Extract With Markers Pattern

By Shai Ben-Yehuda

Zooming Algorithm to extract data from html

Context

Extracting specific data from web pages is not an easy task. The html can be very big and complex, the pages can vary in html structure, and/or the site itself can change over time.

For example, we want to extract the prices of 1000 books from the amazon web site. However, the html of the page that shows each book is large and complicated, pages of different books may have a different html structure and amazon may change the pages over time.

Problem

How to write the “best” algorithm that will extract data from web pages?

Forces

Rigidness – the code should resist to site changes over time.

Simplicity – the code should be small and readable.

Minimize the effort – minimize coding/test time.

Controllable/configurable – the smarter the algorithm, the harder it is to control its parameters when it does not deliver. [1] [2]

Understanding the html structure can help. [1]

Understanding natural language & AI technique can help. [1] [2]

Conflicting Forces

Minimize Effort & simplicity versus rigidness – the simplest code that does the job may not resist site changes.

Smart algorithms that facilitate natural language understanding and structure can help locating the data automatically with less coding. However, when not providing the right solution it can be hard to configure and guide such algorithms.

Solution

Use a simple & stupid algorithm that ignores the html structure. Focus on textual markers rather than the structure.

Use the markers to build a multi-step ‘zooming’ algorithm to focus on the relevant area at each step.

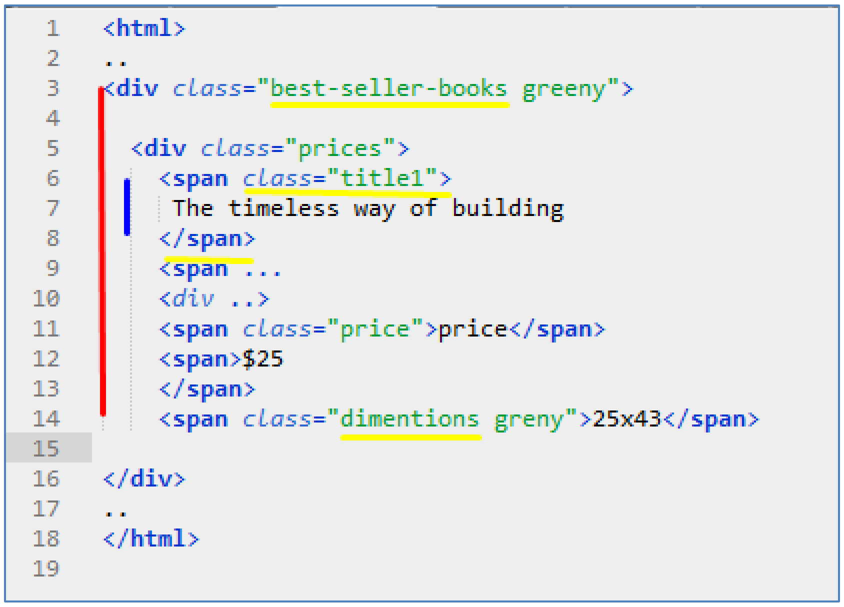
Use ‘begin’ and ‘end’ markets to define the next relevant section for the next iteration.

Try to improve your markers, by finding the markers which seems to be the most stable during time.

In the example below, we want to extract the title.

The first “zoom” is marked the red line and focus on the area between lines 3-14 by providing the markers “**best-seller-books**” and “**dimensions**”.

The next “zoom” is marked with blue and extracts the title by providing the markers **class=”title1”>** and **</span**.



```
 1  <html>
 2  ..
 3  <div class="best-seller-books greeny">
 4
 5    <div class="prices">
 6      <span class="title1">
 7       The timeless way of building
 8      </span>
 9      <span ...
10      <div ..>
11      <span class="price">price</span>
12      <span>$25
13      </span>
14      <span class="dimentions greny">25x43</span>
15
16  </div>
17  ..
18  </html>
19
```

Consequences

‘Extract With Markers’ suggests a simple, small and clean algorithm.

It will be stable if the right markers are chosen.

It is in complete contrary to smart algorithms that may suggest using smart linguistic and classification techniques for the same task.

A More Detailed Example

Let’s explore the extraction of the best seller books details from Amazon.

