ENRICHING THE EXTRACTION OF TOP-K LISTS FROM THE WEB

Shreya U. Wadkar[1], Prof. Nilesh G. Pardeshi[2]

PG Student, Computer Engineering Department, SRESCOE, Kopargaon, India.shreewadkar14@gmail.com[1]

Assistant Professor, Computer Engineering Department, SRESCOE, Kopargaon, India. ngpardeshi@gmail.com[2]

ABSTRACT

The web contains a tremendous amount of data and this data result in to big amount of information. This information on the web is of two type i) Structured data and ii) Unstructured data. In this, we concentrate on structure data. List data is most effective source of structure data for extracting the information from the web. This System deals with "Top-k Lists", web pages that represent a list of k instances of a particular topic or concept. Examples are, "top 10 cricketers in the world", "10 best dancer in the world" etc. Top-k lists are bigger, ranked and of good quality source of information. As a result, top-k lists are highly profitable .In this, we present an efficient method that obtain the target lists from web pages with high exactness. Compared to other structured data, top-k lists are clearer, easier to understand and more exciting for human use, and therefore are valuable source for knowledge mining and information finding. Extraction of such lists can help enrich existing knowledge bases about general concepts and significant as a preprocessing step to produce facts for a fact answering engine.

Keywords: - *Web information extraction, Top-k lists, List extraction, Web mining.*

1. INTRODUCTION

Now a days, web becomes the largest source of information. Web information is unstructured text in natural languages and acquiring knowledge from such natural language text is not easy. But still, some of the web information exists in structured or semi-structured forms. Lists or web tables coded with specific tags such as <ul>, <li>, and <table> on html pages are the examples of these forms. Structured information is valuable to acquire the knowledge easily. As a result, recently, many researchers have concentrate on acquiring knowledge from structured information on the web, particularly, from web tables [1], [2], [3], [4], [5], [6], [7].

Still, it is not clear about how much useful and valuable information we can get from lists and web tables. It is sure that the total number of web tables are tremendous in the entire corpus, but only a small percentage of them contains valuable information and very little percentage of them contains information interpretable without context. Particularly, based on our knowledge, more than 90 percent of the tables are utilize to organize the content on the web. Additionally, a lot of the remaining tables are not "relational. We are only concentrating on relational tables because they are interpretable, with rows as entities, and columns as attributes of those entities. As per Cafarella et al. [2], 1.1 percent of all web tables that that are relational, various are useless without context. For instance, consider we derived a table having 5 rows and 2 columns, also have the 2 columns labeled "Companies" and "Revenue" respectively. But, it is not clear why these 5 companies are merge together (e.g., are they the most profitable, most inventive, or most employee favorable companies of a specific industry, or in a particular area?), and how we should know their revenues (e.g., in which year and in what currency).

We are not known of the extract situations under which the extracted information is beneficial. So, understanding the context is very useful for extracting the information. Sometimes, the context is represented in unstructured text format that machines are not able to interpret. Rather than concentrating on structured data (such as tables) and ignoring context, this paper concentrate on the context that we understands, and then we utilize the context to interpret less structured or nearly free-text information, and guide its extraction.

Specifically, we concentrate on a beneficial and valuable source of information on the web, which we call top-k web pages. A top-k web page represents k items of particular interest. In most cases, the representation is in natural

language text which is not directly machine interpretable, although the representation has the similar format or style for different items. But most importantly, the title of a top-k page usually discloses the context, which makes the page interpretable and extractable. Some typical titles are:

1) 20 Most Influential Scientists Alive Today

2) .net Awards 2011: top 10 podcasts

The title of a top-k page contains at least three pieces of important information:

a) A number k, for example, 20, Twelve, and 10 in the above example, which show how many items are described in the page.

b) A topic or concept the items belong to, for example, Scientists, and podcasts.

c) A ranking criterion, for example, Influential, Interesting, and You Shouldn't Miss (which is equivalent to Best or Top). Ranking criterion is not always given implicitly, in which case we make it identical to the "Best". Apart from these 3 components, some top-k titles include two optional pieces of information: time and location.

