Web Search Engines — Reviews-Rehashed

Himaja Rachakonda, Anirudhan J. Rajagopalan
N14633788, N18824115
hr970, ajr619

March 21, 2016

1 Objective

The aim of "Reviews-Rehashed" is to build a specialized search engine to search product reviews pertaining to a product and its features. The search queries are in theform of <product,feature> tuples. This search engine will crawl reviews from various e-commerce websites such as Amazon and Best-Buy.com to fetch reviews data of various products. It searches for the user query and presents the links to the results after ranking the documents based on the intrinsic quality of the document as well asthe retrieval score. The quality of the document is based on a static score derived from the features of a review like the ratings, review-date, comments etc. We have implemented the search engine for electronics - mobiles data from Amazon.com. There is a huge scope of expanding this to other products and other domains as well.

2 Data Sources

Amazon.com and BestBuy.com were the options considered to collect information regarding products and their reviews. To build a crawler to fetch the data, a preliminarystudy was conducted on the following to understand the website page layout and structure —

1. User behaviour to search for a particular product and review.
2. Patterns in the URLs of the websites to reach to a particular product / product category / review / etc.
3. Examining the sitemap.xml of Amazon.com and BestBuy.com to get all the URLs of the domain.

Sitemap is a list of pages of a web site accessible to crawlers or users. It can be either a document in any form used as a planning tool for Web design, or a Web pagethat lists the pages on a Web site, typically organized in hierarchical fashion. Sitemaps make relationships between pages and other content components. It shows shapeof information space in overview. Sitemaps can demonstrate organization, navigation, and labeling system.

Sitemaps are a useful tool for making sites built in Flash and other non-html languages searchable. If a website's navigation is built with Flash, an automated search program would probably only find the initial homepage; subsequent pages are unlikely to be found without an XML sitemap.

This helped us to make sure all the review pages are searchable by our search-engine and also to find useful patterns which help in identifying each product and reviewwithin each of the domains of Amazon and BestBuy. Because of the vast expense of the data and the crawling required to be done, we have restricted our project to Amazon Review Data pertaining to Mobile Phones under Electronics Sections. This can be scaled easily to other categories and domains seamlessly with appropriate crawler jobs.

All the review pages of a particular product with an ASIN Id can be found at a URL in the following format -

www.amazon.com/<Anything>/product-reviews/<ASIN>/

Every Review of a product has a unique Review ID, and therefore each review can be accessed by a URL in the following format -

http://www.amazon.com/gp/customer-reviews/<Review-ID>/

3 Data Collection

Data Collection was executed in two phases -

Amazon Product Crawler: Fetches all the unique Ids for all mobile phone products from Amazon.com. This is required to construct the seedUrls for each product as mentioned above in the Data Sources Section.

Reviews Crawler: This crawler take the output of the the Amazon Product Crawler as the input to construct a seed Url for each product Id. All the unique IDs (ASINs) are stored, which are used to fetch Reviews from each product. These reviews can be accessed from this page through the links in the pagination bar. Eachpage consists of 10 reviews and the crawler goes to the depth until which there are no more review pages to be crawled for that product. Therefore, this product level URL is given as a seedURL for every product to crawl across various review pages through the links in the pagination bar.

The data has been collected for around 500 produts with a total of 150000 reviews. The following data about each review is collected as indexable fields based on whichthe review will be scored during the search time –

1. Review Content

2. Review Title

3. Review Date

4. Number of Comments to the review

5. Number of people found this review helpful

6. Number of images posted by the user in this review

7. Ratings of the review

8. Is the product a Verified Purchase

9. Review Length

10. Review Id

11. ASIN (Product Id)

4 Search Engine Architecture

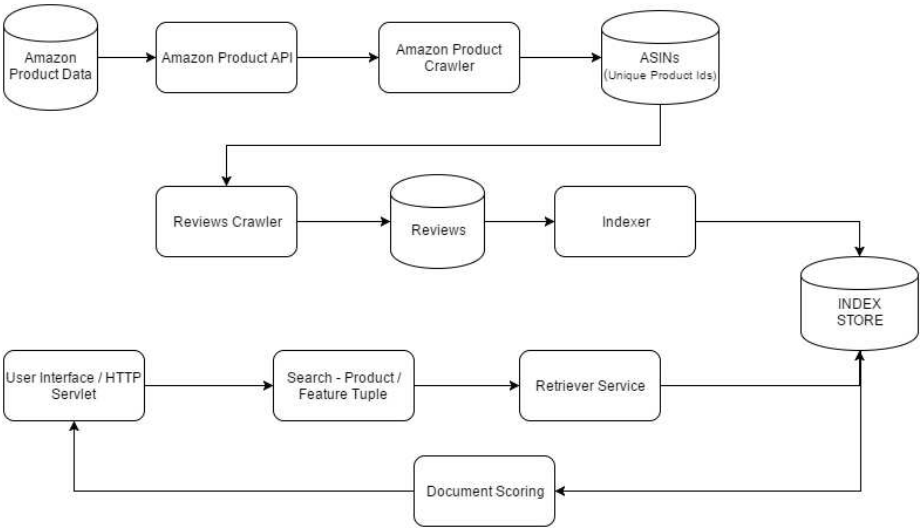

Figure 1: Reviews-Rehashed Search Engine Architecture

The search engine will be built by using the following different components.

