ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017)

AGTI's Dr. Daulatrao Aher College Engineering, Vidyanagar Extension, Karad

Vol. 4, Special Issue 4, January 2017

Efficient Extraction of Top-k Instances from Web

Prof. Sayali Shinde[1], Tejaswi Shewale[2]

Prof, Computer Science, DACOE, Karad, India[1]

Student, Computer Science, DACOE, Karad, India[2]

Abstract: Finding proper information from web pages is very difficult. Because we face problems such as most of available data contains unnecessary information such as some product advertisements, Facebook or twitter posts. One more problem is, obtained data is not in structured format. To overcome these problems, we introduce a system which mainly focuses on extracting exact information in top-k list format. List data is very eventful source to retrieving information. This paper work on information extraction from top-k web pages which contains top-k instances for open domain knowledge based. For example- "Top 10 IT companies in India". As compare to structured information from web, Top-k list data is cleaner and ranked. Top-k data has interesting semantics. We propose a system which gives direct top-k list when user enters a search query within minimum time. Extraction of top-k list depends on 1] Extracting web URLs and its titles 2] Removing dust from web URLs 3] Using extraction algorithm extract exact top-k list.

Keywords: Top-k list, structured data, data extraction, DOM parser, top-k web pages.

I. INTRODUCTION

Recently, World Wide Web (WWW) is voluminous source of information. Information on web is in the form of structured and unstructured data. Extraction of data from the web pages is called web mining. Extraction of data unstructured data is very difficult structured data is in the form of HTML and XML language which contains tag such as <ul>, <li>, <table>. [2] [4] [5]
How we know the extracted data from list and tables is valuable or not. The quantity of data available on web is dilated. But most of data is worthless and very small amount of data interpretable without context. Web contains large number of tables and most of them are not relational tables. It is easy to interpret relational tables. In relational table we consider rows as an entity and column as attribute of an entity.

Suppose we have a extracted table which contain 3 rows and 3 columns of names "mobile", "company", "price" respectively. Still we can't understand why these 3 companies are grouped together (e.g. are they have good storage capability, maximum battery backup, famous companies, same feature) how we should interpret their price. From this we can't understand which context of the information is helpful for us. Understanding of context is essential for interpretation. But most of time the context is in natural languages or unstructured text which cannot be interpreted by machine. So that we focus on context which we can easily understand and use that context for information extraction.

Proposed system takes top-k pages from web which is rich source of information. This system work for getting result in top-k list from web which contains expected result. Top-k list is very high quality and cleaner. Information in top-k list has interesting semantics. Some examples of top-k list are-

- Top 20 dangerous animals in India.
- Top 15 tallest mountains in world.
- 10 best players of cricket.

There top-k pages consist three important fragments-
1) The value of k. e.g. 15, 10 as shown in above examples which indicate number of items.
2) A topic or concepts. Example- animals, mountains, players.
3) Ranking Criterion. Example-dangerous, tallest, best.

There are two additional fields' time and location which are optional.
Today's technology is much faster and advanced. On the web, when user enter search query, user gets lots of links from search engine. Users have to go and check first link. If users get proper result then search is stopped. Otherwise, users have to go in second link to check whether it has expected result or not. If it has proper information then search stopped. This process is continued until user get exact result. This entire process is very time consuming and lengthy.

Due to this reason the system is focusing on rich and precious data from web that we get from top-k list. Therefore we get top-k pages for information extraction. We choose top-k list for data extraction for following reasons-

1. Top-k information is cleaner than different types of information on the web. As we know a large portion of the information on the web is in free content, and free content is difficult to decipher. Web tables are organized, however just a little rate of them contain important and helpful data. Interestingly, Top- k information is much cleaner.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017)
AGTI's Dr. Daulatrao Aher College Engineering, Vidyanagar Extension, Karad
Vol. 4, Special Issue 4, January 2017

2. Top- k information is ranked. Not at all like web tables, have which contained a set of items, item in a top-k list generally ranked by the title of the top-k page. Ranking is essential in information extraction.
3. Top- k information has interesting semantics. One reason why Top-k information is profitable is on account of every list has a context we can interpret, and the context is normally an interesting one.

