# Product Characterization from User Reviews

Leandro Kieliger, Quentin Bacuet
*Department of Computer Science, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland*

***Abstract*—In this project we build a data processing pipeline which is able to extract from user reviews product characteristics such as quality or performance. Our approach uses part-of-speech tagging to retrieve basic impressions on products and extracts qualities by identifying the best bigram collocations based on point-wise mutual information and ratio of likelihood scores. The characteristics are segmented into positivity classes by analyzing user sentiments. The pipeline developed is applicable to any type of product and has direct real-world application possibilities.**

## I. Introduction

When shopping online, we often rely on the feedback provided by other internet users to decide whether the merchandise we are considering buying is worth the money. Most electronic commerce platforms allow their users to rate their purchases with a system of points and to leave written feedback. However, when the number of reviews becomes large, it can become increasingly difficult to fully grasp the impressions of the whole community since only a small fraction of reviews can be shown at any given time.

The aim of this project is to complement those means of providing feedback by automatically analyzing and aggregating reviews of multiple users. One simple and efficient way of achieving this is to look for pairs of words, also called bigrams, whose individual elements belongs to specific parts-of-speech. The most interesting categories in terms of opinion mining are adjective-noun pairs such as ”good quality”, adverb-past participle such as ”reasonably priced” and verb-adverb like ”works well”. Our data processing pipeline first identifies user impressions by analyzing the frequencies of adjective occurrences. More specific characteristics are found by searching for bigram collocations which belong to one of the previously mentioned categories. Each collocation is assigned to a positivity class based on user sentiment. Finally, product characteristics can be aggregated by brand to study how different brands are perceived by the customers.

Section IV describe the implementation behind adjective tagging and collocations searching while V focuses on the task of analyzing user sentiments. Section VI shows how results can be aggregated for each brand and VII summarizes our findings.

## II. Data Collection

For this project we utilize a collection of user reviews extracted from Amazon, spanning 1996 to 2014, which is freely available for research purposes at [1]. More specifically, we use the dense 5-core subset for electronics reviews and the full metadata file from which we extracted entries related to electronics. The rationale for this choice is that we need a minimum amount of reviews to be able to extract meaningful results for a product. In addition, we built a dictionary for compound words based on the set of all Wiktionary pages titles. This set can be found at [4].

## III. Exploratory Data Analysis

### A. Formats

Both Amazon datasets store their data in JSON format, where each line contains a different JSON object. From the reviews dataset, the properties that turned out to be useful for our project are the following:

| | |
|---|---|
| **overall** | The star-rating of the product from 1 to 5. |
| **asin** | The unique id of the product. |
| **reviewText** | A textual user review. |

The properties of interest in the metadata file are enumerated below:

| | |
|---|---|
| **title** | The name of the product. |
| **categories** | The categories to which the product belongs, stored as an array. |
| **asin** | The id of the product. |
| **brand** | The name of the manufacturer. |
| **price** | The price of the product. |

Finally, the titles of all Wiktionary pages are stored in a tab-separated file, with one title per line. Spaces in the original page titles have been replaced by underscores.

### B. Distribution

To gain initial insights about the reviews dataset, we perform a simple statistical analysis. First, figure 1 shows that the review lengths follow a Pareto distribution where k = 1 and $x_m$ = 0. This distribution is actually frequently used for this type of text analysis.

### C. Correlations

No two properties are highly correlated. Reviews length show a slight positive correlation with price which would suggest that users are more willing to provide detailed feedback for products for which they spent more money. It is worth noting that price is definitely not linearly correlated to the overall score given by the users. More expensive does not mean better.

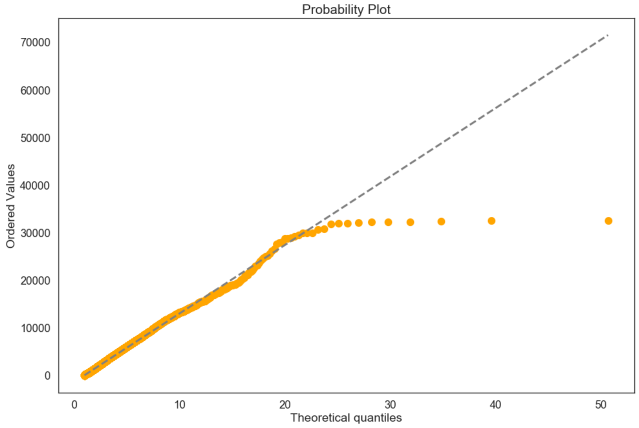

Fig. 1. Review length distribution and the Pareto distribution. Note that reviews length are capped at five thousand words, which roughly corresponds to thirty thousand characters

TABLE I
CORRELATION MATRIX

| | Price | Review length | Overall |
|---|---|---|---|
| **Price** | 1.00 | 0.22 | 0.01 |
| **Review length** | 0.22 | 1.00 | -0.08 |
| **Overall** | 0.01 | -0.08 | 1.00 |

## IV. Impressions and characteristics extraction

The goal of the first stage of the pipeline is to identify recurring impressions from users on a given product. To do so, we identify adjectives and superlatives that occur with high frequencies.

