A WEIGHT BASED APPROACH TO EXTRACT TOP-K LIST FROM THE WEB

Mahesh Dabade, Shriniwas Gadage

Abstract— **The WWW contains huge amount of data and this data is a source of large amount of information. The information obtained from web is in two forms 1) structured & 2) unstructured data (Example of structured data is web tables, list etc and example of unstructured data is NLP). List data is one of the most important sources for information retrieval. Extracting information from structured data was the main focus of researchers as compared to unstructured data. So we have focused on extracting information from both the structured data and unstructured data. We specifically extract data from WebTables because we focus on context about the information that we can spot on rather than focusing on context free structured data. Here in this paper, we highlight expensive as well as rich source of information on the web, those are top-k web pages that describe top-k instances of any particular topic of interest which is very useful for search. We propose an enhanced Top-k list extraction system in which the system will give the user direct top-k list when user fires top-k query. This top-k list extraction system depends upon 1) Extracting the links from the web 2) Extracting data from the links using Extraction algorithm.**

Index Terms— **Top-k lists, information retrieval, web information extraction, web mining, data tag path**

I. INTRODUCTION

The World Wide Web contains very large amount of information and extracting useful information from the web is called web mining. However, information extraction takes two forms 1) Extracting structured information 2) Extracting unstructured information. The structured information is in HTML or XML language and this information is in specific tags such as <ul> and <table>. A question often arises about the value of knowledge that we get when we extract data from these lists and web tables. It is no doubt that the entire corpus is huge and contains large amount of information but only a small percentage of them contains useful information and even smaller percentage of them contains information that is interpretable without context. According to Caferlla et. al.[3] there is a small percentage of web tables that are relational and most of them are meaningless without context. For example, suppose we have extracted a table that contains 4 rows and 2 columns named book and price respectively. It is still unclear why these 4 books are grouped together (e.g. - are they most famous books, are they written by same author, published by the same publisher) and how we should interpret their price. In other words, we don't know the context of the information under which it will be useful. Understanding the context of information is of very much importance. In most cases, the context of information is expressed in unstructured text that systems cannot interpret. So, we focus on context that we can understand and then use the context to interpret less structured information and guide its extraction. We focus on top-k pages which are rich source of information on the web. The top-k describes the top-k instances of any particular topic of interest. Most often the title of the top-k page discloses the context related to the web page which makes the page interpretable and extractable. Some typical examples are:

- Top 10 tallest buildings in the world
- 20 most influential scientists in the world
- Ten most expensive Hollywood actors

The title of a top-k page should contain at least three types of information: 1) a number k 2) a topic/concept 3) a ranking criterion.

As the time passes the technology is becoming advanced and faster. On the web when user fires any query, user will not get direct result that results are nothing but number of links provided by the search engine, the user has to visit every links and has to find proper or correct data manually. It means that search engine is providing most preferable or related links but not the direct result. For the result, user has to visit each link, if user gets results or Particular information then search is stopped otherwise, user has to visit next links, the same procedure is repeated until user gets the desired results or information. This normal process takes a lot of time of the user. Due to this issue the system is focusing on rich and valuable data from the web that we get from the top-k list. The top-k list has more understandable and interesting context and hence, we target top-k pages for extraction of information. Following are the reasons why we selected top-k list for data extraction:

- Availability of top-k data on the web is large and it is rich.
- Top-k data is of high quality
- Top-k data is already ranked
- Top-k data has interesting semantics

In short, the top-k list extraction performs 3 main tasks:

1. Recognize the top-k pages: we recognize top-k pages from a huge corpus by parsing the titles of the web

***Mahesh Dabade**, Computer Department, G. H. R. C. E. M). Pune, India.*
***Shriniwas Gadage**, Computer Department, G. H. R. C. E. M Pune, India.*

ISSN: 2278 – 1323

pages. Basically we segment the URL in segments and assign weights to the information which are tuples of a top-k page.

2. Extract top-k list: For each top-k page, we extract a list of k items. For this purpose we have developed an algorithm known as spectral data extraction algorithm in which we extract the list from the page body.

3. Assign weights to the list: We measure the correlation between list and titles and then assign weights to them so that we get the web page which has more weights.

