Project Report

on
Web Content Scraper

Submitted by
Balagopal K S
120260028

Under the Guidance of
Dr. Pushpak Bhattacharyya

Phd Guide
Raksha Sharma

Department Of Computer Science and Engineering
Indian Institute of Technology, Bombay

Abstract

Having a big corpus is the basic problem of the machine learning algorithms. In this project we address this problem through a tool “Web Content Scraper”. “Web Content Scraper” is a program that mines the review web sites recursively just by specifying root web page. This report presents the detail of the project “Web Content Scraper”. We currently focus on generating a tool to extract corpus for the movie domain from www.rottentomatoes.com. But this approach is not limited to the movie domain or www.rottentomatoes.com . It can be applied to any domain and to any review web site, that has a big collection of reviews.

Acknowledgment

I am pleased to acknowledge Prof. Pushpak Bhattacharyya for his invaluable guidance during the course of this project.

I extend my sincere thanks to Ms. Raksha Sharma who continuously helped me throughout the project and without her guidance, this project would have been an uphill task.

Balagopal K S
120260028

Contents

1. Introduction

Having annotated corpus with its class is the basic need of learning algorithms. An automated approach was nevertheless essential for generating a large annotated data set. We present an approach to extract reviews written for movies with their assigned polarity from a well known movie review website that is, www.rottentomatoes.com. We choose this website because it has umpteen number of links to other review web pages. These links refers to either newspapers or critic's own web page. The considered web site has an efficient way to assign polarity to reviews.

The recursive structure of this site was analyzed and the scraper was made in such way that it gets the links from the rotten tomatoes site and mine each site given for reviews. The programming language Python 2.7 was used for the development. The flexibility and simplicity of the language made it best suited for this project.

The road map for rest of the report is as follows: section 2 describes the polarity assignment procedure by rotten tomatoes. In the same section we discuss the overall polarity assignment for a movie. Motivation behind this work is discussed in section 3. Methodology is described in section 4. Section 5 gives a flow chart for overall all working of scrapper. Section 6 illustrates experimental setup with discussion over issues related to experiments. In the end we discuss the issues came across while scrapping corpus in section 7. Final result of the project is put down in section 8. In section 9 we discuss future enhancement possibilities.

2. Polarity Assignment

Polarities are first assigned to each review independently. Then overall polarity of a movie is determined by considering all reviews , received for that movie.

2.1 Review polarity assignment

Rotten Tomatoes staff first collect on-line reviews from authors that are certified members of various writing guilds or film critic associations. To become a critic at the site, a critic's original reviews must garner a specific amount of "likes". Top Critics are generally ones that write for a notable newspaper. The staff then determine for each review whether it is positive ("fresh", marked by a small icon of a red tomato) or negative ("rotten", marked by a small icon of a green splattered tomato).

2.2 Movie polarity assignment

The website keeps track of all of the reviews counted, recently released films *The Dark Knight Rises* (2012) reported 304 reviews. If the positive reviews make up 60% or more, the film is considered "fresh". If the positive reviews are less than 60%, then the film is considered "rotten". In addition, major film reviewers like those at Roger Ebert.com, Desson Thomson, Stephen Hunter, Owen Gleiberman, Lisa Schwarzbaum, Peter Travers and Michael Phillips are listed in a sub-listing called "Top Critics", which tabulates their reviews separately, while still including their opinions in the general rating.

3. Application of the work

There can be many application of the output received from the work. Since the output of the project is a big annotated corpus, so it can be used in any application which requires annotated corpus. Here, we are mentioning two such important applications.

3.1 Sentiment analysis

Sentiment analysis refers to classifying documents according to the sentiment they convey. Sentiment reflects the emotional content in the document or the attitude of the speaker to the subject matter. Sentiment classifiers have many applications in society. Designing of an automated system for sentiment

analysis needs corpus. Annotation of corpus with polarity facilitates implementation of machine learning techniques for sentiment analysis. (Corpus is a large collection of texts in naturally occurring language, It is used for linguistic analysis but it may also be used as a tool in second language teaching and learning).

