ContentEx: A Framework for Automatic Content Extraction Programs

Linhai Song[1,2], Xueqi Cheng[1], Yan Guo[1], Yue Liu[1], Guodong Ding[1]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing
2. Graduate School of the Chinese Academy of Sciences, Beijing
songlinhai@software.ict.ac.cn, cxq@ict.ac.cn, guoy@ict.ac.cn,
liuyue@ict.ac.cn, dingguodong@software.ict.ac.cn

***Abstract*—Web pages are often decorated with extraneous information (such as navigation bars, branding banners, JavaScript and advertisements). This kind of information may distract users from actual content they are really interested in and may reduce effects of many advanced web applications. Automatic content extraction has many applications ranging from providing data for web mining to realizing better accessing the web over mobile devices. In this paper, we propose ContentEx, a framework for automatic content extraction programs, which we use to organize codes of automatic content extraction programs and to facilitate the development of related solutions. We also introduce how we extract content from forum pages in this framework to fulfill the requirement from our actual application.**

Keywords- automatic content extraction; ContentEx

I. Introduction

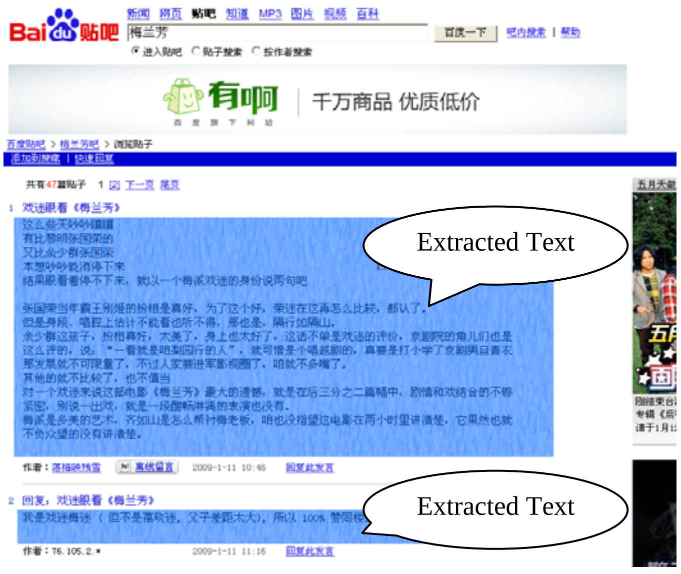
Web pages are often decorated with extraneous information which includes navigation bars, branding banners, JavaScript and advertisements. This kind of information distracts users from actual content they are interested in. Automatic content extraction from web pages has many applications such as providing data for web mining, enabling the visually impaired to access the web more easily, and helping access the web over mobile devices like PADs and cellular phones.

Since web pages have become an important source of information, there is a growing concern over content extraction among researchers in Informatics. For example, we need to find hot topics from forum pages in our work. Formerly, html tags are removed and remainders are used as input of a TDT (Topic Detection and Tracking) algorithm. Apparently, there are so many noises in the input, and these noises mislead the TDT algorithm. So, we turn to content extraction to help solve this problem.

Figures 1 shows an example of the problem we encounter. The parts marked Extracted Text are targets which we want to extract in our work.

Many previous methods to realize information extraction are based on wrappers, and these methods introduce a lot of work to produce and manage wrappers. To avoid such troublesome work from methods based on wrappers, we choose automatic content extraction algorithms as our solution. The input of this kind of algorithms is only one page, and the output is got based on heuristic rules designed in these algorithms. Web pages are so different that distinct methods may be needed to get ideal results. So how to organize and reuse related codes is a new problem we encounter. In this paper, we propose ContentEx, a framework for automatic content extraction programs, which we use to solve the new problem. And we expect that the framework and pre-defined modules implemented in this framework can facilitate the development of other related solutions. The method we use to extract content from forum pages is also discussed in this paper. Experimental results show that the method can fulfill our requirement.

This work was supported by the National Basic Research Program of China (973 Program) (No. 2004CB318109).


新闻 网页 贴吧 知道 MP3 图片 视频 百科
梅兰芳
百度一下
吧内搜索 | 帮助
进入贴吧 贴子搜索 按作者搜索
有啊 千万商品 优质低价
百度旗下网站
百度贴吧 > 梅兰芳吧 > 浏览贴子
添加到搜藏 | 快速回复
共有47篇贴子 1 [2] 下一页 尾页
Extracted Text
Extracted Text

1 戏迷眼看《梅兰芳》

这么些天吵吵嚷嚷
有比黎明张国荣的
又比余少群张国荣
本想吵吵能消停下来
结果眼看着停不下来，就以一个梅派戏迷的身份说两句吧

张国荣当年霸王别姬的扮相是真好，为了这个好，荣迷在这再怎么比较，都认了。
但是身段、唱腔上估计不能看也听不得，那也是，隔行如隔山。
余少群这孩子，扮相真好，太美了，身上也太好了，这话不单是戏迷的评价，京剧院的角儿们也是
这么评的，说：“一看就是咱梨园行的人”，就可惜是个唱越剧的，真要是打小学了京剧男旦青衣
那发展就不可限量了，不过人家要进军影视圈了，咱就不多嘴了。
其他的就不比较了，也不值当
对一个戏迷来说这部电影《梅兰芳》最大的遗憾，就是在后三分之二篇幅中，剧情和戏结合的不够
紧密，别说一出戏，就是一段酣畅淋漓的表演也没有。
梅派是多美的艺术，齐如山是怎么帮衬梅老板，咱也没指望这电影在两小时里讲清楚，它果然也就
不负众望的没有讲清楚。

