Top-k List Web Data Extraction from Dynamic Pages: TWD

[1] Shital A. Aher, [2] Seema.B.Siledar

[1] PG Scholar ,[2] Assistant Professor
[1] Computer Engineering Department,
[1] Marathwada Institute of Technology, Auragabad, India

***Abstract* - The World Wide Web is currently the largest source of information. This large database which contains information in all area but finding particular information or extracting accurate data from web is difficult. This paper is concerned with "top-*k*" pages, which are web pages that describe a list of *k* instances of a particular topic or concept. Examples include "the 10 tallest persons in the world". Top-k list are larger and richer in high quality of data, which is helpful to enrich existing knowledge bases about general concept which are used to answer a facts queries. The extracted lists can also be used as background knowledge for Q/A system. The paper concentrate on an efficient method, called Tag Path Clustering that extracts top-k list from web pages with high performance.**

***Index Terms* - top-k list, Candidate picker, ranker, title classifier.**

I. INTRODUCTION

The World Wide Web is currently the largest source of information. However, most information on the web is unstructured text in natural languages, and extracting knowledge from natural language text is very difficult. Still, some information on the web exists in structured or semi-structured forms, for example, as lists or web tables coded with specific tags such as <ul>, <li>, and <table> on html pages. As a result, a lot of recent work has focused on acquiring knowledge from structured information on the web, in particular, from web tables. In this paper, instead of focusing on structured data (such as tables) and ignoring context, we focus on context that we can understand, and then we use the context to interpret less structured or almost free-text information, and guide their extraction. Specifically, we focus on a rich and valuable source of information on the web, which we call top-k web pages. A top-k web page describes k items of a particular interest.

Examples of top-k titles:

1. Top 10 automobile brands in India.
2. Top 15 android apps of 2013.
3. Five most popular social networking sites.
4. Top 10 banks in world

The title of a top-k page contains at least three pieces of important information: i) A number k, for example, 20, Twelve which indicates how many items are described in the page; ii) A topic or concept the items belong to, for example, Scientists, children's Books iii) A ranking criterion, for example, Influential, Interesting. Some top-k titles contain two optional pieces of information: time and location. For example, 2011 and USA in the above example. In this paper, we present an efficient method that extracts top-k lists from web pages with high performance. Our system is designed to extract "top-*k*" lists from web pages. Basically, it performs three tasks: 1) Recognize a "top-*k*" page; 2) Extract the "top-*k*" list; 3) Understand and process list content. The input of the system is any HTML web page and the output is the extracted "top-*k*" list of the page.

II. RELATED WORK

There were many previous attempts to extract lists or tables from the web. None of them targets the "top-*k*" list extraction that is studied in this work. In fact, most of the methods are based on either very specific list-related tags such as <ul>, <li> and <table> or the similarity between DOM trees and ignore the visual aspect of HTML documents. These approaches are likely to be brittle because of the dynamic and inconsistent nature of web pages. In the top-k list extraction problem, there are some techniques can be categorized as follows: 1. Heuristic methods 2.Automatic extraction rule discovery 3.Similarity-based extraction 4.Visual model and features. 3 & 4 are more practical, as 1 & 2 are not very robust against complicated web pages. 4 has better accuracy since web pages are rendered for visual presentations, thus the visual models should be more expressive and intuitive in representing a list or table. 3 is often more efficient in time as it does not need to render the page. WebTables [3] Extract web lists or tables based on very specific list-related tags, such as <UL>, <OL>, <DL>, and <TABLE>.IEPAD [10] Identifies repetitive substrings as list patterns in an encoded document/web page.MDR [4] Extract data records of the same type based on the similarity between DOM trees which is measured by edit distance. Miao et al. [5] Based on a similarity measure between visual signals, they perform clustering of tag paths and rebuild the structure of data in the form of sets of tag paths. Ventex [6] Uses CSS2 visual box model instead of DOM trees to represent web pages, and extract web tables based on several rules and heuristics. HyLiEn [7] is a hybrid list extraction approach as it not only utilizes the visual alignment of list items but also takes advantage of structural feature (DOM tree). And it claims a remarkable improvement compared with Ventex[6].

