Introducing Ruby to Web Profile Mining

***Abstract*— The Web in a monstrous source of information. Completely, utterly stuffed full of interesting facts and figures, pretty much about anything under the sun. Most of this data, is not traditionally stored in the record set form of data. This paper aims to propose such an open source tool that would combine all the existing api's under a single api built in Ruby for communicating the interesting insights that can be refined from it. This API will most promisingly recognize most of the Web's information as data and make in availble for examonation, inspection and exploration. The Web contains a dynamic and rich collection of hyperlink information. It allows users to access these web pages, domain specifications, Web service, use the information contained within the web pages and provides numerous tool for data mining and web mining. Ruby provides test driven development which is a great tool, as proficient as the api we are using in it. Ruby programming language was designed to remove barriers between programmers and their ideas and thier goals pretty much in the same way that the api is trying to integrate diffrent api's under one system.Accessing the rich Web content with Ruby makes it simple and easy**

***Index Terms*—Web Mining, Domain Specifications,Test driven development, Ruby Api Integration .**

I. INTRODUCTION

The word Web Mining is a method used to scrap through various web resources to gather required data, which allowes an individual or a company to promote business, understanding marketing dynamics, new promotions floating on the Internet, etc. There is a growing trend among companies, organizations and individuals alike to gather information through web data mining to utilize that information in their best interest.The Web is changing fa st over time and so is the users interaction in the Web suggesting the need to study and develop models for the evolving Web Content, Web Structure and Web Usage. World wide web is a enormous amount of widely dispersed, interconnected, beneficial and dynamic hypertext information. It has used in diffe rent needs of us in various stages like communication, business, entertainment and so on. Web data mining is not only focused to gain business information but is also used by various organizational departments to make the right predictions and decisions for things like business development, work flow, production processes and more by going through the business models derived from the data mining.

Web data mining technology is opening avenues on not just gathering information but it is also raising a lot of concerns related to data security. There is loads of personal information available on the internet and web data mining had helped to keep the idea of the need to secure that information at the forefront.

II. WEB MINING

Web mining proceeds to discover useful information or knowledge from the Web hyperlink structure, page content, and usage data. Although Web mining uses many data mining techniques, it is not purely an application of traditional data mining due to the heterogen eity and semi structured or unstructured nature of the Web data. Many new mining tasks and algorithms were invented in the past decade. Based on the primary kinds of data used in the mining process, Web mining tasks can be categorized into three types: Web structure mining, Web content mining and Web usage mining.Two different approaches were taken in the web mining.

i)process-centric view" -defined web mining as a sequence of tasks.

ii)"data-centric view" -types of web data that was being used in the mining process.

1. Data finding: It is used to extract the data from online text resources available on web.
2. Information/Data selection and pre-processing: This process transforms the original retrieved data into information.
3. Generalization/Conceptualization:Individual web sites as well as across multiple sites.
4. Analysis/ Indepth: It involves the validation and interpretation of the mined patterns.

III. MINING USING A RUBY GEM

Going through a collection of videos I started keeping a track of them, copying and pasting their names, rank, and artist into a csv so that I could use columner to organize them and answer questions like: which artists had the most videos in the list; how many videos did the billboards have in it;

IV. DOM TRAVERSAL

A quick look at the html rendered lets us get an overview of the DOM structure used and accordingly formulate our xpath traversal strategy.

Although native DOM traversal like getElementByID might be faster, xpath allows for addition of more simpler, and maintainable code, which is the heart of the Ruby

ISSN: 2278 – 1323

philosophy.

In the scraper code, we include 'rubygems', 'nokogiri', which is a ruby based XML/HTML parser and toolkit, and 'open-uri', which is used for http requests to the Billboard site.

Our pull_video function fetches the contents of a pre defined page, and searches through the DOM via the xpath query for the artist data, and the video data. This data is appended to our CSV file, after which, we make a call to the next page.

Our assumption is that the pages are all coded consistently, hence it is very important that web developers have used Semantic code, otherwise, it makes the parser work harder and thus lowers the general parsing ability of our code.

