|  |  |
| --- | --- |
| Carnegie Mellon | Capstone Project |
| Tag Cloud Compare - Concept Document | |

# Abstract

Reviews have become one of the most important sources of information guiding the behavior of potential buyers of a product. They present many more detail than ratings and describe important advantages and disadvantages of a specific product. One issue with reviews is that they are usually long and take time to read; especially in cases were hundreds of reviews are attached to a single product.

Since best-selling products usually get hundreds of reviews, it's interesting to discern if they provide enough information to allow for automatic retrieval and summarization. The goal would be to extract adjectives or small descriptors from these reviews and present them to users in the form of tag clouds, so they can easily compare their frequency and impact without the need to read all reviews. This format would allow an easier understanding of characteristics, pros and cons of a specific products using limited space, making it especially suitable for comparisons between products.

Another issue with comparisons is the lack of some details or specifications on the product description, details that may be important to buyers. Users in these cases rely on other user comments to find out the specific features of products, which cannot be found in normal filters in the merchant's website. Our plan is to allow users to fill these holes with additional information, so the total crowd-sourced effort provides accurate, detailed and meaningful comparisons.

# Vision / Objectives

* Allow quick and deep representation of product characteristics as short tags, replacing long reviews.
* Provide an intuitive interface to manipulate, persist and/or share product comparisons.
* Use crowdsourcing to collect and compare product details not present on official product descriptions.
* Support crowdsourcing efforts by automatically synthesizing of long reviews from Amazon's website.

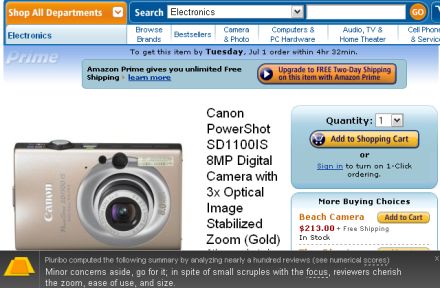
# State of the Art

## Customer Review Summarization Tool

### Pluribo (<http://www.crunchbase.com/company/pluribo>)

Pluribo is a Firefox extension tool that scans all the reviews on any Amazon webpage and summarizes them into a single paragraph. It prepares summary of what people liked and what people didn't like about the product. Pluribo displays the summary at the bottom of the browser. Developers have stated that the tool works well on electronic items and they would update it for other categories of items.

Pluribo output screen shot (<http://tinyurl.com/9q9kzz5>)



## Tag Cloud Generators

### Non-Interative Cloud Generation

#### Wp-Cumulus

* <http://wordpress.org/extend/plugins/wp-cumulus/>

A wordpress plugin that present a dynamic rotating cloud of wordpress pages.

### Interactive cloud generation

#### Tagul, Wordle, Tagxedo

* [http://www.tagcloud-generator.com/](http://www.tagcloud-generator.com/demo.php)
* <http://www.sis.pitt.edu/~dparra/p113-trattner.pdf>

## Product Comparison Websites/Plugins

Several sites were researched. For each one, the site Alexa (<http://www.alexa.com>) provided a description and insights into the general popularity, traffic, query statistics, etc.

### Epinions (<http://www.epinions.com/>)

Epinions adds a shopping layer over amazon, ebay, dell, newegg etc websites. It allows product search. It ranks the search results fetched from amazon, ebay, dell, newegg etc retailers and creates its own ranked results. It allows product comparison. It allows user logins, comments, product suggestion, reviews and many other features of a social network.