Goto http://www.amazon.com and search for ‘best sellers’ page. We would like to extract the titles and prices of the best sellers.

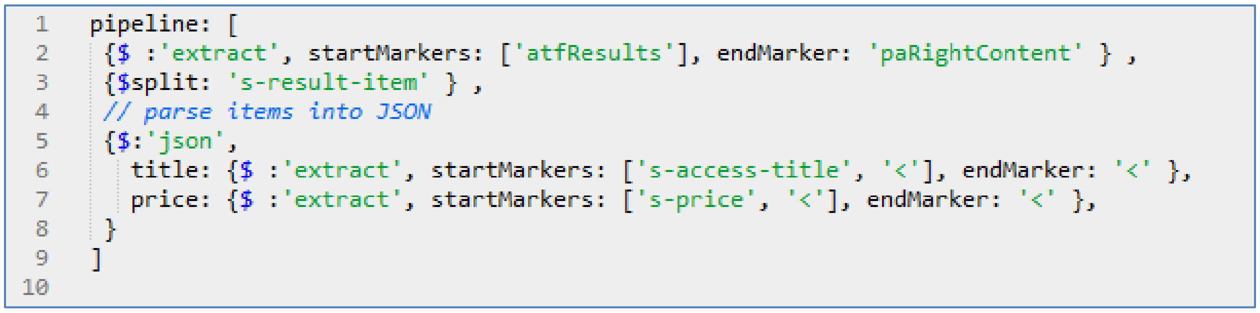
The page that shows the best seller (Jan 2015) size is 460KB with 20 results in the first page. When using the debugger to inspect the html element that contains the title of a book, we see it is around 10 levels deep below the ‘body’ element. Understanding the html structure is not an easy task.

Using the html debugger we can identify some markers without understanding the structure, and come up with the following algorithm.

1. Focus on the result section between `atfResults` & `paRightContent` *markers.*

2. Split between the items using the *'s-result-item’ marker*

3. Extract the title & price from each item using more markers



```
pipeline: [
 {$ :'extract', startMarkers: ['atfResults'], endMarker: 'paRightContent' } ,
 {$split: 's-result-item' } ,
 // parse items into JSON
 {$:'json',
   title: {$ :'extract', startMarkers: ['s-access-title', '<'], endMarker: '<' },
   price: {$ :'extract', startMarkers: ['s-price', '<'], endMarker: '<' },
 }
]
```

The code below is written in javascript in a declarative form which is activated by a zooming algorithm engine. Anyway, you can write your own code with your own language of choice

Discussion About Rigidity

One of our goals was to make the parser rigid enough to survive site changes as much as we can.

First, we should first ask ourselves what we anticipate to be stable in our target site. In many cases we can not find a good enough solution for that. Nothing is guaranteed to be stable.

In our amazon books example, we would like to look for a repeating section that contains title and price. A very smart algorithm can do that, yet even that can be deceiving. We may come out with a list of non-best sellers that was promoted by amazon in a different section at the left.

The marker approach allows a human programmer to choose the best markers. In the example it was majorly css classes that reflect the semantics of the content. Css classes with semantic names rather than style may be regarded as more stable. E.g. : s-access-title and not s-bold. However, we would better understand what does the 's' prefix means, and what is behind 'access'.

Nothing is guaranteed to be stable. Yet, there are more stable elements than others, and humans can make intelligent guesses.

Another aspect to be considered is the simplicity of the algorithm. If it fails, can a programmer fix and update it in a few hours. Automatic tests are very relevant in this context, and this is another pattern.

Known Uses

Itemfield Inc. used this technique to parse data in their ContentMaster product (2006).

The product allowed visualization of the structure of the zooming process with smart previews.

The product was successful and the company was acquired by Informatica (2006), and now (2014) this product and technique is part of Power Center.

References

[1] Automatic Web Data Extraction based on Genetic Algorithms and Regular Expressions. By Barrero, Camacho & R-Moreno. http://arantxa.ii.uam.es/~dcamacho/papers/DMMI-BarreroCamachoMoreno.pdf

[2] HTML Extraction Algorithm Based on Property and Data Cell. 2013 IOP Conf. Ser.: Mater. Sci. Eng. 46 012035