Our approach is able to fetch a big annotated corpus, that can be used directly for the sentiment analysis purpose.

3.2 Information Retrieval

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on meta data or on full-text. *Web search engines* are the most visible IR applications.

Corpus obtained by our approach can be used as a resource for information retrieval.

4. Methodology

The main aim of the project is to create a big corpus that can be used for various applications. Manual extraction of corpus from HTML pages is a tedious job. Besides this it requires additional efforts for cleaning of corpus after extraction. To address this problem , we generate a tool named as “Web content scraper”, that extracts corpus automatically. Web content scrapper is a program that produce output in two phases.

- A crawler that navigates through all the movies in the site www.rottentomatoes.com and scrap the reviews listed under each movie.
- A corpus of movie reviews with a sentiment assigned to each.

4.1 Scraper Design

The program was developed using Python 2.7 and the python package "Beautifulsoup 4.0". Beautiful Soup is a python library for pulling data out of HTML and XML files. It is designed for quick turnaround projects like screen scraping. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching and modifying a parse tree.

The program was developed in different modules. The whole development procedure can be mainly divided into two parts.

1. Making a Scraping Script for a single review site:

First part was to make a script that will extract the review alone from a single site that is provided. Since there can be many different sites to scrapped, a common script was made for this purpose. All the visible text in the review site was converted into strings. Each paragraph, each independent text in the site was converted to separate strings. From these strings the HTML codes were removed. Now, in-order to get the review alone, the strings were filtered according to their size. Only large sized strings, which will be the review parts were kept. Now these strings together, in order forms the review and it was saved to a text file. Before saving each string was also checked for the presence of HTML encoding and if found those were replaced.

2. Automating the extraction for all reviews:

The second step was to make the extraction automated for all the listed reviews of each movie in the rottentomatoes database. For this purpose the structure of rottentomatoes site was analyzed. Each alphabet was searched for in the site and from the results movies were selected. Though the result page contains many links, links to movie page alone was to be selected. For this divs named "media_content_block" were first found. These blocks contains the link to movie page and main cast also. In order to remove cast link and select movie page link alone, hyperlink tags with attribute target having value "_top" alone was found. This at last provided the movie page link alone.

From the movie page program navigates to the review lists section. From there link each review site was found by looking for hyper link tags with inner value “Full Review”. The sentiment of the review already mentioned with the review, was found by checking whether the div has class fresh or rotten, where fresh relate to positive and rotten to negative. Now from each review site review was mined using the scrap script.

5. Overall working of project

This flow chart describes the overall working of the “Web Content Scrapper”