Top-k system has been develop to find out top-k lists from a web that contains billions of pages. Top k list is linked with very high quality and key information, specifically evaluate with web tables, it contain tremendous amount of high quality information. Furthermore, top k lists are related with the context which is more valuable and accurate in quality analysis, search and other systems.

2. LITERATURE SURVEY

M.J.Cafarella, E. Wu, A. Halevy, Y. Zhang, and D. Z. Wang [2] described the Web Tables system, which is the first large-scale attempt to acquire and leverage the relational information embedded in HTML tables on the Web. This system acquire web lists or tables on very specific list-related tags, such as <ul>,<ol>, <dl>, and <table>.

B. Liu, R. L. Grossman, and Y. Zhai, [3] it describes mining data records from web page which proposed to acquire data records of the similar kind based on the similarity between DOM trees, which is calculated by edit distance. A tremendous amount of information on the web is presented in regularly structured objects. A list of such objects in a Web page usually describes a list of similar items .Mining data records is beneficial because it allows us to integrate information from more than one sources to provide value-added services.

G. Miao, J. Tatemura, W.-P. Hsiung, A. Sawires, and L. E. Moser [4] this introduces a method for record extraction that captures a list of objects in a more robust way based on a holistic analysis of a Web page. This method concentrate on how a distinct tag path appears repeatedly in the DOM tree of the Web document. Rather than comparing a pair of individual segments, it compares a pair of tag path occurrence patterns (called visual signals) to calculate how likely these two tag paths represent the similar list of objects. This method presents a similarity measure that captures how intently the visual signals appear and interleave. Clustering of tag paths is then performed based on this similarity measure, and group of tag paths that form the structure of data records are extracted.

W. Gatterbauer [5], represents an abnormal and promising approach towards organized information extraction from the Web, specifically, from web tables. This approach uses a model of the aesthetic illustration of web pages as made by a web browser and, therefore, changes the issue of information extraction from the lower degree of rule model like HTML tag structure, CSS, JavaScript rule, etc. to the top degree of aesthetic functions like 2-D topology and typography.

Chang, C-H., Lui, S-L.[6]IEPAD, a system that automatically discovers extraction rules from Web pages. The system can automatically recognize record boundary by repeated pattern mining and multiple sequence alignment. The discovery of repeated patterns are identified through a data structure call PAT trees. Additionally, repeated patterns are further proceed by pattern alignment to comprehend all record instances. Based on this system extract data records. This new track to IE involves no human effort and content-dependent heuristics.

3. PROBLEM DEFINITION

To develop a system for extraction of top-k instances of a topic from the web pages with listing of top instances of different topics to improve the performance and effectiveness of searching over the web.

Let a web page be a pair (t, d) where t is the page title, and d is the HTML body of the page. A page (t, d) is a top-k page if:

1) 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, m is a ranking criterion, t is temporal information, l is location information.

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 a is a taxonomy.

b) The pair wise syntactic similarity of the k items is greater than a threshold.

3) The top-k extraction problem can be defined in terms of three functions:

a) Title recognition F1: (t, d) = (k, c, m, t, l).
b) List extractor F2: (k, c, d) = I where I is the set of terms which are instance of c and |I|=k.
c) Content Processor F3: (c, d, I) = (T, S) where T is a table of attribute values for the elements in T and S is its schema.

4. EXISTING SYSTEM

Many approaches have been reported in the literature to extract lists or tables from the web. None of them targets the "top-k" list extraction. Most of the methods are based on either very specific list-related tags [2] 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 inflexible because of the dynamic and inconsistent nature of web pages. More recently, several groups have attempted to use visual information in HTML in information extraction. Most notably, G.Miao [4] were designed to compare the rendered visual model or features with the corresponding DOM structure and achieved remarkable improvements in performance. However, these techniques indiscriminately extract all elements of all lists or tables from a web page, therefore the objective is different from that of this work which is to get one specific list from a page while filtering all other lists as noise. The latter poses different challenges such as distinguishing ambiguous list boundaries and identifying unwanted lists.

