IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. III (May-Jun. 2014), PP 129-133
www.iosrjournals.org

Enhancement in Extraction of Top-K List

Dipali Patil[1], Nitin Dhawas[2]
[1]*Sinhgad Institute of Technology,Lonavala, India*
[2]*Sinhgad Institute of Technology,Lonavala, India*

Abstract: *In today's busy schedule finding proper information within less time is important need. When user fires top-K list or any other query, user get multiple links as output. User has to visit sites and have to search the proper result manually. But one more problem is there all data available on web is not in same format. These two problems are solved by using proposed enhanced top-k list extraction system. It will give user direct top-k list as result when user fire top-k list as query. This top-k list extraction system depends upon top-k list extraction algorithm. This system handles all web pages that may be structured, unstructured or semi-structured. Also it consumes user time to find proper information.*

Keywords: *Candidate picker, parser,ranker, top-k list, title classifier*

I. Introduction

World Wide Web contains very huge amount of information. Extracting useful information from web is called as web mining. But it takes two forms: (1) Extracting structured form information (2) Extracting natural language text. The structured form information is mainly in HTML or XML language. But information tags like <li>, <table> and <ul>. Again the question arises, 'is this tabular data is valuable?'. Many times the answer is NO. User may get huge tables on web but inside those tables only small amount of information is valuable.

Understanding of context is very important because in list extraction user must know the relation between listed items. For example, table which contains two columns first is Book name and second is prize of the book. This table contains five entries. But can't get why those books are listed together, e.g. are they most famous books, are they from same regional language, are they written by same author, are they published by same publisher etc. In short we don't known under which parameters information is collected together and what is the use of this table. To avoid this situation user must know the context. Generally the context is written in natural language. But machine can't interpret natural language.

As time passes technology is going to become advanced and faster. On web when user fire any query, user will not get direct result. Result is nothing but the number of links provided by search engine; user has to visit every link and has to find proper or correct data manually. It means search engine is providing most preferable or related links not the direct result. For result user visit first link if user gets result or particular information then search is stopped. Otherwise user has to visit next link if that link contain proper information the search is stopped and if not same procedure is repeated until user will get result. This normal process takes user's lot of time.

In 2008 work performed related to extracting tabular form information from HTML pages. But disadvantage of this system is it retrieves structured tabular data only. Above we discuss about tabular data available on web. In 2009, research continued with topic mining contagious and non-contagious data records. But this method is having less accuracy. Next research was performed on hybrid approach for discovering general list and extracting it from web. It gives efficient and fast result but it provides general list only not a ranked list. In 2013 evolution is done in retrieving top-k list data from web pages. The main disadvantage of this system is, it is applicable to HTML pages only.

Due to above issue this system is focusing on rich and valuable data from web that we get with the help of top-k list. This list describes k number of items of particular type. All items from top-k list are bounded with each other through particular parameters. Following are the reasons due to which we target top-k pages for extraction:

1. Availability of top-k data on web is large and it is rich.
2. Top-k data is of high quality.
3. Top-k list data is already ranked
4. Top-k list data has interesting semantics

The paper is organized as follows: section II introduces top-k list concept, gives top-k list examples and also multiple segments of top-k title. Section III discusses detail about proposed system which includes work flow of system and proposed algorithm. Section IV presents the mathematical model. Section V discusses applications based on proposed system. The experimental results are reported in section VI. Section VII summarizes our contributions and concludes the paper.

II. Top-K List

Top-k list is a list which contains k number of ranked elements. Where, k is any integer value. As compared to web tables top-k list contain rich and valuable data.Following are some examples of top-k titles:

1. Top 10 automobile brands in India.
2. Top15 android apps of 2013.
3. Five most popular social networking sites.
4. Top 10 banks in world
5. Twelve most interesting books in philosophy.

Every top-k page title contains minimum three piece of information:

i. **k**: for example, 10, 15, five, twelve in above example. That tells how many items are in top-k page.
ii. **concept or topic**: it tells which kind of item is retrieved. For example, automobile brands, android apps, social networking sites, banks, books etc.
iii. **criteria for ranking**: it decides on which basis ranking is provided to provide.

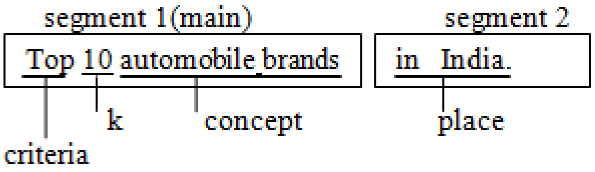

Fig 1. Example of Top-k Title

Above fig shows example of top-k title. Top-k title may have many segments. The above example shows only 2 segments, First segment is main segment and second segment contain other modifiers. Segment 1 contains criteria- top, k-10 and concept-automobile brands. Second section gives information about place i.e. India. In many top-k title additional information about place or time is provide.

III. Problem Statement

Let a web page be a pair (t, d) where tis the page title, and d is the HTML body of the page.