作者：落梅映残雪 离线留言 2009-1-11 10:46 回复此发言

2 回复：戏迷眼看《梅兰芳》

我是戏迷梅迷（但不是葆玖迷，父子差距太大），所以 100% 赞同楼

作者：76.105.2.* 2009-1-11 11:16 回复此发言

Figure 1. The shaded areas are the parts we want to extract from forum pages

II. Architectrue of ContentEx

ContentEx is a framework for automatic content extraction programs. Content extraction algorithms implemented in this framework share two common characteristics:

- The input is only one page. No structure specific wrappers are required and no training stage is used. The algorithms combine several heuristic rules to extract content.

- These algorithms present each input page as a DOM tree.

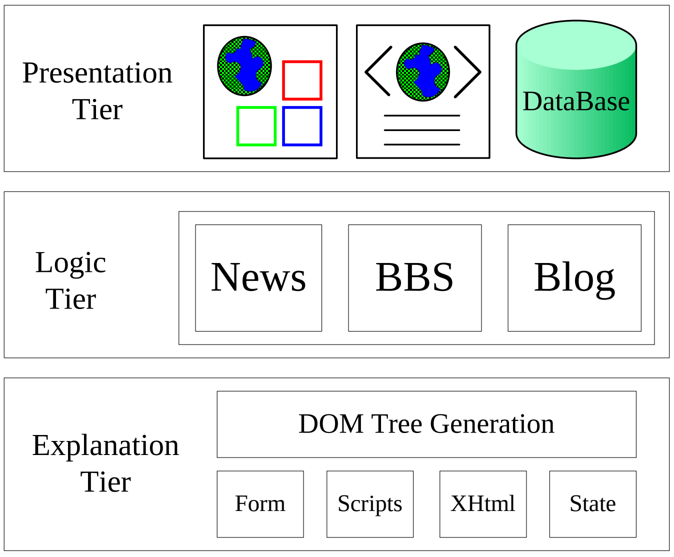


Figure 2. Architecture of ContentEx

Figure 2 shows the architecture of ContentEx. There are three layers in ContentEx: Explanation Tier, Logic Tier and Presentation Tier. Functions of them are discussed as follows:

A. *Explanation Tier*

Input pages are preprocessed and changed into DOM trees in this layer. We have provided four preprocessing functions in ContentEx: *Ignoring Form tags*, *Removing scripts*, *Converting to XHTML* and *Removing statements*. The first three functions are proposed in [1], and our experiment shows that *Removing statements* can also improve qualities of results.

The four preprocessing functions are explained as follows:

- *Ignoring Form tags*: Some tags are used to perform interaction between users and web pages, and these tags are classified as form tags in [1]. Usually these tags don't contain content.
- *Removing scripts:* Scripts are codes executed by browsers, and they are not content we care.
- *Converting to XHMTL*: This function is used to generate error-free web pages.
- *Removing statements*: Statements includes statements about copyright, statements about disclaimer and statements about posting messages complying with local laws.

B. *Logic Tier*

We realize content extraction algorithms in this layer. Ideally, one generic method can be found to accomplish content extraction from news pages, forum pages and blog pages, which are all important information sources on the web. But layouts and design criteria of these three kinds of pages are so different that different methods may be needed to get ideal results. The method we use to extract content from forum pages is discussed in Section 3.

C. *Presentation Tier*

Results of content extraction are displayed in this layer. We have published content extraction as a web service, which can be accessed through Internet. And results of content extraction can be stored in XML format or in a database.

III. Algorithm

Considering features of forum pages, we use filters to remove noises in forum pages and use remainders as extraction results. There are two types of filters to extract content from forum pages in ContentEx: link table filter and common pattern filter. They are described as follows:

A. *Link Table Filter*

The link table filter is designed to remove navigation links, advertisements and multiple pieces of related links in each input page. Crunch [2, 3] proposes a link table remover which eliminates table cells with a low text-to-link ratio. It is restricted to pages that use a HTML <table> layout, and is not always effective. In ContentEx, we realize a self-search link table filter, and the algorithm we use in the filter is shown in Figure 5.

```
1   Construct DOM tree $T$ for the input page;
2   for each terminal node $N$ of $T$:
3     if N is a text node:
4       $textCnt(N)$ = word count of text in $N$;
5       $linkCnt(N) = 0$;
6     else if N is a link node:
7       $textCnt(N) = 1$;
8       $linkCnt(N) = 1$;
9     else:
10      $textCnt(N) = 0$;
11      $linkCnt(N) = 0$;
12  for each non-terminal node $N$ of $T$:
13    Initialize $textCnt(N) = 0$ and $linkCnt(N) = 0$;
14    for each child $C$ of $N$:
15      $textCnt(N) = textCnt(N) + textCnt(C)$;
16      $linkCnt(N) = linkCnt(N) + linkCnt(C)$;
17    Calculate
      $Score(N) = \frac{textCnt(N) - linkCnt(N)}{textCnt(N)}$
18  for each node $N$ of $T$ in DFS order:
19    if $N$ is a link node:
20      if $score(parent(N)) < Threshold$:
21        $N = parent(N)$;
22        while $score(parent(N)) < score(N)$:
23          $N = parent(N)$;
24        Delete $N$ from $T$;
```