5. PROPOSED SYSTEM

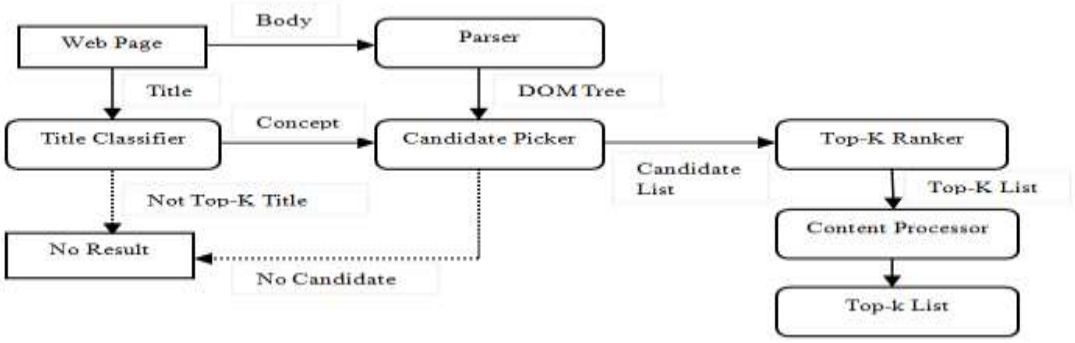

Figure-1: System Overview

This system aims to find out top-k lists from web that contains millions of pages. Top k list is linked with very high quality and key information, particularly evaluate with web tables, it contain large amount of good quality information.Fig.1 shows the Top-K system, which consist of following modules.
1) Title Classifier, which attempts to recognize the page title of the input web page.
2) Candidate Picker, which extracts all potential top-k lists from the page body as candidate lists.
3) Top-K Ranker, which scores each candidate list and picks the best one.
4) Content Processor, which post processes the extracted list to further produce attribute values, etc.

5.1 Title Classifier

The title of a web page helps us identify a "top k" page. This title is enclosed in <title> tag in HTML page. The aim of the classifier is to identify "top-k like" titles, the probable name of a top-k page. This title represents the topic of "top-k" list.Fig.2 shows the flow of an title classifier.

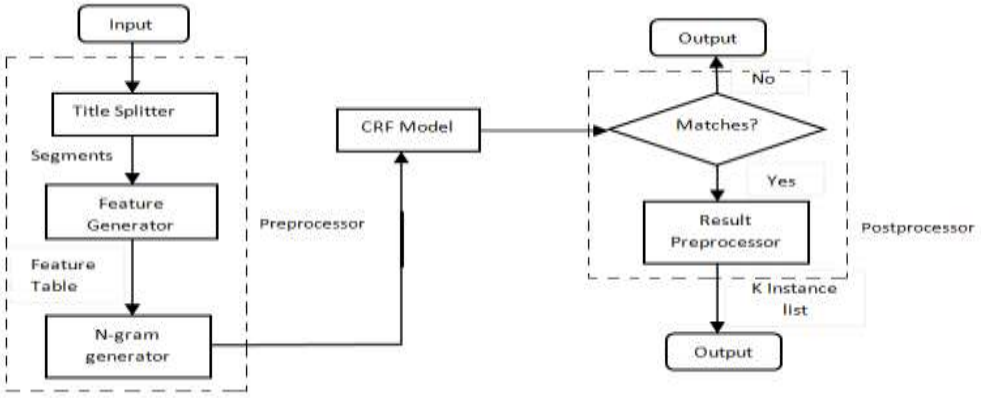

Figure-2: Flowchart of Title Classifier

A "top-k like" title contains more than one segments, which are separated by separator. From this segments only one segment describes topic of the page and rest of the segments shows the additional information. We therefore split the title in to number of segments. We trained a Conditional Random Fields (CRF) [17] model from both positive and Negaive sample titles to recognize "top-k like" title. The classifier also transfers the cardinal digit word (word like ten or fifteen) into the number k, and outputs a set of concepts which are mentioned in the title.Fig.3 shows a sample of "top-k" title.