II. LITERATURE SURVEY

1. Automatic Extraction of Top-k Lists from the Web
Zhixian Zhang, Kenny Q. Zhu, Haixun Wang, Hongsong Li [1] this paper work on extracting information from top k web pages, which contains top k instances of an interested topic. The method used in this paper improved performance by lists and focusing on the context. It doesn't concentrate on the visual region of the lists.
In the event that list is separated into more than one page it may not get included totally. Author exhibited algorithm that concentrates such top k records from the web snapshot and structure of every list was found.

2. Extracting data records from web using tag path clustering
G.Miao, J.Tatemura, W.P.Hsiung, A.Sawires, L.E.Moser [2] author presents strategy for extraction of records that gives list of elements based on analysis of web page. Work of this paper is motivated by experience in developing an automatic information extraction component of a system. The method concentrate on how distinct tag path appears in web document DOM tree.

Every data record contains different data attributes. There is no one to one mapping between HTML code structures to data record arrangement. Identification of data attributes offers the capability of better utilization of web data.

3. A System for Extracting Top-K Lists from the Web
Zhixian Zhang, Kenny Q. Zhu, Haixun Wang[3] defines extraction of general lists and tables from the web. It is based on recognize, extract and understand top-k list content from web pages. This paper is different from data mining jobs, because compared to structured data as top-k list is easy to understand, clear and hence interesting semantics.

Professional base is an example of general purpose knowledge. It is moreover conceivable to build an research engine for "top-k" lists as a strong truth addressing machine. 4 stage extraction systems have illustrated its capacity to recover large number of "top-k" records at a high accuracy.

4. Web Tables: Exploring the Power of Tables on the Web
Michael J. Cafarella, Alon Halevy, Zhe Daisy Wang[4] author exploring the power of tables on the web concentrate on separating the information from the web tables or tables that depend on certain or related tags yet, the limitation here is that the information in the web table may not contain context and subsequently context might be disregarded. The particular related tags can be <UL>, <OL> and <TABLE>. MDR i.e. mining information records from the website pages.

III. PROBLEM DEFINITION

Information extraction from top-k web pages for open domain knowledge based. Web page contains pair of (t, d) where t is a title of page and d is HTML body of page.
A page (t, d) is a top-k page only if:

1) Title t of the web page contains five tuples (k, c, m,t,l) where k is an any natural number, c is noun-phrase concept, m is a ranking criterion, t is temporal information, l is location information.
Here k and c are must require, while m, t, l are optional tuples.

2) From the body of page d, we can extract k and only k items such that:
a) Each item represents an entity that is an instance of the concept c in an is-a taxonomy;
b) The pair wise syntactic likeness of the k items is greater than a threshold.
Here, the syntactic likeness is closeness between two terms. For example, suppose title of web page i.e. t is "Top 10 dangerous animals in India", from this we can extract k = 10, c = " animals", m = "dangerous", t = null and l = "India".The top-k extraction problem can then be characterized as three sub-problem :
1) Title recognition T: (t, d) → (k, c, m, t, l)
2) List extractor L: (k, c, d) → I where I is the set of terms which are instances of c and |I| = k
3) Content extractor C: (c, d, I) → (T, S) where T is a table of attribute values for the elements in I and S is its pattern.