Preprocessing module: This module takes care of preprocessing the data and dumps the processed data into Lucene.

Amazon WebCrawler : This module fetches all the ASINs (unique Product Ids) for Mobile phones under Electronics Sections

Reviews WebCrawler : This module fetches all the reviews of the products from the product Ids retrieved from the Amazon WebCrawler module.

Document Scoring : This module implements the ranking algorithm by which the documents are retrieved in the search algorithm.

Retriever Service : This module receives the search query from the Web interface and interacts with the index store to retrieve results. The retrieved results, based on ranking algorithm in Doc Scoring module, are returned back to the web interface.

Web interface: A web interface for the user to issue the search queries. This will be an interface with two text boxes. One for the product name and another for the feature name.

Index store: This is a Lucene index store, an inverted index, which maintains all indexable data required while retrieving search results. The data will be

populated and updated periodically by a crawler job that looks for updates to the data. The data will be stored in lucene after preprocessing.

5 Document scoring

The documents are scored based on a personalized page rank algorithm which takes the intrinsic quality of the document into consideration as well as the lucene score generated. In many search engines, we have available a measure of quality g(d) for each document d that is query-independent and thus static. This quality measure may be viewed as a number between zero and one. The net score for a document d is some combination of g(d) together with the query-dependent score induced by lucene. This kind of ranking algorithm demands an accumulation of an evidence of a document's relevance from multiple sources.

The personalized pageRank algorithm gives 25% weightage to the quality of the document and 75% weightage to the lucene score of the document. The quality of the document is again a combination of weights given to each of the dependent facors.

The pageRank algorithm is as follows -

$$pageRank(d) = 0.25 * g(d) + 0.75 * (lucenescore) \tag{1}$$

The quality $g(d)$ of the documents is as follows

$$\begin{aligned} g(d) &= 0.6 * H + 0.20 * V + 0.13 * C + 0.07 * I, \\ \text{where } H :&= \text{ Number of people who found the review helpful,} \\ V :&= \text{ the purchase is a verified purchase,} \\ C :&= \text{ Number of comments to the review,} \\ I :&= \text{ Number of images in the review} \end{aligned}$$

As all the dependent factors and the lucene score are unbounded non-negative variables with different frequency distributions, it is necessary to normalize them before we use them in the pageRank algortithm in equation (1) to have consistent scores. A sigmoid mathematical function is chosen to normalize these unbounded variables with a multiplacative factor of the x-axis to obtain scaled values between 0 and 1, based on distributions of each of these variables.

A sigmoid function is a bounded differentiable real function that is defined for all real input values and has a positive derivative at each point. ts range is always between 0 and 1.

$$S(t) = \frac{1}{1 + e^{-t}}. \tag{2}$$

The following graph shows the spread of the sigmoid function for various multpicative factors –

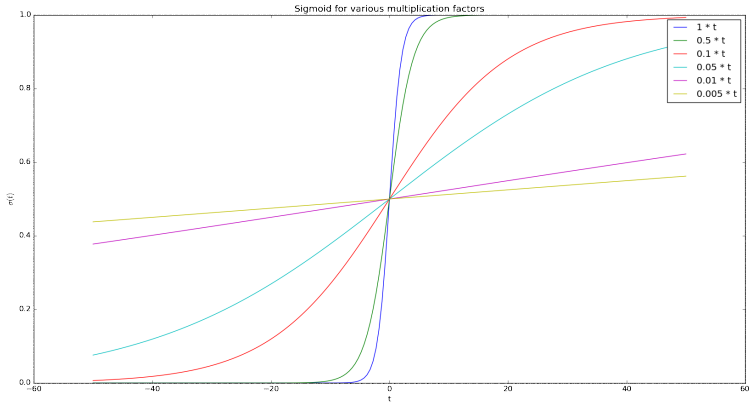

Figure 2: sigmoid variations with different multiplicative factors

6 Observations

positive cases where it works perfectly well Straight forward cases work well.

cases where it does not work well conjunctive queries within features will not show reviews with both the features discussed. Queries with null product name do not show accurate results with current scoring mechanism, return accurate results with just lucene score. adjective + feature does not always return accurate results

7 Experiments

different multiplicative factors

sentences indexing

8 Challenges

Understanding patterns from sitemap.xml was a time - taking task.

Manually checking the weights was a tedious task, evaluation was difficult. Have to train the data to tune these weights.(future work)

9 Future work

NLP module: This module will be invoked during the query time. This is essentially a machine learning model that was pre trained using the data

collected by the crawler. We use this to rank and get the list of all matching sentences for a given query.

Score parameters tuning

opinion mining

10 To DO

references

proof reading

hyper references