V. CODE

A. Figures and Tables

Because the fi

```
RSvideos.rb
    require 'rubygems'

require 'nokogiri'   # Nokogiri in
an add-on, installed with: gem
install nokogiri

    require 'open-uri'

    $root      =
    'http://www.billboards.com'

    csvHeader  = 'Record
    #,Reference,Rank,Artist,video,L
    ink'

    $recordNum = 0

    HEADERS_HASH = {"User-Agent" =>
    "Ruby/1.9.3"}

topOfList =
'http://www.billboards.com/music/li
sts'\

                '/the-500-greatest-
    videos-of-all-time-20110407'\

                '/smokey-robinson-
    and-the-miracles-shop-around-
    19691231'

    def pull_video videoPage

       doc          =
Nokogiri::HTML(open(videoPage,
HEADERS_HASH))

       reference    =
doc.xpath("//div[@class=\"listI
temDescriptonDiv\"]/h3").text

       artist, video = split_title
reference

       place        =
doc.xpath("//span[@class=\"List
ItemNumber\"]").text

       csvRecord    =
"\"#{$recordNum += 1}\",\"\


"\"#{reference}\",\"\

                    "#{place},\"\


"\"#{artist}\",\"\


"\"#{video}\",\"\


"#{videoPage}"

       $f.puts csvRecord unless
$f.nil?

       get_next doc

end

def split_title reference

    parts  = reference.split(/,
[\']?/,2)

    artist = parts[0]

    if parts[1] =~ /.*'$/

       video = parts[1].chop

    else

       video = parts[1]

    end    return artist, video

end

def get_next doc
```


ISSN: 2278 – 1323

```
   nextNode  =
doc.xpath("//a[@class=\"listPag
inationControls next\"]")

   nextHref  =
nextNode.xpath("@href")

   nextItem  = $root +
nextHref.text

   if nextItem == $root ||
$recordNum > 501

       nextItem = nil

   end   return nextItem

end

puts "Starting with:
#{topOfList}"



$f =
File.open('RSvideos.csv','w')

$f.puts csvHeader unless
$f.nil?

nextItem = pull_video topOfList

while nextItem     nextItem =
pull_video nextItem

end
```

`$f`.close unless $f.nil?

VI. THE RUBY COLUMNER PARTNERSHIP

This is not really a partnership. Neither of the players knows of the other, so it's more like a happy meeting of functionality, but that makes a dull section heading.

VII. `RSvideos.csv`

The single most common element in this data mining system is the CSV file "`RSvideos.csv`". The Ruby script creates it. columner accesses it and makes its data accesible for analysis. There's nothing extra-ordinary about the name "`RSvideos.csv`" either, any other would do but I had a bigger idea and sensible naming is always a good format.

VIII. March ahead – give it a try.

This Ruby code is free for everyone toy use . You can use it in any website with follows consistent code. Play it and get your own data set of the Billboards 500 GREATEST VIDEOS OF ALL TIME.

IX. The usual debuggers.

`RSvideos.rb` works but it's definitely not prone to minor glithches. Running from my home it works along properly running it from my office the content before FOB 1402 it grabs a bunch of the videos and then fails, with an error I think has to do with proxies or the specification of a Proxy Agent.

This should work, but make no promisises.

VII. CONCLUSION

Mining these videos was a great interesting exercise, one that helped me from the shackles of a too-rigid conception of what columner-analyzable data is. I got this the project working I found myself interested in more and richer content to explore. Thus writing simple ruby code, can make information of big data.

ACKNOWLEDGMEN

Yukihiro “Matz” Matsumoto, https://www.ruby-lang.org/en/ For the consstant inspiration to be happy programmer.

REFERENCES

[1] Aaron Patterson sparklemotion/nokogiri · GitHub
[2] Yukihiro “Matz” Matsumoto, https://www.ruby-lang.org/en/

First Author Neil Fernandes MCA Insititiute of management and computer studies, Thane. Web- developer at Web-Media solutions, Siolim Bardez, Goa

ISSN: 2278 – 1323