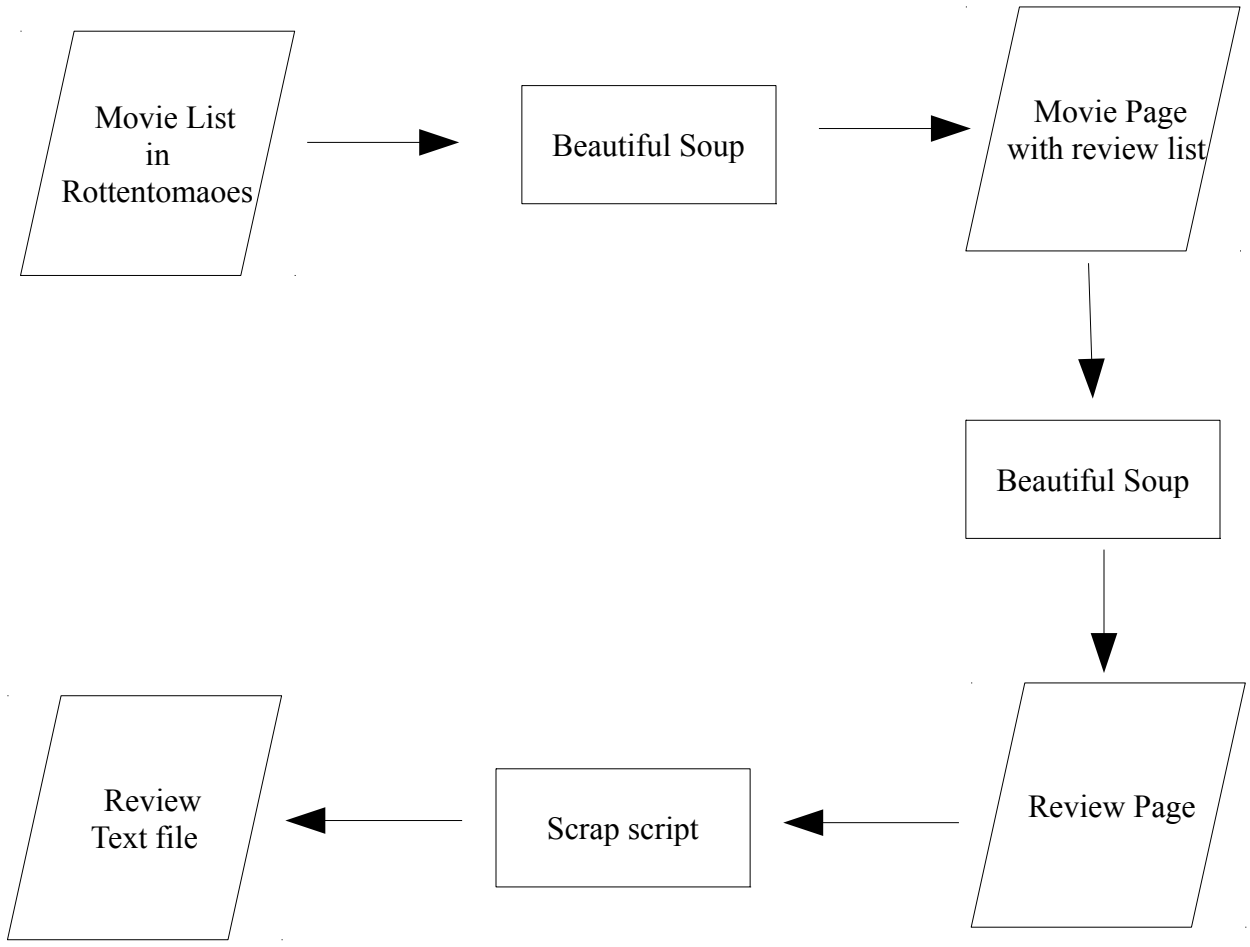

Figure 1: Working of Scrapper

5.1 Working of Scrap Script

This flow chart describes the working of Scrap Script.