Rest of the paper is organized as follows. Section II introduces the problem definition. Section III discusses the Literature Review. Section IV discusses the detail architecture of the system. Section V describes some Implementation details and experimental results. Section VI concludes the paper.

II. LITERATURE REVIEW

A lot of data on the Web is contained in consistently organized items, which we call data records. Such data records are imperative in light of the fact that they regularly display the fundamental data of their host pages, e.g., arrangements of items or administrations. It is valuable to mine such data records keeping in mind the end goal to concentrate data from them to give quality included administrations. Existing programmed strategies are not acceptable due to their poor exactnesses. The problem of extraction of data from the web presented in this work belongs to the general area of structured data extraction. In this form, many techniques have been devised and have a rapid growth of techniques have been devised and have a rapid growth of improvement. Google sets[2] and WebTables[3] focus on extracting the data from the web tables or tables that are based on very specific or related tags but the limitation here is that the data in the web table may not contain context and hence context may be ignored. The specific related tags can be <UL>, <OL> & <TABLE>. MDR i.e. mining data records from the web pages [4] is proposed to extract large amount of information on the web which is contained in regularly structured objects which are called Data records. Miao [5] introduced visual signal which is a vector describing the tag path occurrence patterns. There are in performing the data extraction 1) Detecting Visually repeating information 2) Extraction of data Records. Hylien[6] is a hybrid method to extract list which uses the visual method to extract list which uses the visual alignment of list items and also takes advantage of structured feature. IEPAD [12] identifies repetitive substrings as list patterns in an encoded document/web pages. Zhang [13] introduced a concept of working prototype of top-k extraction system. For future enhancements in the system, the Threshold Algorithm can be used which is one of the well known algorithms [14]. By the help of threshold algorithm, instead of selecting a main list from candidate list, the system can create its own list. Fagin [15] used TA to utilize aggregation functions to combine the scores of objects in different list and computes the top-k objects based on combined scores.

III. PROBLEM DEFINATION

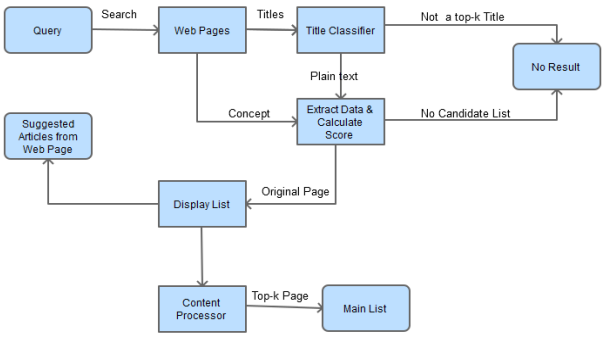
In this section, we formally outline the matter of extracting Top-k lists from the web. Let an web page be a combine (t,d) wherever t is that the page title, and d is that the HTML body of the page. A page (t, d) could be a top-k page if: 1) From title t we are able to extract a 4-tuple (k, c, m, l) where k could be a number, c could be a concept, m could be a ranking criterion, t is temporal information, l is location info. Note that k and c compulsory, while m, t, and l are optional. 2) From the page body d we are able to extract k and solely k items such that: each item is an instance of the entity it represents. For example, suppose t is "Twelve Most fascinating Children's Books in USA", we are able to extract k = one, c = "children's books", m = "interesting" and l = "USA". If the body of the page contains precisely twelve similar parts such as "Harry Potter" and "Alice in Wonderland", then we are able to conclude this can be a top-k page. The top-k extraction will then be outlined as 3 Sub problems

(In terms of 3 functions):

1) Title recognition tr : (t, d) -> (k, c, m, l)
2) List extractor
3) Content extractor

IV. OUR APPROACH

Figure shows the block diagram of the system. When the user fires or enters a query, the query is sent to the search engine; the search engine then parses the query and after crawling the web pages generates the results. After fetching the links from the search engine, the search component displays the links in its panel.