Figure 3. Link Table filter Algorithm

This algorithm converts each input page into a DOM tree in Step 1. Firstly, it counts the number of words (*textCnt*) and the number of links (*linkCnt*) for each terminal node in Steps 2-11. Secondly, it counts these two numbers for non-terminal nodes in Steps 12-16, and *textCnt* and *linkCnt* of each non-terminal node are the sum of the *textCnt* and *linkCnt* of all its children. It calculates text-to-link ratio scores for non-terminal nodes in Step 17. In Steps 18-24, it traverses through the DOM tree in DFS (Depth First Search) order to find link tables and remove them. The search process of link tables starts from any link node and a link table is found when the score of parent of the link node is below a pre-defined *threshold*. The link table is further extended in Steps 22-23 until the largest possible link table is found. The discovered link table is then removed from the DOM tree in Step 24. Traversing in DFS order guarantees that deleting nodes will not impact traversing. Extending link tables from bottom up will avoid deleting useful information in the input page. Compared with link table remover in [2, 3], our algorithm makes no assumption about any specific tag.

B. Common Pattern Filter

Our extraction target from forum pages is content of posts and content of replies as shown in Figure 1. So other visible parts such as author names, time of post and information of authors are all noises we want to remove from forum pages. In order to improve precision of our extraction, we sum up some patterns of noises frequently appearing in forum pages, and use regular expressions to remove noises following in these patterns. Patterns we provide in ContentEx are discussed as follows:

- *Time pattern*: Forum pages often contain timestamps to indicate when threads are post. We sum up some time patterns to delete text nodes which consist of a few words and these words match time patterns.
- *IP pattern*: Some forum pages show authors' IP addresses which are also common noises.
- *Colon pattern:* We delete text nodes with a colon as their last non-blank word.
- *Copyright pattern:* We delete text nodes containing string "all right reserved". Because these nodes only show copyright messages of input pages, and are not the content we care.
- *Repeated short text pattern:* Repeated short texts are usually generated automatically and are used as navigation messages. So we can remove them.

IV. Experiment

A. Test Data

When dealing with pages produced by the same template, effects of content extraction algorithms are often similar. So we try our best to test the algorithm on pages produced by different templates. Our test pages are from 221 forums. Some of them are produced by popular software (like "Discuz!"), and others are designed by their owners. We select 2 pages from each of these sites. So the total number of pages we use to test how we extract threads from forum pages is 442.

B. Performance Measures

- *Text-based Precision-Recall:* Text-based Precision is defined as the text number of content in extraction results divided by the total text number of extraction results. And Text-based Recall is defined as the text number of content in extraction results divided by the total text number of content. Experience tells us that when text-based precision and recall are both above 90%, content extraction algorithms can fulfill actual requirements. So we choose the baseline of text-based precision-recall of our systems as 90%.
- *Processing Time:* ContentEx is designed to fulfill actual demand. So processing time is also an important evaluation index we must consider.

C. Result

The precision and recall of our forum content extraction algorithm are 93% and 99% respectively. They are both above our baseline. If more filters are used, we may get a higher precision but a lower recall.

J. Prasad and A. Paepcke in [1] propose that *Converting to XHTML* is very expensive and has little effect. So when test *Processing Time*, we don't convert input pages into XHTML. Under this condition, our ContentEx can handle 67 forum pages in one second.

V. Conclusion

In this paper, we propose ContentEx, a framework for automatic content extraction programs, which we use to organize and reuse codes related to content extraction. We expect that by using this framework and pre-defined modules in this framework we can realize solutions of content extraction more easily. Algorithms which can be integrated into ContentEx must be totally automatic and must present each input page as a DOM tree.

In this paper, we also discuss how we realize content extraction from forum pages. Considering features of forum pages, we use filters to extract content. Experimental results show that our solutions can fulfill the requirement from our actual applications.

References

[1] J. Prasad, and A. Paepcke, "CoreEx: Content Extraction from Online News Aritcles," http://ilpubs.stanford.edu:8090/832/

[2] S. Gupta, G. K. Kaiser, P. Grimm, M. F. Chiang, and J. Starren, "Automating Content Extraction of HTML Documents," World Wide Web, 2005, vol. 8, pp. 179-224.

[3] S. Gupta, G. Kailer, D. Neistadt, and P. Grimm, "DOM-based Content Extraction of HTML Documents," In WWW'03: Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, 2003, pp.207-214..