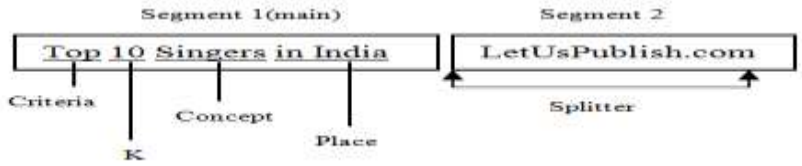

Figure-3: Sample of Top-K Title

5.2 Candidate Picker

Candidate picker collects a set of lists as a candidates using HTML page body and the number k. Each item in the list is a text node in the page body. Structurally list is presented as a list of HTML nodes with identical tag path. A tag path can be defined as the path from root node to a certain tag node in the DOM tree, which can be presented as a sequence of tag names. Note that there may be more than one list in a given web page of same size. Items in a "top-k" list having similar format and style therefore they share identical tag path. The system use two basic rules for selecting candidate lists:

1) K items: A candidate list should contain exactly K items.

2) Identical tag path: The tag Path of each item node in a candidate list should not be different.

Tag path clustering method is used for extracting the candidate lists from input page. This method determines the tag path for each node. Tag path clustering method, shown in Algorithm 1.This algorithm process the web page according to the above two basic rule. This algorithm recursively calculate the tag path for all node within HTML page, and collect text nodes with identical tag path into one node lists. Node lists contain exactly k nodes which get selected in candidate set.

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 check
4) if table contain the tag path of node n
then
5) list = table[n.TagPath]
6) else
7) create empty list
8) list = new empty list
9) table[n.TagPath]=list
10) insert n in to list
11) for each node i compute the tag path
12) TagPathClustering(i, table)
13) end

5.3 Top-k Ranker

As there are multiple candidate lists, we usually select only one of them as the main list. Top-k Ranker positions the candidate lists and pick the top positioned list as the "top k list". This is to be done by utilizing scoring function, a weighted sum of two feature scores below:

1) P-Score: P-Score measure the correlation between the "top-k" list and title. P-score of a list can be determine as:

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

Where LMI (n) is the longest matched instance in the text of node (n) and Len (n) is the length of the entire text in node n.

2) V-Score: V-Score determines the visual area occupied by a list, as the main list of the page tends to be enormous and more important compared to other minor lists. V-score of a list can be determine as:

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

5.4 Content Processor

Content Processor takes the input as a "top k" list and extract main items as well as their attributes. Content processor give us structure information for each item in the list. Sometimes the text node within HTML represent each item which may have structure itself or is semi structure. For Example: Painted by Picasso. The content processor infers the structure of the text by building a histogram for all separator tokens such as "By", ":" and "," from all the items of the "top-k" list. If we find a sharp spike in the histogram for a particular token, then we successfully find a separator token, and we use that token to separate the text into more than one fields.

6. MATHEMATICAL MODEL

6.1 Set Theory

Let, S = {I; P; IO; O}
Where,
S: Top-k system
I: Set of inputs
P: Set of processes
IO: Intermediate outputs
O: Set of outputs/Final output
1) I = {i}
Where,

i: Web page.
2) P = {p1; p2; p3; p4}
Where,
p1: Recognizing the title of web page.
p2: Extracting the candidate list from web page.
p3: Ranking the extracted candidate list.
p4: Extracting attribute of each item.
3) IO = {io1; io2; io3; io4}
Where,
io1: Title is recognized.
io2: Candidate list get extracted.
io3: Candidate list is ranked.
io4: Attribute of each item in list get extracted.
4) O = {o}
Where,
o: Extracted Top-k list from web page.

6.2 Venn Diagram

Venn diagram shows the mapping of the input, process and output relation of the system. It also represent the interaction between different processes along with input and output.

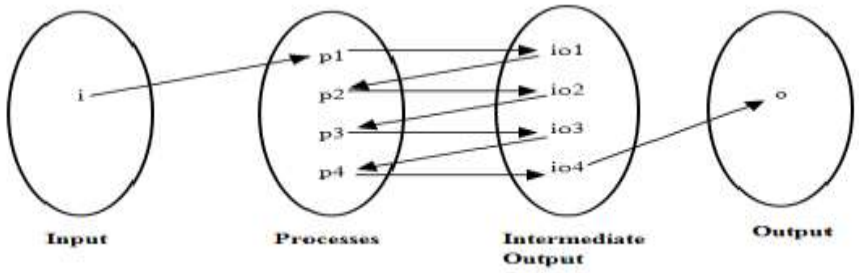

Figure-4: Venn diagram

6.3 Process State Diagram:

Here, process p1, p2, p3 and process p4 are denoted by Q0, Q1, Q2 and Q3 respectively. Where Q4 is a final state that is top-k list as shown in Fig.5.