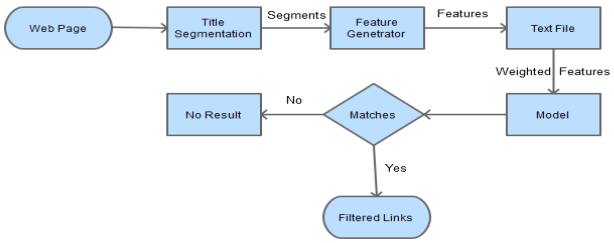
This links are unfiltered links, meaning that they may contain the relevant answers and also there exists a possibility of having false positives. The Title Classifier attempts to identify the page title. If the page title of the web page matches with the query keywords then the title classifier displays the list of URLs otherwise it outputs No Result. From the body of the web page, the system extracts the main concept, with the number k & the concept, the Extract Data & Calculate Score component extracts the data from the web page while filtering out the unwanted lists and advertisements. Then the data Extract and Calculate Score

ISSN: 2278 – 1323

component will assign weights to the lists obtained from the web pages. The weights are assigned by obtaining a correlation between the query and the tags in the html web pages. It means it finds the longest matching instances from the page body and correlates with the concept which is entered by the user and assigns weights to each list based on the criteria discussed above. This component then displays all the extracted list and their respective weights. The display list component then shows the list which has assigned the maximum weight. The web page is displayed in its original form but with removing ad and unwanted lists. This means that the user can see only the main list from the web page. The Display List component then also shows the suggested articles from the main list so that the user will not be left in the dark when he wants to request a page related to the concept.

A) Title Classifier: The title of a web page helps us to identify a top-k page. The main aim of the Title Classifier is to recognize the "top-k" titles. The reason for focusing on top-k titles is that it represents the topic of "Top-k" list.

0



B) Extract Data & Calculate Score: Given a number from the title of a page and the body of the web page, this component collects a set of candidate lists. To extract the data from the page, a new improved algorithm called clustering algorithm is introduced. The flaw of the tag path algorithm in the existing system was, from a given page, to extract the list structures, the HTML page was considered to have nodes with identical tag paths, the condition was that it must have k instances and should have identical tag path. Identical tag path means each item node should follow the same tags sequence. e.g.: If first node in a list have tags html/body/h2, html/body/fig, html/body/fig/figcaption, then the other nodes in the list must also have to follow same pattern otherwise the list will not be executed. The clustering algorithm makes use of the tag paths and follows three main steps. 1. Visual signal Extraction 2.Similarity Extraction 3.VisualSignal Clustering. In the Visual signal extraction, the tags of the html page are considered to find their respective tag paths and from the tag paths, the visual signals are obtained. For this purpose an inverted index characterizing the mappings from HTML tag paths to their locations in the HTML document can be built for each web page. After getting the visual signals from the tags, the similarity between the visual signals is calculated. the similarity function calculates the likelihood that the data samples follow the same visual signal patterns then these visual signals are then clustered in the same group. The advantage here is that tags does not need to have the same tag path every time. by using this algorithm, it has become more flexible to extract data from web pages which are in natural language.

C) Display List: The display list gives the user with a page that has got the maximum weight, this web page is in the original form but the system removes the unwanted data and advertises from the web page and provide user with a list of suggested articles.

D) Content Processor: The main list is then transformed into a tabular format in a manner such that the user would find it easy to read the data from the list.

E) Algorithm:

Algorithm 1 Weight based Algoirthm For Title Classifier

```
procedure TitleClassifier
while CurrentLine←new Criteria +new K values
if QueryQriteria←Criteria
QueryCriteria←Criteria
CriteriaWeight←CurrentlineCriteria
end if
for each Title∈QryTitle do
if t[i]is numeric
QryK←t[i]
end for
if QueryKWeight←K
end if
Segments←p_title+Splitter(\\)
for each Segment ∈Title
Segment1+Segment2+Segment 3←Splitter
if Segment[i]←QueryConcept+QueryCriteria+QueryK
if QueryAdjective←Empty List
QueryAdjective←QueryAdjective
end if
end if
end if
end if
if Knum← EmptyList && Concept←Empty List
Diff←CriteriaWeight−QueryCriteriaWeight
Diff1←KWeight−QueryKWeight
end if
```

Algorithm for title Classifier with weights

Algorithm 1 Tag Path Clustering Method

```
1:  procedure TAGPATHCLUSTERING(n,table)
2:      n.TagPath ← 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 ← table[n.TagPath];
6:          else
7:              list ← new empty lists;
8:              table[n.TagPath] ← list
9:          end if
10:         Insert n into list;
11:         return;
12:     end if
13:     for each node i ∈ n.Children do
14:         TagPathClustering(i, table);
15:     end for
16:     return;
17: end procedure
```