However, adjectives alone are not sufficient to describe a product. They are merely a way of paraphrasing the star-rating present on almost any selling platform. Indeed, suppose the qualitative adjective ”good” appears several times for a laptop. Which property of the laptop is generally considered good? It may be its performances, the quality of color rendering of its screen or virtually any other property. This is why we look instead at bigrams collocations to extract product characteristics. The categories that proved to be the most useful are listed along with a few examples below:

**adjective-noun**
*good quality, high performance, large screen*
**adverb-past participle**
*well made, poorly assembled, reasonably priced*
**verb-adverb**
*works well, runs great*

Before diving into implementation details we introduce the principal tool employed for this analysis, the Natural Language Tool Kit (NLTK) package for python. Among a plethora of useful features, NLTK allows us to tag words with their appropriate part-of-speech. The tags generated this way conform to the nomenclature used by the Penn Treebank Project[8].

Tags listed below are essential to our pipeline. Note however that this list is not exhaustive and that these tags can appear as prefix for other tags. For example, **JJ** is a prefix for **JJS** which is the group of superlative adjectives. We let the reader refer to the full tags description[8] for more information.

| | |
|---|---|
| **JJ** | Adjective |
| **RB** | Adverb |
| **NN** | Noun |
| **VB** | Verb |

### A. Impressions extraction

To extract first user impressions from the reviews we regroup and concatenate the review strings by product. Our approach then consists in a sequential scan of the reviews with one-word strides. Each time an adjective is detected, the two previous words are checked for negations. This permits capturing expressions such as *”good”*, *”not good”* or *”not so good”*. While this simple approach allows us to detect qualitative adjectives, it also tends to produce a large amount of unwanted results for compound words. For example, reviews for hard drives exhibit repetitions of the word ”hard drive” because ”hard” by itself is picked as an adjective by this method. We therefore devised a way of filtering out most of those unwanted results.

### B. Filtering compound words

Filtering pairs of words which happen to form a meaningful noun can be done efficiently by verifying if the pair has an entry in an English dictionary. However, for electronics reviews this requires an up-to-date dictionary with respect to modern words. Most readily available English words datasets either do not fit those requirements or are priced unreasonably high. Our solution is to use page titles from the Wiktionary, a free online dictionary project of the Wikimedia Foundation. The advantage of this approach is that it can also be used to filter out any tuple of words, also called n-grams, that happen to form a noun.

### C. Collocation Extraction

As explained in the beginning of this section, a better approach for characterizing products is to look for pairs of words that happen frequently. Such occurrences are called collocations. In general, we define collocations as apparition of n-grams at a higher frequency that one would expect if the elements of the n-grams were independent. Again, it is crucial to filter out bigrams that happen to form valid nouns as they perturb the characterization process.

There exists several metrics that can be used to detect interesting collocations. The most popular are raw apparition frequency, point-wise mutual information and ratios of likelihoods. As the name suggests, the first simply looks at the frequency at which pairs of words appear in the reviews. The idea being to select the most frequent ones. Point-wise mutual information (PMI) is defined as:

$$\textbf{PMI}(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1) \cdot p(w_2)} = \log \frac{p(w_2|w_1)}{p(w_2)}$$

It measures the amount of information provided by the occurrence of event $w_1$ about the occurrence of event $w_2$. In our case, $p(w_1)$ and $p(w_2)$ represent the frequency of the words in the reviews and $p(w_1, w_2)$ represents the frequency of the sequence of the words. Unfortunately, PMI has been shown to favour collocations that appear less often[9]. The last metric is the log of the ratio of likelihoods:

$$L(w_1, w_2) = \log \frac{L(H_1)}{L(H_2)}$$

Where $H_1$ is the hypothesis that the occurrence of $w_2$ is independent of the previous occurrence of $w_1$, ie: $P(w_2|w_1) = P(w_2|\neg w_1)$, and $H_2$ the hypothesis that it is in fact dependent.