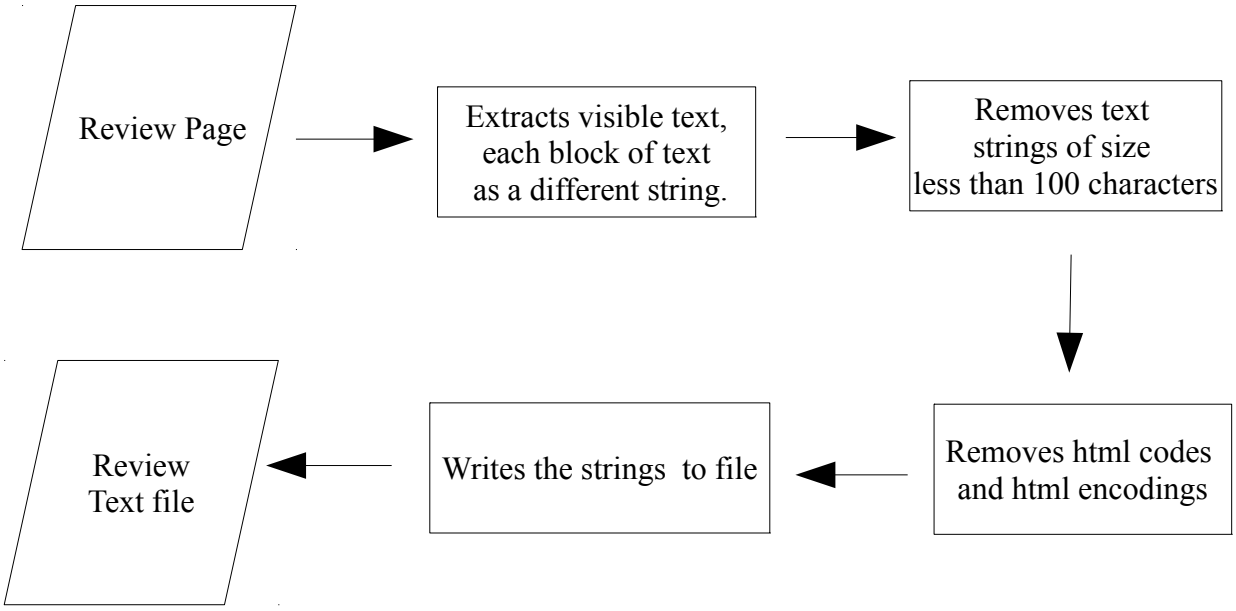

Figure 2: Working of Scrap Script

6. Experimental setup and Discussion

The scrap script was primarily tested on many review sites. From these tests the string size criteria was selected to be 100 characters. The whole program was run on different movie pages in rottentomatoes and later on different alphabet searches. Some of the difficulties that arose during this step are as follows:

- Some sites block programs and rottentomatoes was one of such sites. We over came this problem by browser spoofing. A browser header was given to scraper which made it look like a browser to any server. This easily bypassed the blocking.

- The alphabet search results of rottentomatoes were using Ajax. Ajax database makes it possible to have different pages with single address. Hence it was impossible to navigate through result pages based on address. We used the link to Ajax database for rottentomatoes to overcome this problem. The database link for search 'b' and page 1 is www.rottentomatoes.com/search/ajax/?searchtype=movie&search=b&page=1

7. Corpus Analysis

Corpus contains basically two folders, one for positive reviews and other for negative ones. The reviews are saved under folders named with the respective movie name. The scrap script was common for all sites and hence there is chance for noise in some reviews. Basically the criteria was based on string size which gives possibility of the reviews having some comments also along with them.

Some websites use HTML encoding in the reviews, since we are scraping the review from the HTML code, these encoding can be found in the review. An attempt has been made to overcome this by replacing the encoding. But since there are lot of encoding some of them could have been missed. Any new encoding can be replaced by adding them to the encoding list in the program.

In case of bad sites, they are skipped by the program. Still some sites could be accessible but have no review. This results in blank text files.

8. Result

A Corpus containing 9,024 positive reviews and 5,483 negative reviews was created. A total of 14,507 movie reviews were extracted containing reviews of about 175 movies. This count can be increased by running the scrapper for a longer time. There is an option to resume the tool, on starting again it will continue from the last movie whose review you have been extracting.

9. Conclusion and Future work

Our approach is able to extract 14,507, annotated with the polarity. Our approach can be applied to any website, that can help in the process of collecting corpus. Some of the possible improvements and uses that can be done in future are as follows:

- In order to remove noise, a separate program for scanning the text files and removing unnecessary text can be made.
- Same program method can be used for other sites. Analyzing the new site structure and making necessary changes.
- The noise problem can be solved if we work on a site that has all reviews in itself rather than linked in other websites. This could help us to make a scrap script specifically for the site itself, reducing the possibility of noise.

Bibliography

http://www.python.org/

http://www.python.org/doc/

http://www.crummy.com/software/BeautifulSoup/

http://www.crummy.com/software/BeautifulSoup/bs4/doc/

http://en.wikipedia.org

http://stackoverflow.com