Algorithm for Data Extraction with weights

4.6 ADVANTAGES OF THE SYSTEM

1. Weight Assignment to Title Classifier Component: For retrieving accurate web pages from the web, the system divides each and every URL into multiple segments, the Algorithm then divides the segments into multiple parts to

reach the particular page having the title. After getting the title of the web page, the algorithm matches the number k and criteria entered by the user with the title of the web page. For this purpose, there is a facility to maintain a vocabulary of both the conditions and have assigned preference weights to the words in the vocabulary. If the words and number in the vocabulary match with the number and concept in the query and if the weight difference between them id found to be zero then the system selects that particular title else the links are filtered out. In the existing system, there was no facility to identify whether the links obtained in the title classifier are relevant or not, so there was a high possibility of occurring of false positives, By assigning weights to the words and numbers in the dictionary, the system was successful in filtering out most of the false positives. Hence, accurate results were obtained.

2. Automation of Extraction of Data and Calculation of Score with Weight Assignment: In the existing system, after getting a set of candidate lists, the lists were given to the top-k ranker component, the top-k ranker then used to rank the lists, this ranking was based on two criteria's, the first was word count of entire text node and the second was the visual area occupied by a list, but the drawback here was that the ranking criterion lacked flexibility. In the proposed Data Extract & Calculate Score component, the two components candidate picker and top-k ranker are merged for automating the system and the other extra feature added is that the new improved algorithm takes into account the longest matching instances from the text nodes i.e. the algorithm looks for the tags in the HTML web page that have the words with the concept requested by the user, if the match is positive then data is extracted and the score is calculated.

3. Elimination of Irrelevant Lists, Ads and Presenting User a List of Suggested Articles: After score calculation, the list that has been assigned maximum weight will be presented to the user. This web page will be in original form. The main advantage here is that there will be no irrelevant data found on the web page plus additional advantage would be a list of that the user will be presented with a list of suggested articles so that the user will be able to fire the query next time about the concerned concept whenever he requests for that particular concept/information.

4. Identification of Alphabetical k: One of the major improvements in the system is when the user types a alphabetical k, the existing system was unable to respond with relevant results. This consideration was taken while implementing the proposed system. in the vocabulary of k, the weights are also assigned to alphabetical k which is equivalent to its numerical counterpart. When the user types a alphabetical k and enters the search button, he will be presented with the same results.

The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

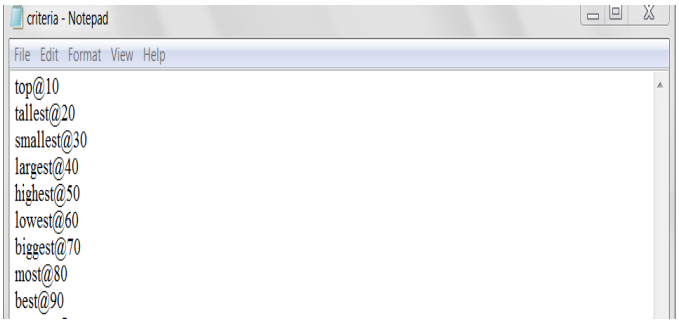
V. EXPERIMENTAL SETUP

The system is built using Java framework (version jdk 6) on Windows platform. The Netbeans (version 6.9) is used as a development tool. The system doesn't require any specific hardware to run; any standard machine is capable of running the application. In this section, we present some preliminary results of the system from two experiments. These two experiments test the precision and recall of the two main functions, namely title recognition and list extraction, respectively. The experiments were conducted on a PC with 1GB RAM and 2.70GHz Dual-Core Intel CPU having internet speed 512 kbps. The top-k extraction is a brand new topic in web mining. Although there have been many previous attempts to extract general lists and tables from the web, none of them target on top-k lists and are able to solve this specific problem. Therefore, we cannot set up any direct comparison with those methods. Instead, we compare several versions of our system, to show the significant improvement against our previous demo system.