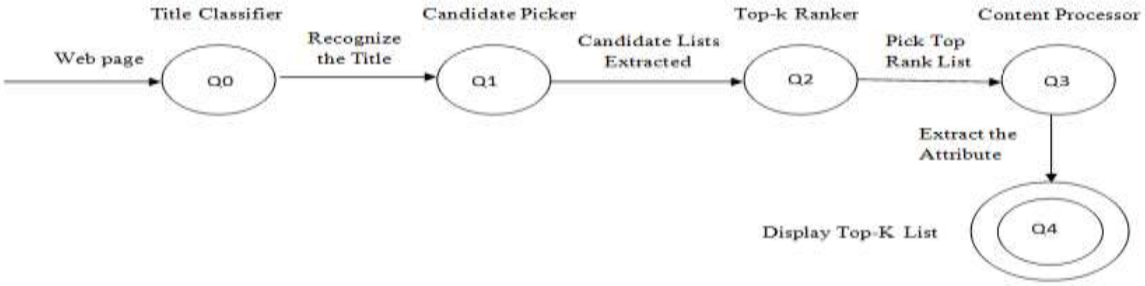

Figure-5: Process State Diagram

7. EXPERMENTAL RESULTS

Input to the system is a web page. First we browse the dataset and select the web page. Selected web page get parse by system and title of selected web page get extracted which is enclose in tag <title> in HTML page .For example consider the title of web page is .net Awards 2011: top 10 podcasts. System check the title whether it is top-k or not as shown in fig.6. System generate the feature table of an given title as shown in fig7.System extract the candidate list and top-k list of given web page as shown in fig8. we conducted these experiments on a 4GB RAM PC and 2.70GHz Dual-Core Intel CPU with 42 top-k pages and 10 not top-k pages. These top-k pages included 'k' in it and not top-k pages were without 'k' so no output was displayed for such pages.