III. THE PROPOSED SCHEME

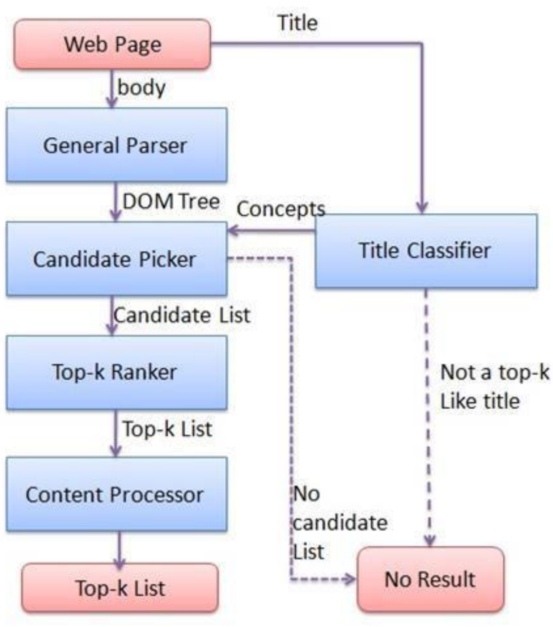


Fig.1. Proposed System

In this paper, we focus on a rich and valuable source of information on the web, which we call top-k web pages. A top-k web page describes k items of a particular interest. In most cases, the description is in natural language text which is not directly machine interpretable, although the description has the same format or style for different items. But most importantly, the title of a top-k page often clearly discloses the context, which makes the page interpretable and extractable.

MODULES DESCRIPTION

• Title Classifier
•Candidate Picker
• Top-K Ranker
• Content processor

Title Classifier: The title of a web page (string enclosed in <title> tag) helps us identify a top-k page. A classifier to recognize top-k titles is positive titles and negative titles. The following simple rules to identify or create positive training samples.

1. **"top CD": If a title contains the word "top" followed** by a number, it is likely to be top-k title.
For example, "top 10 NBA players who could be successful general managers".
2.**"top CD" without "top": A title which satisfies the "top** CD" rule is still a top-k title with the word "top" removed.
3. **"CD JJS": "JJS" stands for superlative adjectives. If** a title contains a number followed by a superlative adjective, it is likely to be a top-k title.
For example,"20 tallest buildings in China".
4.**"CD RBS JJ": "RBS" and "JJ" stand for superlative** adverbs and adjectives, respectively. If a title contains a number, followed by a superlative adverb, and followed by an adjective, it is likely to be a top-k title. For example, "5 most expensive watches in the world".

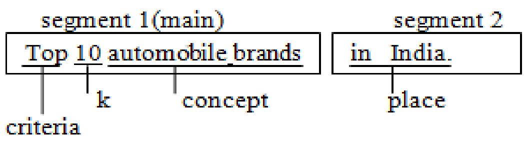


Fig 1. Example of Top-k Title

Candidate Picker: This step extracts one or more list structures which appear to be top-k lists from a given page.
i) K items: A candidate list must contain exactly k items.
ii) Identical tag path: The tag path of each item node in a candidate list must be the same.

The *Tag Path Clustering Method, shown in Algorithm 1* , process the input page according to the two basic rules the algorithm recursively computes the tag path for each node (Line 2), and groups text nodes with identical or very similar tag path into one node list. When this procedure completes, we get a set of node lists, those of k nodes are selected into the candidate set.

Three additional pattern-based rules to further filter the candidate lists:

1) **Index:** There exists an integer number in front of every list item serving as a rank or index: e.g., "1.", "2.", "3.".
2) **Highlighting Tag:** The tag path of the candidate list contains at least one tag among *<b>, <strong>, <h1-h6> for highlighting purposes*
3) **Table:** The candidate list is shown in a table format.

