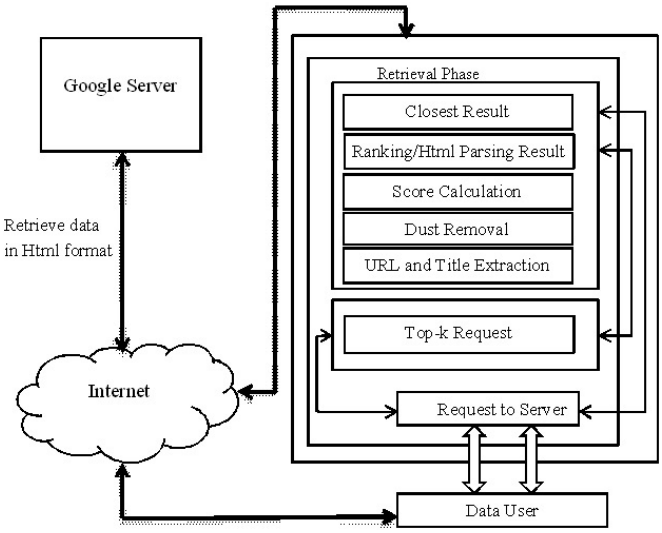


Figure 1: Efficient Top-k List Extraction System

Algorithm 3: Working of data user
Input: Top-k query
Output: Top-k query result
Steps-

1) User enters top-k query in search engine.
2) User extracts titles from raw html data obtained from server.
3) User removes dust content from list.
4) User performs edit distance algorithm and calculates score.
5) User performs parsing to find out table.
6) If table is not present then user displays closest result page.

5. Experimental Result

Input to the system is a web page. It follows all steps as explained before and resulted in top-k listed table , if table is not present then closest result page displays as output. We conducted these experiments on a 4GB RAM PC and 2.70GHz Dual-Core Intel CPU.