In practice, the best collocations retrieved according to PMI and likelihood scores turned out to be complementary and we therefore chose a hybrid solution in which we select the top ranked collocations of each measure. Please refer to the IPython notebook of this project to see the full comparison between the metrics.

## V. Sentiment Analysis

To identify user opinions on certain product characteristics, we associate to each collocation a sentiment score. If the score is above an arbitrary threshold that we chose to be 0, the characteristic is considered positive while a negative score represents a negative connotation.

### A. Analysis with SentiWordNet

A first attempt of categorization was made using SentiWordNet[7], a lexical resource based on WordNet[5][6] which attributes to each word meaning a positivity, neutrality and negativity score.

We implemented a class on top of NLTK's support for SentiWordNet in order to provide a convenient interface to tokenize reviews and classify bigrams. However, this approach was not able to correctly classify opinions which are positive but written with negatively connoted words and vice versa. For example, using this technique, ”low price” will be negatively connoted because of the negative connotation of ”low” while in general it is a positive property of a product.

### B. Collocation occurrences counting

As the results produced by the first approach were not statisfactory, we devised our own method, based on counting the collocations. First, we introduce the notation used for our sentiment metric:

$S(c)$ The sentiment associated with the collocation c, either $1$ or $-1$ for respectively positive and negative sentiments.

$\eta(c, r)$ The number of times the collocation c appears in the review r.

$s(r)$ The sentiment score of the review $r$, $s(r) \in \{-1, 1\}$. The reviews with a star-rating lower than 3 are mapped to $-1$ whereas ratings higher than 3 are mapped to $1$. The remaining reviews are discarded.

$\sigma(x)$ A modified signum function that returns 1 if the $x > 0$, -1 if $x < 0$ and a random number $z \in \{-1, 1\}$ otherwise[1].

We define the sentiment metric as:

$$S(c) = \sigma(\sum_{r\,:\,s(r)=1} \eta(c, r) - \sum_{r\,:\,s(r)=-1} \eta(c, r))$$

In words, the sentiment of a collocation is defined as the sign of the number of times the collocation appears in strongly positively connoted reviews minus the number of time the collocation appears in strongly negatively connoted reviews. Therefore if the sign is negative, the collocation appeared more in negative reviews than positive ones and we deduce that it should be negatively connoted. The opposite case is analogous.

Even if this method seems simple, it produces very conclusive results. In fact, we alternatively trained a multilayer perceptron with the task of predicting the positiveness of reviews based on their collocations, by first vectorizing the reviews as the number of apparition of each bigrams and then training with the label obtained from the overall score. We obtained very similar results and therefore chose the solution with the occurrences counter since it is much more interpretable. The following table shows the results for some sentimentally marked bigrams:

TABLE II
Collocation counting method results

| Collocation | Score | Collocation | Score |
|---|---|---|---|
| low price | 1 | low light | 1 |
| high quality | 1 | original battery | 1 |
| low quality | -1 | is intuitive | 1 |
| well made | 1 | outstanding lens | 1 |
| poorly made | -1 | weak signal | -1 |
| good product | 1 | great processor | 1 |
| special edition | 1 | good price | 1 |
| dead pixels | -1 | great bag | 1 |

## VI. Application to Brand Names

It is interesting to examine how brands are perceived by the customers based on extracted collocations and whether certain brands tend to make products that exhibit common characteristics.

[1] Note that this only happen for bigrams that are either not present any review (this does not happen in our analysis), or perfectly balanced across reviews. In the latter case those would most likely indicate that they are not collocations of particular interest.

## A. Brand Rankings

We assign a score to each product based on the collocations extracted from its reviews. Those collocations are associated with a score which is then aggregated over all the products of each brand. Before defining the score metric we introduce the following notation:

$score(p)$ The score of the product p.

$C_p$ The set of all the collocations obtained with our pipeline that appear at least once in the reviews for product p.

$f(c,p)$ The frequency of apparition of collocation $c$ in the reviews of product $p$, expressed as a percentage.

$S(c)$ The sentiment associated with the collocation c as developed in section V-B.

The final score of a product is then defined as:

$$score(p) = \sum_{c \in C_p} f(c,p) \cdot S(c)$$

Which is simply the sum of all collocations sentiment multiplied by their respective apparition frequency. The score can be qualitatively analyzed as follows:

**High positive score** if the reviews of the product contain lots of collocations and those collocations are mostly positive.

**Score close to 0** either if the product contains lots of collocations but they are equally good or bad, or few collocations.

**High negative score** This case is analogous to the first one.[2].

If we have two different products with reviews that in one case contains a single positive collocation of percentage K, and in the other case contains multiple different positive collocations of total percentage K, the two products will have the same score, as we would expect.