From Alexa: "The site has a bounce rate of approximately 67% (i.e., 67% of visits consist of only one pageview), and the site is particularly highly ranked in the city of Washington (DC) (#787). Roughly 24% of visits to it are referred by search engines. It's visited frequently by females who are in the age range 55-64 and received some college education."

|  |  |  |
| --- | --- | --- |
| 1 | Epinions | 0.42% |
| 2 | vacuum cleaner reviews | 0.27% |
| 3 | toaster oven reviews | 0.20% |
| 4 | automatic litter box reviews | 0.19% |
| 5 | eyebrow cutter | 0.19% |
| 6 | ebates | 0.12% |
| 7 | sleep number bed reviews | 0.12% |
| 8 | user reviews | 0.12% |
| 9 | tempurpedic reviews | 0.12% |
| 10 | washing machine reviews | 0.10% |

Top queries that direct to Epinions

### Q-Compare (<http://www.q-compare.com/#home>)

Q-Compare allows users to search item from amazon and then it automatically searches the similar items from ebay. User can click on one of the several ebay items. The website provides very limited product comparison. It is a good concept with lots of scope for improvement.

From Alexa: "Q-Compare is visited most frequently by users who are in the age-range 25-34, have no children and are graduate school educated."

|  |  |  |
| --- | --- | --- |
| 1 | ebay vs amazon price comparison | 17.80% |
| 2 | website compare amazon and ebay prices | 17.33% |
| 3 | price compare ebay amazon | 17.21% |
| 4 | q compare | 17.04% |
| 5 | software that compares ebay prices to amazon prices | 8.54% |
| 6 | compare.com | 4.95% |
| 7 | ebay and amazon comparison | 2.57% |
| 8 | amazon price comparison | 2.17% |
| 9 | compare prices amazon | 1.46% |
| 10 | qcompare | 1.15% |

Top queries that direct to Q-Compare

### Shopping Assistant (<https://addons.mozilla.org/en-us/firefox/addon/shopping-assistant/>)

Browser plugin. Allows a user to select products to compare. Shows the product details side by side for multiple product features.

### CompareAmazon (http://compareamazon.com)

Firefox plugin. User can compare only the Amazon products. The plugin extracts the following information for the target products and populates it side by side for comparison:

* Title
* List Price
* Your Price
* You Save
* Shipping
* Availability
* Label/Manufracturer
* Model
* Category
* UPC
* Features
* Description
* Category

### Pluscompare ([www.pluscompare.com](http://www.pluscompare.com))

It is an Amazon associate website. It adds a single layer over Amazon products. It allows product search. It has its owns heuristic to rank advertised items. It finally redirects you to Amazon product page once you want to buy something.

User can click a button to add it to the comparison bucket. But the compare functionality doesn’t work at present.

### Priceheat (http://priceheat.com/)

Just compares the price of the products. Plugin not working at present.

# User Requirements

## Product Selection

### Manual Insertion

A user may manually insert an Amazon URL or product id to add it to a comparison.

### Product Search

A user may input search terms and the system uses Amazon's search functionality to provide relevant results.

### Product Suggestion

Based on the products already being compared, a user might receive suggestions about products with similar characteristics.

Note: It might be that Amazon already provides this feature. How to use it or improve it based on tag information already present should be investigated.

### Result filtering by tags (requires analysis)

Tags also provide a way to drill down or filter items, allowing users to select the desired specifications and descriptors. An auto-complete tag search box could allow adding more filters.

Note: An interesting question is if it's possible to allow both AND and OR semantics when adding tags to a specific search.

Note: This is a complex functionality in the case were no tags have been generated for a specific product, filtering out potentially interesting results.

## Product Comparison

### Basic comparison display

A user should see all selected products displayed side by side, either vertically or horizontally, to allow for easy comparison. This basic scheme should follow some basic conditions, allowing the user to:

* Change the order in which products are displayed.
* Permanently remove a product from the comparison (unless the same product is added again).
* Temporarily set aside a selected product from the comparison and putting it back later.
* Select one or more new products to add to the comparison without losing or reselecting those already in it.

### Reviews and opinions

A user should be able to quickly determine the relative advantages and disadvantages of the different products. For this purpose, opinions should be presented in a way that allows the user to quickly scan for the most relevant information. The interface should allow the user to:

* Distinguish between positive, negative and neutral opinions.
* Determine the relative global importance or impact of each opinion, taking into consideration the number of users contributing to it.
* Quickly determine the positive/negative balance of opinion.
* Add an opinion.
* Vote on existing opinions, increasing or decreasing its importance.
* Mark an opinion as positive or negative, to account for potential misclassification or human error. Overwhelming evidence should be required to effect a change.