IV. PROPOSED SYSTEM

Figure shows architectural flow of the system. When we enter search query, we get unstructured web result relevant to search query. In that we get billions of web URL's. Then perform following 5 steps one by one to get expected result. These 5 steps are:
Step 1: Extract web URL's and its titles
Step 2: Remove dust from web URL's
Step 3: Run Levenshtein distance and Sort result by distance
Step 4: Run HTML parser
Step 5: Get closest result

In step one, to extract web URL's and its titles we use JSON parser. JSON parser parses unstructured data and extract closest 10 URL's with page titles relevant to that search query and display it. We provide limit of 10 URL's to reduce time and get result instantly.
In second step we remove dust from URL's. To remove dust we check URL's which contain dust such as image,

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017)
AGTI's Dr. Daulatrao Aher College Engineering, Vidyanagar Extension, Karad
Vol. 4, Special Issue 4, January 2017

audio, video, YouTube, Gmail, yahoo, twitter, LinkedIn, Facebook etc. If any one of these dust is present in URL then mark it as true otherwise false.
In third step, we run Levenshtein distance algorithm on list obtained in second step and sort result by distance in descending order. Display the sorted URL's with distance.
In fourth step, we parse obtained URL's by using DOM parser. Extract data from only those URL's which is marked as false.
In fifth step, if the data is not present in tabular form then display the whole page in first URL.

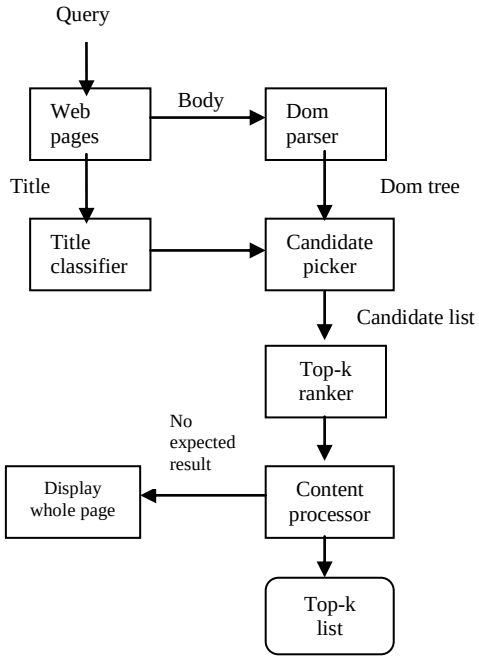


Fig.1Architecture diagram

Proposed system contains following main components:

A. Title Classifier:
Top-k page is identified by title of web page. Hence, it is necessary to classify page title. There are a few purposes behind us to use the page title to perceive a top-k page.

1) In most cases, a page title helps to present the topic of the main body.
2) While the page body may have changed and complex formats, top-k page titles have moderately comparative structure.

Title analysis is lightweight and profitable. Following figure shows example of top-k title.

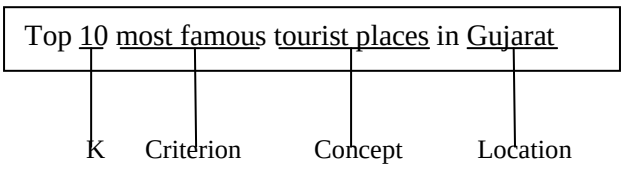


Fig.2.Example of top-k title

Title contain numerous segments, there is only one segment depicts topic or concept of the list. Here value of k (e.g. 10) and concept (tourist places). Top-k title also contains additional segments such as ranking criterion ("top", "most famous").

B. Candidate picker:
Utilizing HTML page body and the number k, the candidate picker accumulates a set of list items as candidates. Every list item is a text node in the page body. List contains only 10 titles and its URLs closer to search query. These 10 items obtained using JSON parser by parsing unstructured data.
These 10 URLs also contain dust such as image, audio, video, YouTube, Gmail, yahoo, twitter, LinkedIn, Facebook etc. We should have to remove dust. If any one of these dust is present in URL then mark it as true otherwise false. We take only those URLs as candidates which are marked as false.

C. Top-k ranker:
Top-K Ranker positions the candidate set and picks the top positioned list as the top-k list by using Levenshtein distance algorithm.
Levenshtein distance algorithm calculates difference between two strings. Levenshtein distance also called as edit distance because it is difference between minimum numbers of single character edits.