- A page (t, d) is a top-k page if:
- From title t we can extract a 5-tuple (k, c, m, t, l)
- Where k is a natural number, c is a noun-phrase concept defined in a knowledge base, m is a ranking criterion, t is temporal information, lis location information. Note that k and c are mandatory, while m, t, and l are optional [2].
- From the page body 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 similarity of the k items is greater than a threshold.

- The top-k extraction problem can then be defined as three sub-problems (in terms of three functions):
- Title recognition F1 :

(t, d)→(k, c, m, t, l)

- List extractor F2 :

(k, c, d) → I
Where I is the set of terms which are instances of c and |I| = k

- Content extractor F3 :
  (c, d, I) →(T, S)
  Where T is a table of attribute values for the elements in I and S is its schema.

IV. Proposed System

The system consists of the following components:

- Title Classifiers, which attempts to recognize the page title of the input web page.
- Candidate Picker, which extracts all potential top-k lists from the page body as candidate lists.

- Top-K Ranker, which scores each candidate list and picks the best one;
- Content Processor, which post processes the extracted list to further produce attribute values, etc.

1. Title Classifier

The title of a web page (string enclosed in <title> tag) helps us identify a top-k page. There are several reasons for us to utilize the page title to recognize a top-k page. First, for most cases, page titles serve to introduce the topic of the main body. Second, while the page body may have varied and complex formats, top-k page titles have relatively similar structure. Also, title analysis is lightweight and efficient. If title analysis indicates that a page is not a top-k page, we chose to skip this page. This is important if the system has to scale to billions of web pages.

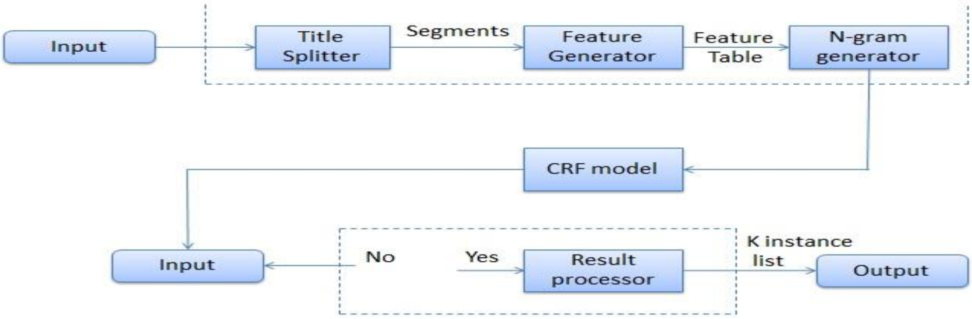

Fig. 2 Flow Chart of Title Classifier

2. Candidate Picker

This step extracts one or more list structures which appear to be top-k lists from a given page. A top-k candidate should first and for most be a list of k items, Visually, it should be rendered as k vertically or horizontally aligned regular patterns. While structurally, it is presented as a list of HTML nodes with identical tag path. A tag path is the path from the root node to a certain tag node, which can be presented as a sequence of tag names.

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.

3. 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:

i) P-Score:

P-Score measures the correlation between the list and title. In Section IV-A, we extract a set of concepts from the title, and one of them is the central concept of the top-k list. Our key idea is that one or more items from the main list should be instances of that central concept from the title. For example, if the title contains the concept “scientist”, then the items of the main list should be instances of the “scientist” concept. The Probase taxonomy provides large number of concepts and their instances. For instance, the “scientist” concept has 2054 instances in Probase. We calculate the P-Score of each candidate list L by

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

ii) 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:

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

4. Content Processor

After getting top-k list, we extract attribute/value pairs for each item from the description of the item in the list. The goal is to obtain structured information for each item. Furthermore, by analyzing the title, we obtain valuable information like the location is “the united states” and the date is “2010”. We describe three major steps in the content processor.

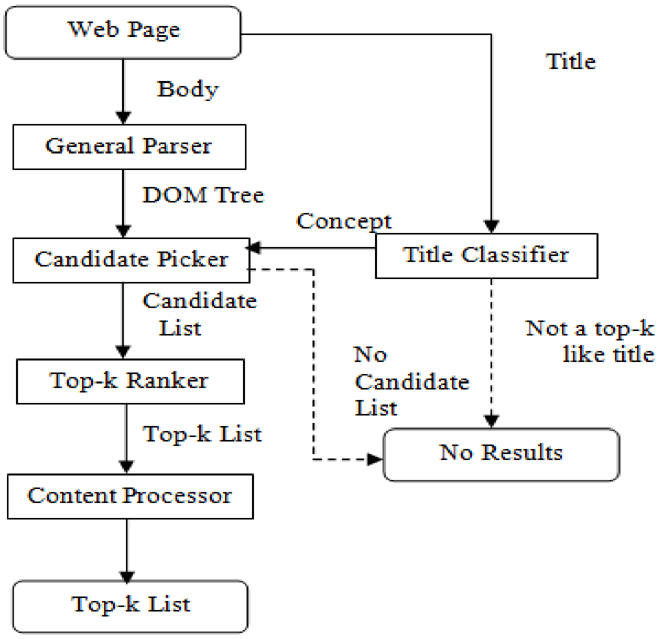

Fig. 3 System Flow

i) Infer the structure of text nodes:

In many cases, the text node that describes each item may have some inner structure, or is semi-structured. We infer the inner structure of the text by constructing the frequency distribution for each potential separator tokens such as “By”, “:” and “,” from all the items of the top-k list.

ii) Conceptualize the list attributes:

Once the list items are broken down into attribute values, it is useful to infer a schema for these attributes. In our system, we utilize three methods to conceptualize list attributes.

iii) Table head:

If the list is shown in a table format, i.e., satisfies Rule Table and the table itself contains a head, we can use the table head to conceptualize the table directly. Generally, we can find the table heads by the <th>tags.

iv) Attribute/value pair:

In some cases, the list may contain explicit attribute/value pairs. Generally, if every element of a column contains the same text and ends with a colon, we will consider that column as the attribute column and the column to the right as the value column. Then we can use the attribute name to conceptualize the corresponding values.

v) Column content:

If neither table heads nor attribute value pairs are available, the default method is to conceptualize the extracted column contents by a method proposed by Song et al. [4], using Probase and a Bayesian model. For each text column, we use the longest known probase instance in the text to represent the text node and thus obtain an instance set of size k:

E = {e_i, i 2 1…., k}. We then need to find a concept that best describes the instance set. The probability of concepts given the instance set E can be estimated by a Naive Bayesian Model.

$$P(ck|E) = \frac{P(E|ck)P(ck)}{P(E)} \propto P(ck) \sum_{i=1}^{M} P(ei|ck)$$

V. Applications

We are building a Q/A system using the top-k data to answer queries such as “tallest persons in the world”, or “What are best-selling books in 2010” directly. For this there must be a valid query provided as an input to the product. If the query is valid then an instant result is provided otherwise there could be an unsatisfied result.

VI. 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. N gram model improve the accuracy of result. We used html dom tree with the pruning techniques which help to minimized time complesity of our project. Following table show the performance of rule based, learning based and our proposed method.

Algorithm	**Precision**	**Recall**	**F-Measure**
Rule-Based	95.5	71.9	82.0
Learning-Based	97.5	78.2	86.8
Proposed Method	98.0	79.5	89.4

Table 1: Results of List Extraction

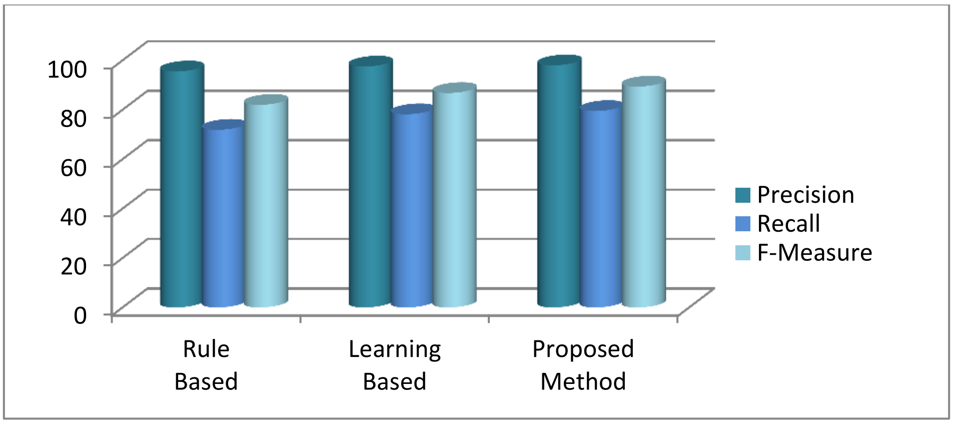

Fig. 4 Graphical representation of Precision, Recall andF-Measure

VII. Conclusion And Future Work

Finally we would like to conclude that we have implemented the extraction of top-k list from the web. For finding out the top-k list. Also we would like to conclude that compared to other structure data top-k list are cleaner, easier to understand and more interesting for human consumption and therefore are an important source for data mining and knowledge discovery. Our project can be extended to find information from links present inside other links and try to reduce computational work.

References

[1] Soumen Chakrabarti Mining The Web: Discovering Knowledge From Hypertext Data".

[2] 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.

[3] G. Miao, J. Tatemura, W.-P. Hsiung, A. Sawires, and L. E. Moser,"Extracting data records from the web using tag path clustering".

[4] Z. Zhang, K. Q. Zhu, and H. Wang, "A system for extracting top-k lists from the web," in KDD, 2012

[5] C.-H. Chang and S.-C. Lui, "Iepad: information extraction based on pattern discovery," in WWW, 2001, pp. 681–688.

[6] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in SIGMOD, 2012.

[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] 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.

[9] 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.

[10] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho and Sau Dan Lee, "Decision Trees for Uncertain Data",IEEE conference,2011.

[11] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu, "Understanding tables on the web," in ER, 2012, pp. 141–155.

[12] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak, "Towards domain independent information extraction from web tables", In WWW, pages 71–80. ACM Press, 2007.