A) Vocabulary for the System

Currently the system has two main vocabulary sections,
1. criteria
2. k

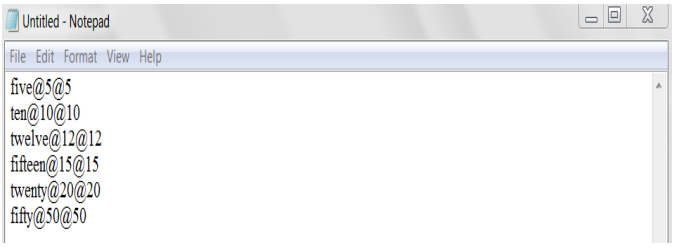
The criteria vocabulary has words which are based on ranking, By using these dictionary, the system is able to recognize the ranking criterion and is able to correlate the criteria entered by the user and then match it with the title of every web page. Following is snapshot for the the vocabulary. Currently it is restricted to few criterions but it can be increased depending upon the necessity of the system. If the criterion other than the words in the dictionary is entered then the system will not display accurate results.



Criteria Vocabulary

The k is the most important aspect in the proposed system, the instances which are extracted depends upon the number k, exactly k items should be fetched from a web page. The web page which has a matching title entered by the user will only be retrieved; no other web pages will be retrieved. One thing should also be taken into account that, the internet has standard for top-k pages only irrespective of the query submitted by the user, the internet will display top-k pages.

ISSN: 2278 – 1323



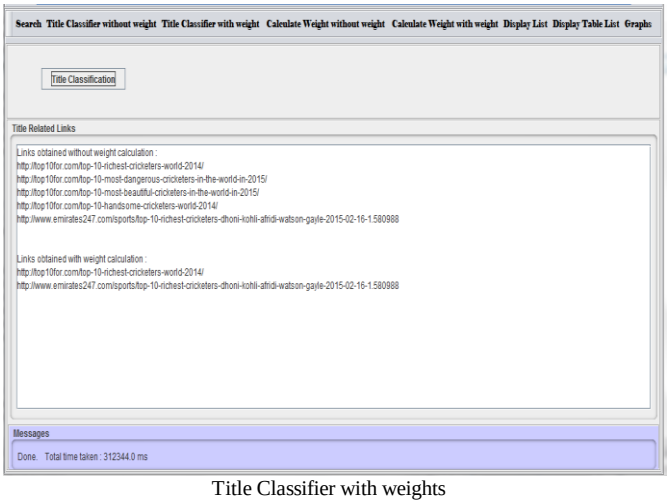
K Vocabulary

VI. DEMONSTRATION SETUP

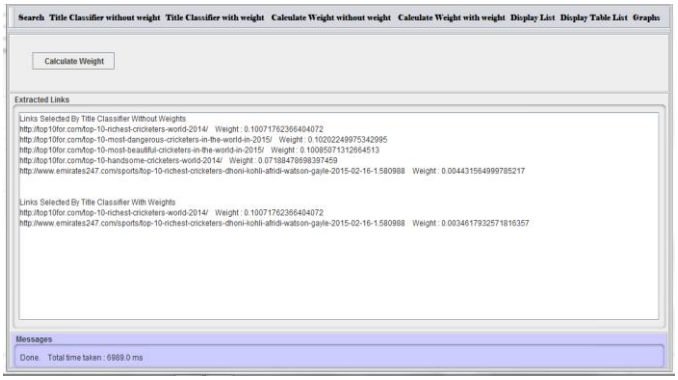
The system consists of 3 main components,