Levenshtein Distance Algorithm
Step1: Set l1 to be the length of str1.
Set l2 to be the length of str2.
If l1 = 0, return l2 and exit.
If l2 = 0, return l1 and exit.
Construct a matrix containing 0...l2 rows and 0..l1 columns.
Step 2: Initialize the first row to 0...l1.
Initialize the first column to 0...l2.
Step 3: Examine each character of str1 (x from 1 to l1).
Step 4: Examine each character of str2 (y from 1 to l2).
Step 5: If str1[x] equal's str2[y], the cost is 0.
If str1[x] doesn't equal str2[y], the cost is 1.
Step 6: Set cell d[x, y] of the matrix equal to the minimum of:
a. The cell immediately above plus 1: d[x-1, y] + 1.
b. The cell immediately to the left plus 1: d[x, y-1] + 1.
c. The cell diagonally above and to the left plus the cost: d[x-1, y-1] + cost.

D. Content processor:
Content processor process the list of URL s obtained from top-k ranker and gives output as tabular format in a manner such that the user would find it easy to read the data from the list. It processes each URL one by one. If first URL contains expected result then display top-k list as a result. Otherwise go to next URL. This process continues until exact result found. If these 10 URL s do not have proper result then we display first URL page as it is.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

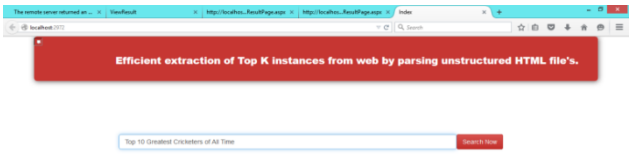
National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017)

AGTI's Dr. Daulatrao Aher College Engineering, Vidyanagar Extension, Karad

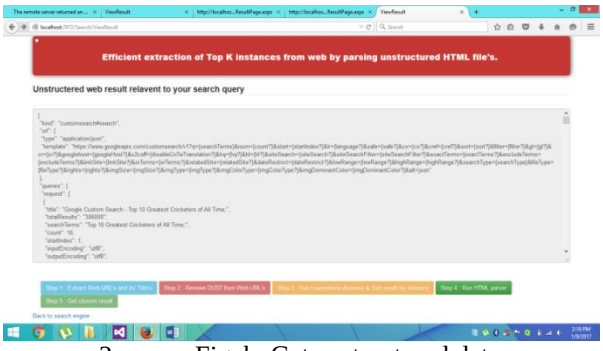
Vol. 4, Special Issue 4, January 2017

II. RESULT

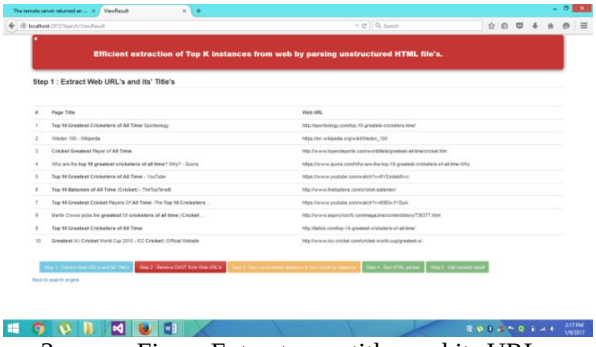
We test our system on the different online website page furthermore, on the different domains we found that our execution approach improves the execution of the system. Following figures shows our proposed system.