Algorithm 1 Tag Path Clustering Method

```
1:  procedure TAGPATHCLUSTERING($n$,$table$)
2:      $n.TagPath \leftarrow n.Parent.TagPath + Splitter + n.TagName$;
3:      if $n$ is a text node then
4:          if $table$ contains the key $n.TagPath$ then
5:              $list \leftarrow table[n.TagPath]$;
6:          else
7:              $list \leftarrow$ new empty lists;
8:              $table[n.TagPath] \leftarrow list$
9:          end if
10:         Insert $n$ into $list$;
11:         return;
12:     end if
13:     for each node $i \in n.Children$ do
14:         $TagPathClustering(i, table)$;
15:     end for
16:     return;
17: end procedure
```

Top-K Ranker: Top-K Ranker ranks the candidate set and picks the top ranked list as the top-k list by a scoring function which is a weighted sum of two feature scores below:

P -Score: P -Score measures the correlation between the list and title. We extract a set of concepts from the title, and one of them is the central concept of the top-k list. We calculate the P-Score of each candidate list L by:

$$P\text{-}Score = \frac{1}{k}\sum_{n\in L}\frac{LMI(n)}{Len(n)};$$

Where LMI (n) is the word count of the longest matched instance in the text of node n, while Len (n) means the word count of the entire text in node n. We divide LMI(n) by Len(n) to normalize the P-Score to [0, 1], and the contribution of each node will be no more than 1/k, which makes sure that one single node's effect doesn't dominate the whole score. In addition, P-Score prefers lists with fewer words, because nodes with many words (e.g., a description paragraph) are less likely to be part of a top-k list.

V -Score: V -Score calculates the visual area occupied by a list, since the main list of the page tends to be larger and more prominent than other minor lists. The V –Score of a list is the sum of the visual area of each node and is computed by:

$$Area(L) = \sum_{n\in L}(TextLength(n) \times FontSize(n)^2)$$

Content processor: The content processor get the list from the candidate lists and tables, it and needs to produce the properly structured results in the form of attribute-value pairs.