Figure 2: User Enters Query

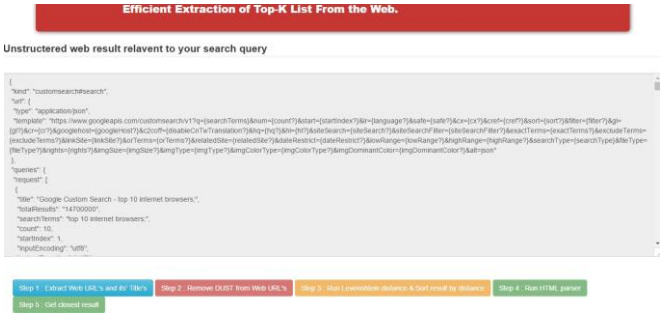


Figure 3: Extracts Raw Data

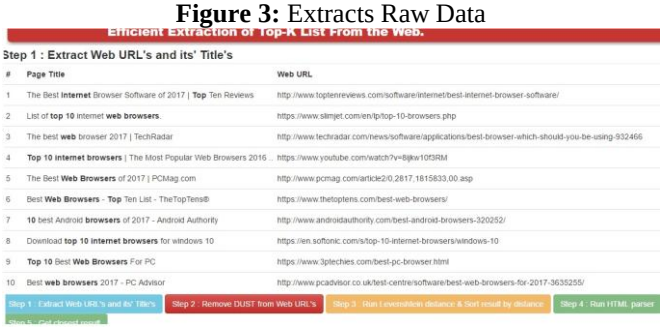


Figure 4: Title Extraction from Raw Data

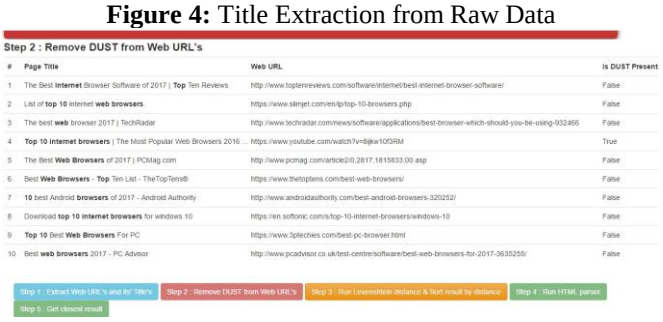


Figure 5: Dust Removing from Title

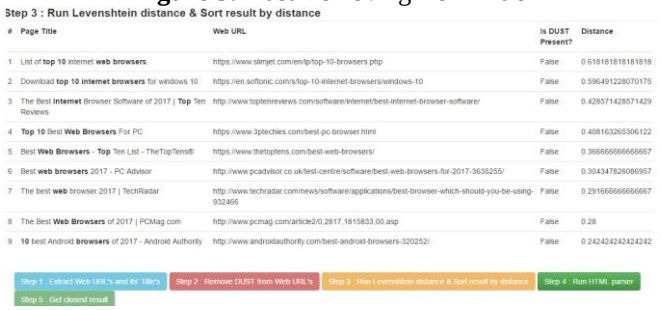


Figure 6: Score Calculation by Edit Distance

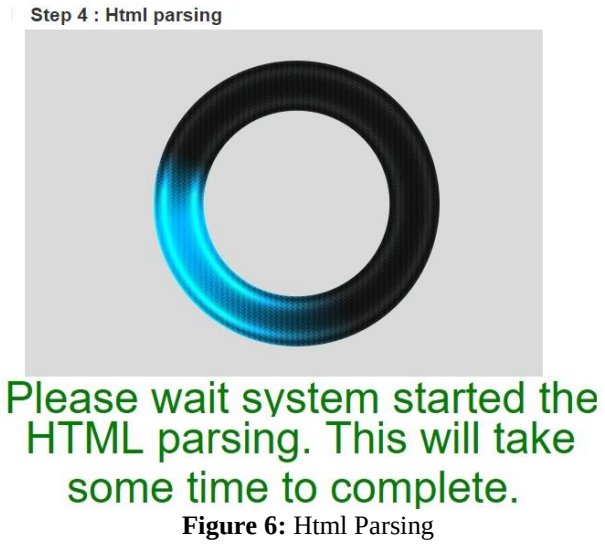
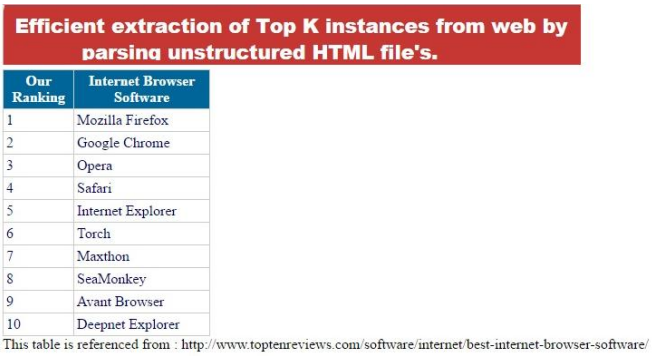


Figure 6: Html Parsing



Our Ranking	Internet Browser Software
1	Mozilla Firefox
2	Google Chrome
3	Opera
4	Safari
5	Internet Explorer
6	Torch
7	Maxthon
8	SeaMonkey
9	Avant Browser
10	Deepnet Explorer

This table is referenced from : http://www.toptenreviews.com/software/internet/best-internet-browser-software/

Figure 7: Result Table

If table is not there in top pages then message displays as expected result can't be parsed so go for final step in which closest result page displays as a result.

6. Conclusion

The proposed system top-k list is easy to appreciate, having high quality and it is more interesting for the human utilization in less time. It overcomes all the limitations of the existing systems. It verifies the correctness of data on demand without retrieving the entire data or without introducing additional contents to the users. Results can be obtained in minimum time. The top-k list data result is exact and cleaner. The result displayed only for the queries which are of general interest, which gets maximum hits, common discussion topic on social media. Minor data or queries get ignored .The system should be able to display result for such queries.

7. Acknowledgement

I take this opportunity to thanks and express my profound gratitude and deep regards to my guide Dr. M. M. Deshpande (H.O.D Computer Engineering Department) who gave me the inspiration to pursue the project and guided me in this endeavor. My profound sense of gratitude towards our Principle Dr. D. G. Borse for his valuable guidance and for providing us all necessary facilities to carry out the project successfully. I am obliged to staff members of A. C. Patil College of Engineering, for their support and cooperation.

References

[1] S. Tong and J. Dean System and methods for automatically creating lists, US Patent: 7350187, Mar 2008.
[2] M. J. Cafarella, E. Wu, A. Halevy, Y. Zhang, and D. Z. Wang, Web tables: Exploring the power of tables on the web, in VLDB, 2008
[3] R. Gupta and S. Sarawagi. Answering table augmentation queries from unstructured lists on the web. PVLDB, pages 289300, 2009.
[4] B. Liu, R. L. Grossman, and Y. Zhai, Mining data records in web pages, in KDD, 2003, pp. 601606.
[5] G. Miao, J. Tatemura, W.-P. Hsiung, A. Sawires, and L. E. Moser, Extracting data records from the web using tag path clustering, in WWW, 2009, pp. 981990.

[6] J.Kowsalya, K.Deepa, Extracting and Aligning the Data Using Tag Path Clustering and CTVS Method International Journal of Advanced Research in Computer Engineering Technology (IJARCET) Volume 2, Issue 4, April 2013.
[7] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: automatic data extraction from data intensive web sites. In SIGMOD, pages 624624, New York, NY, USA, 2002. ACM.
[8] R.C.Wangand , W.W.Cohen. Language-independent set expansion of named entities using the web. In ICDM, 2007.
[9] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, and B. Pollak, Towards domain independent information extraction from web tables, in WWW. ACM Press, 2007.
[10] D. Cai, S. Yu, J. R. Wen and W.Y. Ma. Extracting content structure for web pages based on visual representation. In APWeb, pages 406417, 2003.
[11] F.Fumarola, T.Weninger, R.Barber, D.Malerba, and J .Han, Extracting general lists from web documents: A hybrid approach, in IEA/AIE (1), 2011, pp. 285294.
[12] Hu Yunhua, Xin Guomao, Song Ruihua, Hu Guoping, Shi Shuming, Cao Yunbo and Lim Hang. Title extraction from bodies of html documents and its application to web page retrieval, Microsoft Research Asia, SIGIR 05, August 15-19, ACM, 2005.
[13] Solomon, Matthew and Yu, Cong and Gravano, Luis. Popularity-guided top-k extraction of entity attributes, Columbia University, Yahoo! Research, Web DB 10, ACM, 2010.
[14] Z. Zhang, K. Q. Zhu, and H. Wang, A system for extracting top-k lists from the web, in KDD, 2012.