

A System for Extracting Top-K Lists from the Web

Zhixian Zhang
Shanghai Jiao Tong University
Shanghai, China
zzx1989@sjtu.edu.cn

Kenny Q. Zhu^{*}
Shanghai Jiao Tong University
Shanghai, China
kzhu@cs.sjtu.edu.cn

Haixun Wang
Microsoft Research Asia
Beijing, China
haixunw@microsoft.com

ABSTRACT

List data is an important source of structured data on the web. 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” and “the 50 hits of 2010 you don’t want to miss”. Compared to normal web list data, “top- k ” lists contain richer information and are easier to understand. Therefore the extraction of such lists can help enrich existing knowledge bases about general concepts, or act as a pre-processing step to produce facts for a fact answering engine. We present an efficient system that extracts the target lists from web pages with high accuracy. We have used the system to process up to 160 million, or 1/10 of a high-frequency web snapshot from Bing, and obtained over 140,000 lists with 90.4% precision.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.2.8 [Database Management]: Database Applications—Data Mining

Keywords

Web information extraction, top- k lists, list extraction, web mining

1. INTRODUCTION

The world wide web is by far the largest source of information today. Much of that information contains structured data such as tables and lists which are very valuable for knowledge discovery and data mining. This structured data is valuable not only because of the relational values it contains, but also because it is relatively easier to unlock information from data with some regular patterns than free text which makes up most of the web content. However, when encoded in HTML, structured data becomes *semi-structured*. And because HTML is designed for rendering in a browser, different HTML code segments can give the same visual effect at least to the human eye. As a result, HTML coding is much less stringent than XML, and inconsistencies and errors are abundant in HTML

^{*}This work was partially supported by NSFC Grant 61100050 and MOE New Faculty Award No. 20110073120023.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD’12, August 12–16, 2012, Beijing, China.

Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...\$15.00.

documents. All these pose significant challenges in the extraction of structured data from the web [14].

In this demo, we focus on list data in web pages. In particular, we are interested in extracting from a kind of web pages which present a list of k instances of a topic or a concept. Examples of such topic include “20 Most Influential Scientists Alive Today”, “Ten Hollywood Classics You Shouldn’t Miss”, and “50 Tallest Persons in the World”. We call these pages “top- k ” pages. Figure 1 shows one such “top- k ” page [1]. Figure 1.(a) is a snapshot of the page and Figure 1.(b-d) are some of its segments. The title (Figure 1.(b)) of a “top- k ” page usually contains a number k indicating the list size (20), as well as the head word/phrase (e.g., scientist) which best describes the entities in the list. Figure 1.(c) shows one instance (element) in the list, which not only contains the instance name (Timothy J. Berners-Lee), but also optionally additional information like a picture, a textual description and a link to a relevant wikipedia page. The additional information can be treated as the attributes of the instance. A “top- k ” page can also contain unwanted lists such as Figure 1.(d) which should be filtered out.

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, if any. Table 1 shows the sample output from the page shown in Figure 1.¹

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 [4] such as ``, `` and `<table>` or the similarity between DOM trees [9, 10] 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. More recently, several groups have attempted to utilize visual information in HTML in information extraction. Most notably, Vortex [7] and HyLiEn [6] were designed to correlate 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 extract *one* specific list from a page while purging all other lists (e.g. (d) in Figure 1) as noise. The latter poses different challenges such as distinguishing ambiguous list boundaries and identifying unwanted lists.

We target “top- k ” list data for information extraction for the following reasons. First, there are *large* amount of “top- k ” lists around on the web. We estimate that the total number in Bing’s

¹The actual output is stored in XML format and includes additional information.



Figure 1: Snapshot of a typical “top- k ” page [1] and its page segments

Index	Name	Image	Description	Wiki. Link
1	Timothy J. Berners-Lee	tim-berners-lee_1366736c.jpg	who invented the World Wide Web...	[link]
2	Noam Chomsky	noam_chomsky.jpg	who, though a linguist and philosopher...	[link]
3	Richard Dawkins	richard_dawkins.jpg	whose use of evolutionary biology has shaped...	[link]
...
20	Edward Witten	edward_witten.jpg	whose work on the mathematical underpinnings...	[link]

Table 1: Sample extraction output of “20 Most Influential Scientists Alive Today” [1]

corpus is around 2.24 million (1.4% of total number of pages), and our system can effectively extract up to 75.5% of them. The scale of this data is larger than any manually or automatically extracted lists in the past.

Second, list data is generally *cleaner* than other forms of web data. Free text contains a lot of variation and ambiguity and is known to be hard to understand and extract. General tables, even though structured, can have many different forms (such as row span and column span) and styles, and many of them are not meaningful if we don’t know the schema of the table or the meaning of the headers [12]. Lists, on the other hand, have relatively simpler structures and are easier to identify. What’s more, “top- k ” lists, with their unique semantics, are even cleaner than ordinary lists.

Third, “top- k ” lists are relatively *easier* to understand. “top- k ” list pages share a common style: the title contains a number and the topic or concept of the list. Each list item can be considered as an instance of the page title. The number of items should be equal to the number mentioned in the title. Besides the name of the instance, each list item may contain additional attributes of the instance.

Finally, “top- k ” lists have *interesting* semantics. The fact that the list items are called “top XXX” means that these items are more important, popular or meaningful than an arbitrary list. What’s more, people are always fascinated about rankings. Information of this sort is likely to find a large audience.