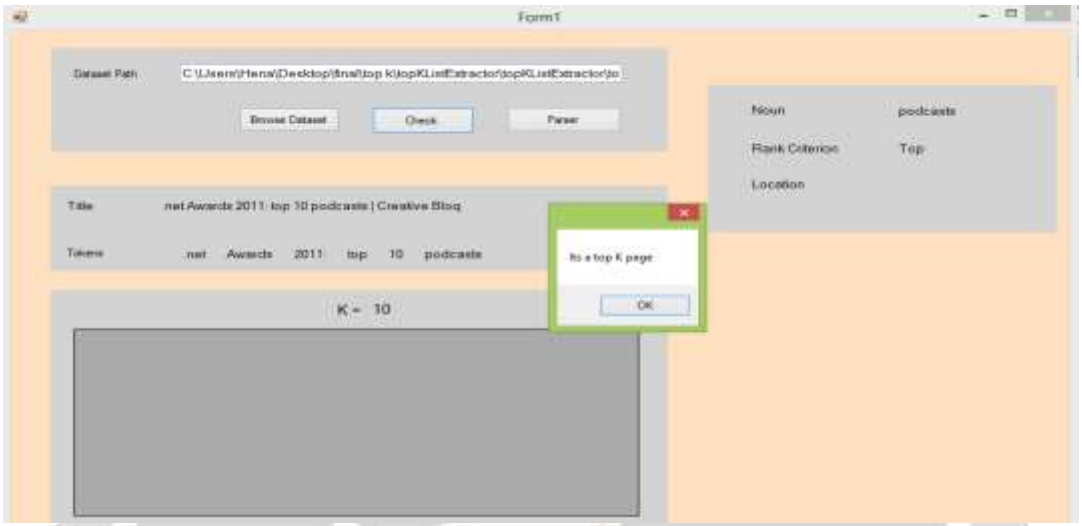

Figure-6: Extraction of Title and Identification of Title

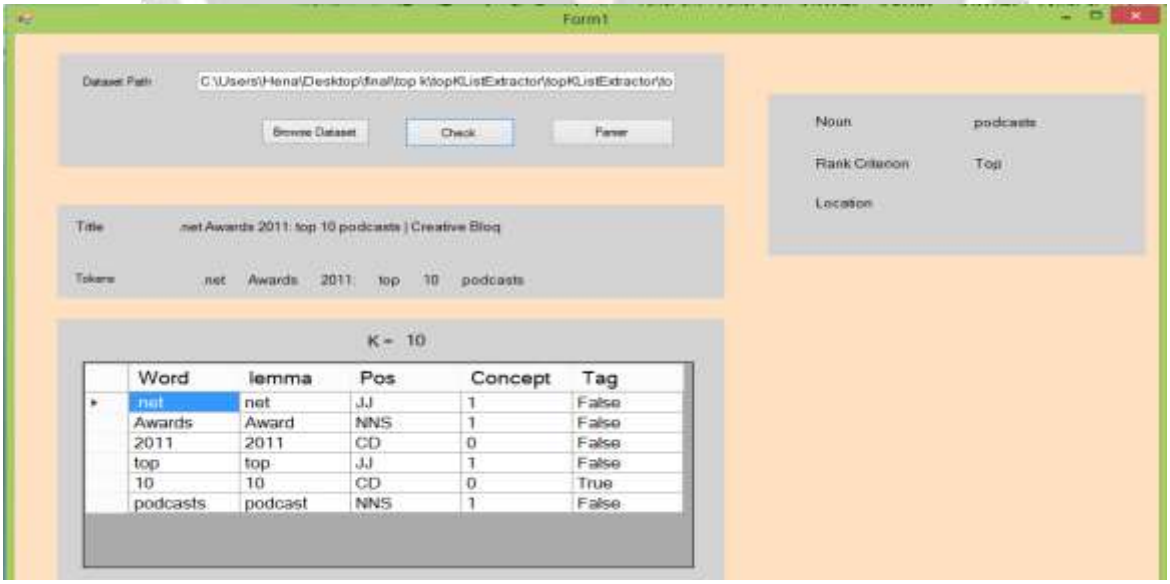

Figure-7: Generation of Feature Table

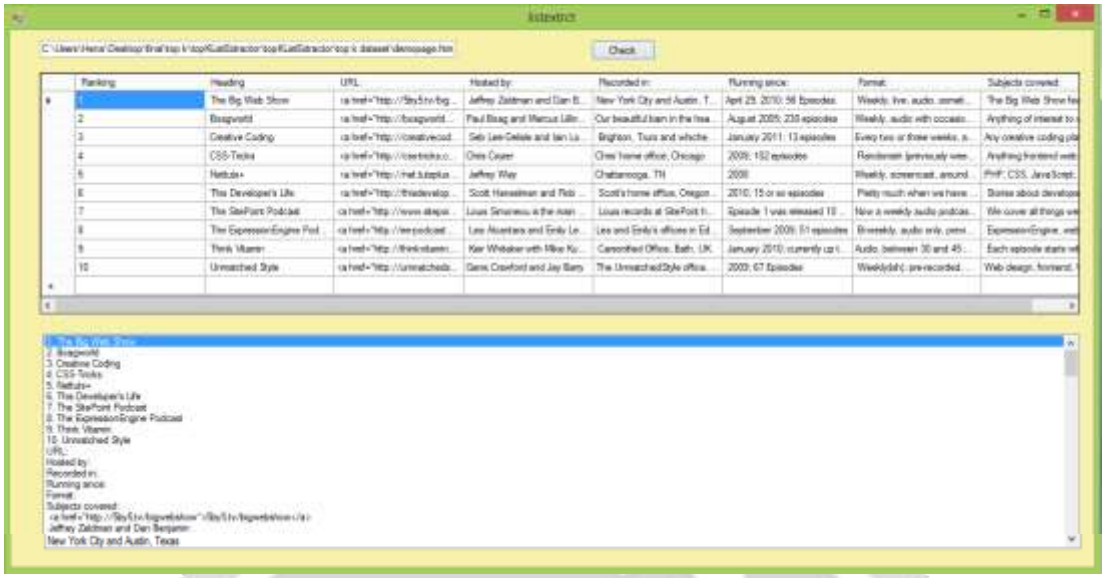

Figure-8: Extraction of Candidate and Top-K List

Table 1: Result table

Sr. No	Title	Output
1.	10 best Hindi films of 2015	List of 10 best Hindi films of 2015
2.	Top 10 famous English writer in India	List of 10 famous English writer in India
3.	Top 10 richest cities in India	List of 10 richest cities in India
4.	List of Doctors	No output
5.	Jaguar cars	No output