1. Title recognition with weights
2. Data extraction with weights
3. Display list

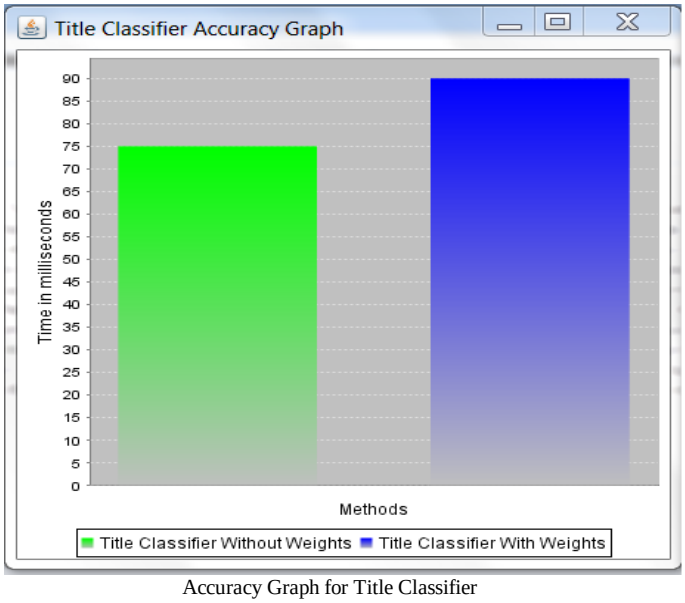
Under the title classifier section, a user can test our system with an arbitrary user query. We obtain the web pages by manually searching through over web pages with "top-k" like titles. A screenshot of the Title Classifier section with blow-ups of extracted content is shown in Figure. Here, a user can type in the query in the textbox and click "title classification" button, the system will retrieve the page in real time and attempt to extract a "top-k" list from it. The output result includes the page title, running time (in millisecond), number k, concepts as well as the "top-k list". Both the result and the original page will be presented after extraction. In data extraction section, a user can click "data extract" button, and the system will analyze it as a page title and return detailed result, including time, features, and concepts.



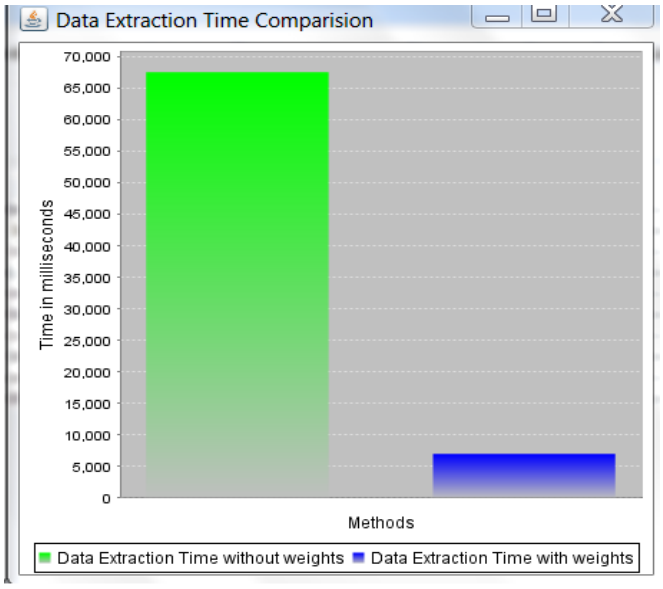
Title Classifier with weights



Data Extraction with weights



Accuracy Graph for Title Classifier



Data Extraction with weights

VII. CONCLUSION

The framework takes care of interesting issue of extracting top-k list from web, which goes for perceiving, extracting and comprehension top-k list from web pages. The extracted top-k list is of ranked and high quality. This top-k information is to a great extent accessible and has interesting semantic. Client can easily get results of top-k query utilizing above framework executed to concentrate top-k list from the web. Additionally we might want to infer that contrasted with other structure information top-k list are cleaner, easier to understand and all the more interesting for human utilization and in this way are a critical important for data mining and knowledge discovery. Also our system can applicable in historical data. And it has more accuracy than the existing system.

VII. ACKNOWLEDGMENTS

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We are thankful to the authorities of Savitribai Phule University of Pune. We are also thankful to the reviewer for their valuable suggestions. We also thank the

college authorities for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

.

REFERENCES

[1] 1. Zhixian Zhang, Kenny Q. Zhu, Haixun Wang and Hongsong Li," Automatic extraction of top-k lists from the web" in IEEE transactions, KDD 2013.
[2] "Google sets," http://labs.google.com/sets
[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, 2008.
[4] B. Liu, I. Grossman, Y. Zhai, "Mining data records in web pages" in KDD 2003, pp 601-606.
[5] G. Miao, J. Tatemura, A. Sawries, " Extracting data records from the web using tag path clustering" in WWW, 2009, pp- 981-990.
[6] . 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.
[7] "Top 20 books of 2010 so far," http://goo.gl/LEzyL.decisions".
[8] "Top 100 newspapers in the united states," http://goo.gl/t

ISSN: 2278 – 1323