### Product specifications

A user should be able to quickly review the most important characteristics of the selected products. These characteristics should be facts, not a matter of opinion. A user should be able to:

* Get a quick overview of the most important characteristics of each product.
* Be able to show or hide specifications of characteristics given his interests.
* Compare the presence, absence or difference of a characteristic across the products being compared.
* Complete or complement the existing information for the products on the comparison.
* Vote on the relevance of a certain characteristic. This voting should be according to its perceived relevance. For instance, weight is a more important characteristic on a laptop than on a desktop PC.

### Persistence

A product comparison should be persistent, in a way that allows the user to retrieve it for later use or even share it to other people. All user input, configuration or customizations should be restored, both in terms of the user interface and the product information.

### Purchase

A user should be able to be directed to the product website on Amazon to buy or review any of the products currently in the comparison.

Note: Depending on details of Amazon's contract terms a local shopping cart might need to be added on the site.

## User registration/personalization

While not mandatory in any way, the user should be able to register in the site in order to access additional features.

### Basic features

A user should be able to register, login, logoff, recover his password and any other standard procedure present in most sites.

### Comparison Storage

A user should be able to store a comparison into his account and retrieve it later on.

### Game features/incentives

A user should be able to earn distinctions on the site based on his collaboration. Types of collaboration include voting, reviewing of modification of product specs. Several kinds of achievements can be added to the user profile based on his input. These achievements should be clearly seen every time that the user is logged in.

Other options include leaderboards or weighting heavy (more committed) users differently, for example by counting their votes twice.

# Technical Requirements

The main feature of the website is to provide product comparison of small and diverse tags instead of long texts. The tags can come directly from users but also from automatic algorithms. The idea is to have a mixture of automation and collaboration that is better than each approach separately.

## Obtaining tags

### Automatic tag discovery from reviews

For each product we can obtain both a description and a series of reviews. Part of speech taggers or similar technology could be used to extract meaningful tags for a product.

### Automatic tagging from categories and filters

Amazon categorizes each product into one or more branches of a fixed hierarchy. These categorizations can also provide meaningful tags. Similarly, each product already has a limited set of search criteria, which values for a particular product could be extracted. For example, photo cameras are pre classified according to brand, optical zoom, resolution, etc.

## Categorizing and weighting

### Tag preprocessing

A steamer will be used to reduce each tag, avoiding unnecessary repetition and stop words will be automatically removed. The total count for each tag will be kept, along with an inverted list with information from the origin of each tag. This preprocessing will be identical for both manual and automatic tags.

### Tag Types / Color coding

Tags can be of 4 different kinds:

* Negative (red): those that describe problems or issues of the product.
* Positive (green): those that describe advantages or positive opinions on the product.
* Neutral (gray): those that are informative but ambiguous (neither positive nor negative).
* Descriptive (blue): those that are facts, like characteristics of the product (size, capacity, etc.).

The color coding of each tag will be inserted directly by the user or given by the rating of a specific review. Some intelligence will be needed in this case to avoid classification errors, since bad reviews can also mention pros and vice versa.

### Crowd-sourced synergy

Since users should be able to interact with the product information, the final weight and color of any given tag should be a weighted sum of both automatic and manual contributions. Contributions from users also have potential as training data for tagging algorithms.

# Open Ideas

## Plot changes on tag clouds over time.

Information provided on reviews may vary over time. A product can for instance get excellent ratings initially but may fail miserably after a few months, creating a new inflow of bad reviews. It might be interesting to account for time into the calculations for a product's tag cloud and any changes could maybe be displayed graphically.

## Automatic tag advertising

Descriptive tags can potentially be used for advertising. Once the algorithms and crowd sourcing have been able to discern small tags that describe each product, Google's advertising platform could be used to bet on those tags that are not too expensive. The click-through behavior can maybe be used as a hint on the usefulness of each tag.