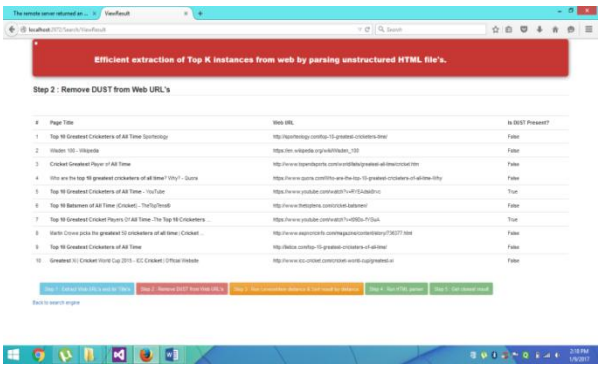
1. Fig. a. user enter search query



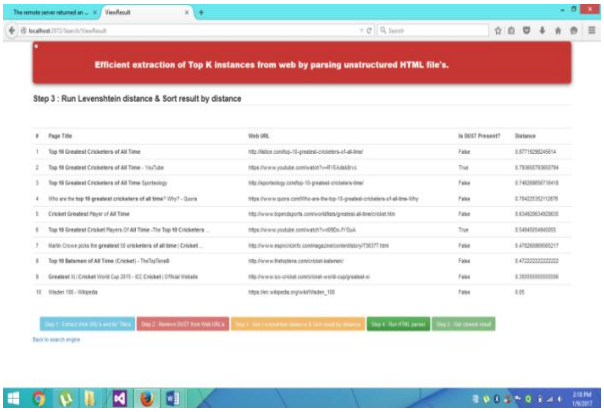
2. Fig. b. Get unstructured data



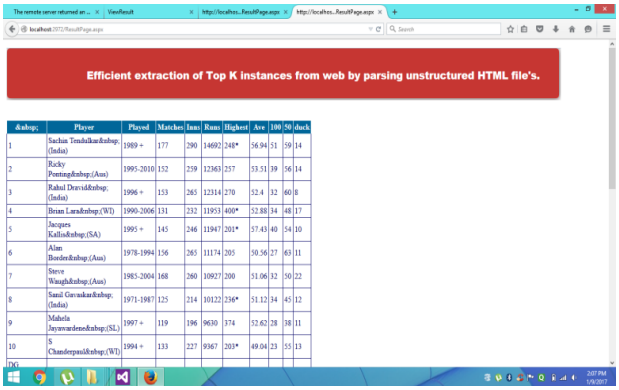
3. Fig. c. Extract page titles and its URLs



4. Fig. d. Remove dust from URLs



5. Fig. e. Apply Levenshtein algorithm



6. Fig. f. Expected result

III. CONCLUSION

Finally, for finding top-k list as a result we have implemented the extraction of top-k list from the web. We would like to conclude that top-k list extraction problem, which goes for recognizing, extracting "top-k" lists from web pages. Understanding of this top-k list easier, clear, more interesting as compared to structured data. Hence, it is different other data mining tasks. Other than these advantages, "top-k" lists are of great importance in knowledge discovery and reduce time consumption because there are huge numbers of web URL s around on the web. Customer can without much of a stress get result of top-k using above structure executed to focus top-k list from the web.

ACKNOWLEDGEMENT

Our team want to thank our guide, teachers and Head of department for their guidance. We are also grateful to the reviewer for their precious suggestions.

REFERENCES

[1] Zhixian Zhang, Kenny Q. Zhu, Haixun Wang Hong songLi , "Automatic Extraction of Top-k Lists from the Web" IEEE ,ICDE Conference, 2013, 978-1-4673-4910-9.

[2] 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. 981–990.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017)

AGTI's Dr. Daulatrao Aher College Engineering, Vidyanagar Extension, Karad

Vol. 4, Special Issue 4, January 2017

[3] Z. Zhang, K. Q. Zhu, and H. Wang, "A system for extracting top-k lists from the web," in KDD, 2012.
[4] M. J. Cafarella, E. Wu, A. Halevy, Y. Zhang, and D. Z. Wang, "Webtables: Exploring the power of tables on the web," in VLDB, 2008.
[5] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Kr¨upl, and B. Pollak, "Towards domain-independent information extraction from web tables,"in WWW. ACM Press, 2007, pp. 71–80.
[6] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu, "Understanding tables other web," in ER, 2012, pp. 141–155.
[7] Priyanka Deshmane, "Top-K list extraction from web pages", in IJCA 2016
[8] B.Naresh, "Dynamic data extraction of Top-K list from the web", IJRAET 2015
[9] Shreya U. Wadkar, "A review on Extracting top-K list from web", IJARCCE 2015