Finally, to get the score of a brand we simply take the average of the score of each of its products.

## B. Brand Characteristics

To obtain characteristics by brand instead of products we define the following rule:

*A collocation **c** is associated to a brand **b** if and only if **c** is a collocation found in reviews of at least **k** different products of **b**.*

# VII. Results

Table III present the result of collocation extraction for a few products while figure 2 shows the scores of some famous brands. We use $k = 3$ as the threshold for characteristic association. Table IV shows a few collocations associated to famous brands. The quantities indicate how many products have such collocations. We let the reader refer to the IPython Notebook associated to this project for the full results and their visualization.

## A. Possible improvements

The pipeline in its current state does not discriminate collocations based on the reviews from which they originate. This would allow malicious user to fool the system by crafting reviews specifically targeted to produce collocations with high PMI or log of likelihoods scores.

Although bigram extraction produces interesting results, it still captures a significant number of falsely interesting collocations and remain a very limited way of characterizing products as nuances and complex opinions cannot be captured.

TABLE III
Top-5 bigram collocations extracted from reviews

| |
|---|
| **AmazonBasics 11.6-Inch Laptop Sleeve**<br>*heavy duty, well made, reasonably priced, little bit, highly recommended* |
| **Water, Dirt and Shockproof iPad Mini Keyboard case**<br>*personal preference, new trent, adjustable stand/arm, high quality, heavy duty* |
| **To Go Velocity Series 40315 HDMI Cable**<br>*high speed, great price, new samsung, high quality, much cheaper* |

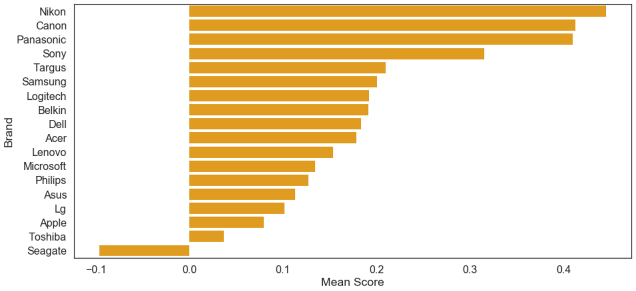

Fig. 2. Rank of some famous companies according to the reviews.

TABLE IV
Most frequent characteristic associated to famous brands

| **Brand: top-characteristic, product occurrences** | | | |
|---|---|---|---|
| Amd: | dual core, 6 | Philips : | good sound, 14 |
| Asus: | dead pixels, 19 | Logitech: | high quality, 50 |
| Belkin: | well made, 30 | Targus: | well made, 25 |
| Canon: | low light, 97 | Samsung: | great picture, 18 |

# VIII. Conclusion

In this project we have successfully extracted from user reviews basic characteristics about the reviewed products and have been able to attribute to each characteristic a positivity score. In addition to its informative value, the positivity score was used to rank and assign characteristics to brands. The pipeline developed this way allows feeding fresh data to the model to update products and brands scores. In addition, the pipeline is applicable to any category of product and with a bit more tuning can have real-world applications on most electronic commerce platforms.

[2] Note that this is less likely to happen as poorly rated products do not live very long on most selling platforms

# References

[1] Julian McAuley, Amazon reviews dataset, University of California San Diego, 2016. *http://jmcauley.ucsd.edu/data/amazon/*

[2] R. He, J. McAuley, (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, In WWW *PDF*

[3] J. McAuley, C. Targett, J. Shi, A. van den Hengel (2015). Image-based recommendations on styles and substitutes, In SIGIR *PDF*

[4] Wiktionary, all page titles in namespace zero dataset, The Wikimedia Foundation. Version: 07 December 2017 12:22. *https://dumps.wikimedia.org/enwiktionary/*

[5] George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[6] Christiane Fellbaum (1998, ed.). WordNet: An Electronic Lexical Database. MIT Press. Cambridge, Massachusetts

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Istituto di Scienza e Tecnologie dell'Informazione Consiglio Nazionale delle Ricerche.

[8] Beatrice Santorini (1990). Part-Of-Speech Tagging Guidelines for the Penn Treebank Project, 3rd revision, 2nd printing. Department of Computer and Information Science, University of Pennsylvania. *https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf*

[9] Christopher D. Manning and Hinrich Schütze (1999). Collocations. In Foundations of Statistical Natural Language Processing MIT Press. Cambridge, Massachusetts. *https://nlp.stanford.edu/fsnlp/promo/colloc.pdf*