We deployed our prototype system on a distributed computing platform and performed extraction on up to 1/10 of a high frequency web snapshot crawled by Bing. Our preliminary results showed that the system achieved 90.4% precision and 57.7% recall. That amounts to the correct extraction of 129,169 lists from a total of 160 million randomly selected web pages.

The work described in this paper is an important step in our bigger effort of automatic constructing a universal knowledge base that

includes large number of known concepts and their instances. To that end, we have already built one of the largest open-domain taxonomy called Probase [15, 13, 12, 11] which consists of 2.7 million concepts and many more instances. The “top- k ” lists we extracted from the web can be an important source for enriching Probase’s instance space. Also, our system enables the construction of an effective fact answer engine [16]. With such an engine, we can answer queries such as “Who are the 10 tallest persons in the world”, or “What are 50 best-selling books in 2010” directly, instead of referring the users to a set of ranked pages like all search engines do today.

Next we will briefly discuss the framework of our system (Section 2) and the preliminary evaluation results (Section 3), and present a plan for demonstration (Section 4).

2. TECHNICAL SPECIFICATION

Figure 2 shows the block diagram of our system. As the input of the system, the web page is first parsed by a HTML parser[3] to obtain a complete DOM representation. Then the title classifier attempts to recognize the page title. If it is a “top- k like” title, the classifier outputs the list size (the number k) and a set of possible concepts mentioned in the title. With the number k , the candidate picker extracts all lists of size k from the page body as candidate lists. Only one of them will be the actual list of interest. With the concept set, the top- k ranker can score each candidate list and pick the best one as the “top- k ” list. Finally the content processor analyzes the list content and extracts the entity names and attributes.

2.1 Title Classifier

The title of a web page (string enclosed in <title> tag) helps us identify a “top- k ” page. The goal of the classifier is to recognize “top- k like” titles, the likely name of a “top- k ” page. In general,

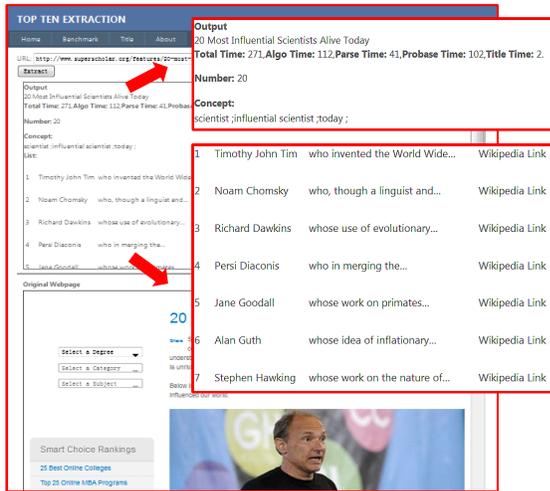


Figure 3: Web Demo GUI: TryItOut

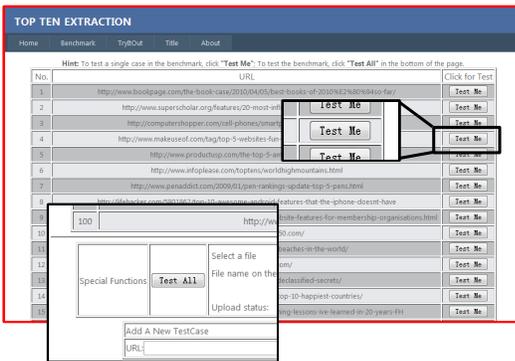


Figure 4: Web Demo GUI: Benchmark

and whether it is a “top- k like” title. A screenshot of this section is shown in Figure 5.

5. CONCLUSION

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. With the massive knowledge stored in those lists, we can enhance the instance space of a general purpose knowledge base such as Probase. It is also possible to build a search engine for “top- k ” lists as an effective fact answering machine. Our proposed 4-stage extraction framework has demonstrated its ability to retrieve large number of “top- k ” lists at a very high precision.

6. REFERENCES

[1] 20 most influential scientists alive today.
<http://goo.gl/KbB90>.



Figure 5: Web Demo GUI: Title

[2] Top-k web demo. <http://202.120.38.145:8088/TopTenSite/Default.aspx>.

[3] Winista htmlparser. <http://www.netomatix.com>.

[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] K. Fisher, D. Walker, K. Q. Zhu, and P. White. From dirt to shovels: Fully automatic tools generation from ad hoc data. In *ACM POPL*, 2008.

[6] F. Fumarola, T. Weninger, R. Barber, D. Malerba, and J. Han. Extracting general lists from web documents: A hybrid approach. In *IEA/AIE (I)*, pages 285–294, 2011.

[7] 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.

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

[9] B. Liu, R. L. Grossman, and Y. Zhai. Mining data records in web pages. In *KDD*, pages 601–606, 2003.

[10] 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*, pages 981–990, 2009.

[11] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, 2011.

[12] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu. Understanding tables on the web. In *ER*, 2012.

[13] Y. Wang, H. Li, H. Wang, and K. Q. Zhu. Toward topic search on the web. In *ER*, 2012.

[14] T. Weninger, F. Fumarola, R. Barber, J. Han, and D. Malerba. Unexpected results in automatic list extraction on the web. *SIGKDD Explorations*, 12(2):26–30, 2010.

[15] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, 2012.

[16] X. Yin, W. Tan, and C. Liu. Facto: a fact lookup engine based on web tables. In *WWW*, pages 507–516, 2011.