IV. MATHEMATICAL MODEL

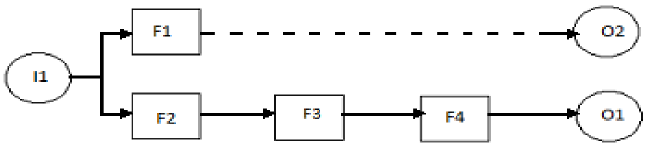


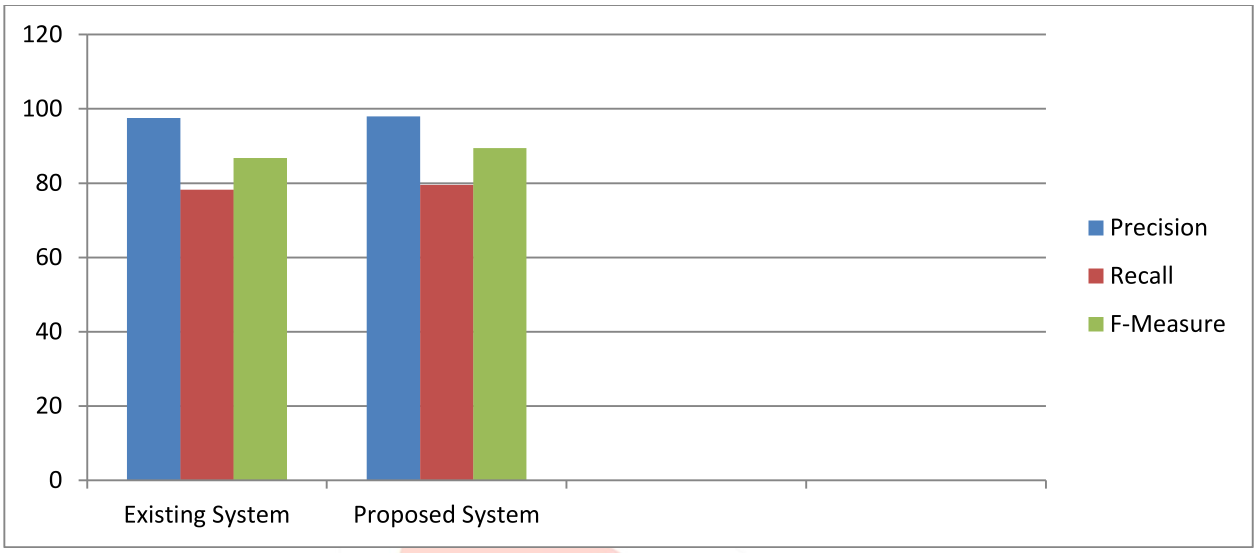
Fig.3.Mathematical Model

Input (I) Parameter
I= [I1]
Where I is a set of Input.
I1= Web page.
Function (F) Parameter
F= [f1,f2,f3,f4,]
Where F is a function for processing
F1= Title Classifier,F2= HTML Parser,F3= Top k- ranker,F4= Content Processor
Output (O) Parameter
O= [O1, O2], Where O is the Output.
O1= Structured schema of input query,O2=If no such top-k title is like query.

V. EXPERIMENTAL RESULTS

We test our system on the various online webpage and on the different domains we found that our implementation approach improves the performance of the system. We used html DOM tree with the pruning techniques which help to minimized time complexity of our project. Following table show the performance of Existing system and our proposed method.

	Precision	**Recall**	**F-measure**
Existing System	92.00%	72.30%	80.96%
Proposed System	94.44%	74.40%	83.23%



VI. CONCLUSION AND FUTURE WORK

In this paper, we define a novel list extraction problem, which aims at recognizing, extracting and understanding "top-*k*" lists from web pages. The problem is distinctive from other data mining tasks, because compared to other structured data, "top-*k*" lists are clearer, easier to understand and more interesting for readers. Besides these advantages, "top-*k*" lists are of great importance in knowledge discovery and fact answering simply because there are millions of "top-*k*" lists around on the web. Our proposed 4-stage extraction framework has demonstrated its ability to retrieve large number of "top-*k*" lists at a very high precision. This project can be extended to find information from links present inside other links and try to reduce computational work.

REFERENCES

[1] Zhixian Zhang, Kenny Q. Zhu , Haixun Wang , Hongsong Li ,"Automatic Extraction of Top-k Lists from the Web",, IEEE , ICDE Conference, 2013, 978-1-4673-4910-9.
[2]Soumen Chakrabarti Mining The Web: Discovering Knowledge From Hypertext Data".
[3] M. J. Cafarella, E. Wu, A. Halevy, Y. Zhang, and D. Z. Wang, "Webtables: Exploring the power of tables on the web", in VLDB Auckland, New Zealand, 2008.
[4] B. Liu, R. L. Grossman, and Y. Zhai, "Mining data records in web pages," in *KDD, 2003, pp. 601–606.*
[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. 981–990.
[6] 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.
[7]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. 285–294.
[8] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu, "Understanding tables on the web," in ER, 2012, pp. 141–155.
[9]Chang, C.-H., Kayed, M., Girgis, M. R., Shaalan, K., A Survey of Web Information Extraction Systems, IEEE TKDE (SCI, EI), Vol. 18, No. 10, pp. 1411-1428, Oct. 2006.
[10] C.-H. Chang and S.C. Lui, "Iepad: information extraction based on pattern discovery," in WWW, 2001, pp. 681–688.
[11] Z. Zhang, K. Q. Zhu, and H. Wang, "A system for extracting top-k lists from the web," in KDD, 2012
[12] Dipali Patil, Nitin Dhawas, "Enhancement in Extraction of Top-k List", Singhad Institute of Technology, IOSR-JCE, Volume 16 Issue 3, ver III (May-Jun), PP 129-133, 2014.