8. PERFORMANCE EVALUATION

The performance of the systemcan be measured in terms of its recall, precision. Precision measures the ability of the systemto extract all the list, while Recall measures the ability of the system to extract only the list that are correct. They are defined as:

$$Precision = \frac{Number\ of\ webpages\ from\ which\ correct\ list\ extracted}{Total\ number\ of\ webpages\ from\ which\ list\ extracted} = \frac{A}{A+B}$$

$$Recall = \frac{Number\ of\ webpages\ from\ which\ correct\ list\ get\ extracted}{Total\ number\ of\ top-k\ webpages} = \frac{A}{A+C}$$

Where A represent the number of web pages from which correct list get extracted, B, the number web pages from which correct list not extracted C, number of web pages in which correct list are present but not get extracted.

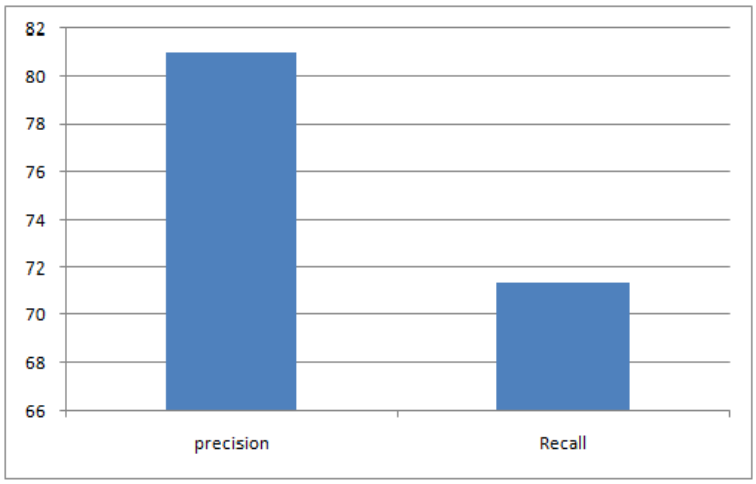

Figure-9: Graph of Precision and Recall for list extraction

9. CONCLUSION

This paper represents a novel and exciting approach of extracting top-k provides from the web. Compared to other structured data, top k lists are clearer, easier to understand and more exciting for human use, and therefore are significant source for knowledge mining and information finding.

The framework accomplishes an 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.

10. ACKNOWLEDGEMENT

I dedicate all my works to my esteemed guide Prof. N. G. Pardeshi, whose interest and guidance helped me to complete the work successfully. This experience will always steer me to do my work perfectly and professionally. I also extend my gratitude to Prof. D. B. Kshirsagar H.O.D. of Computer Engineering Department and Prof. P. N. Kalavadekar P. G. Coordinator who has provided facilities to explore the subject with more enthusiasm. I express my immense pleasure and thankfulness to all the teachers and staff of the Department for their co-operation and support.

11. REFERENCES

[1] Zhixian Zhang, Kenny Q. Zhu, Haixun Wang, Hongsong Li, "Automatic Extraction of Top-k Lists from the Web,"in 2013.

[2] 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.

[3] B. Liu, R. L. Grossman, and Y. Zhai, "Mining data records in web pages," in KDD, 2003, pp. 601606.

[4] 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.

[5] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, and B. Pollak, "Towards domain independent information extraction from web tables," in WWW. ACM Press, 2007, pp.7180.

[6] C.-H. Chang and S.-C. Lui, "Iepad: information extraction based on pattern discovery," in WWW, 2001, pp. 681688.

[7] U. Guntzer, W. Balke, and W. Kieling, "Optimizing multi-feature queries for image databases," in VLDB, 2000, pp. 419428.

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

[9] 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.

[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in ICML, 2001.

[11] Z. Zhang, K. Q. Zhu, and H. Wang, "A system for extracting top-k lists from the web," in KDD, 2012. 27