PRODUCT REVIEW ANALYSIS

Contents

Introduction

This project comes under the category of big data and analytics. Here we are storing and analyzing the product details from e-commerce websites like Amazon.in and Flipkart.com. The main objective of the project is to find the untrusted usernames/userId/Reviews and get the actual product reviews by eliminating fake information.

There is no specific way to filter these out, so various methods form Machine Learning and Natural Language Processing will be used along with Data Mining.

The key focus is to find an approach where fake information can be found out from the data and can create a better recommendation engine based on user's perspective not on user's rating and no. of views of a product.

The project is built on the basis of data mining approach. Here text is converted to numbers and then comparison is being made. More sophisticated way would be using NLP. Where the actual meaning of the review is compared and based on the meaning the fake information can be captured.

Description

1. Technology used

    - Java v8.0
    - MongoDB v3.0
    - Hadoop v2.6
    - Cloudera v5.5

2. Perquisite knowledge

    - DOM parsing
    - Machine learning
    - Basics of data mining and warehousing (Clustering, Classification, Data Transformation…)
    - NoSQL database
    - Hadoop

Process Flow

The following steps describe the working of project in depth:

Scrapper

1. Exec.ExexMain.java is executed first for getting the content based on some keyword. The keyword needs to be passes in the variable called “keyword”.

2. Two threads one for Amazon.in and another for flipkart.com will be executed with input as the above mentioned keyword.

3. The execution will now goes into *FlipkartSearchLinkScrapper.java* and *AmazonSearchLinkScrapper.java.* These file will fire the keyword into the search feature of the e-commerce website. All the links of the product found in the search result will be stored in a list.

4. The items in the list(url of products) created in step 3 is now fed into *AmazonReviewScrapper.java* and *FlipkartReviewScrapper.java.* These files will fetch all product details and iterate over all the comments. These information is stored into MongoDB. (Refer MongoDB_document_structure.pdf)

Analysis

1. Analytics package contains the file related to analytics being done on the stored data. TFIDF of reviews, username filter etc.

2. In *UsernameFilter.java* the system first selects the username whose distance is less than 4 which is being done by levenhstein algorithm. All comments of the flagged usernames are fetched and compared for some similarity pattern via TFIDF and cosine similarity.

   Hence at this point we got the comments of user and username whose review pattern are similar. But there is no guarantee that the user is fake, because two usernames can be identical and their reviews can also be similar.

   Drawback of this approach is lack of guarantee and time consumption approach.

3. To overcome the time consumption problem we introduced hadoop in project. There are map-reduce jobs which calculates the tfidf in seconds and stored in *tfidf_job3* collection

   1st job gets the word count of each word of a review

   2nd job calculates the term frequency

   3rd job calculate the TFIDF

All the job results are stored inside mongodb based on their job numbers. Eg tfidf_job1,tfidf_job2.

4. Now the similarity is comments is need to compare with the date distance(days gap between two similar comment pattern) so that a particular pattern can be found. This is to be done in Spark

Input

Scrapper – search keyword in file ExecMain.java

Analysis – NA

Hadoop MR jobs – mongodb url in all the drivers

Output

Scrapper – product details will be stored into MongoDB

Analysis – a list of usernames

Hadoop MR jobs – result will be stored in MongoDB

Commands

Refer MongoQueries.txt

Notes

1. All the configuration is set in config.property file
2. During this project, MongoDB was installed on windows machine with no authentication and Hadoop was running in Cloudera 5.5 virtual machine.
3. All the dependencies can be found in lib folder or jars.zip file, which should be imported to BuildPath
4. MongoDB authentication is not configured.
5. There are two project one is in the *ProductReviewAnalysis* directory and another inside *ProductReviewAnalysis**Hadoop*
6. Before begin please have a look at all the documents.
7. Change the mongodb url in all the driver classes of Map reduce jobs