Chapter 4

A Proposal for Brand Analysis with Opinion Mining

Francisco Javier Moreno Arboleda,
Gustavo Andrés Angarita Velásquez and
Gustavo León Preciado Jiménez

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66567

Abstract

The popularity of e-commerce sites has increased the availability of product reviews, most of which are overlooked by customers because of their large number. Opinion mining, a discipline that aims to extract people's opinions regarding some topic from reviews, was developed to address this situation. However, the individual interpretation of the reviews is not enough to take advantage of the massive datasets available on the web; a meaningful summary of the set of opinions is necessary to give users an overall insight into the opinions. We propose a system to extract information from Amazon product reviews, which focuses on a time-varying comparison among different brands in a given Amazon product department. In this system, the results are summarized so that users can get a representative and detailed overview of the opinions of (possibly) hundreds of other users regarding the strong and weak points of several brands. This information can be used by customers who want to find high-quality products, or by the enterprises themselves, which could find the aspects with a higher impact in the public perception.

Keywords: e-commerce, temporal evolution, brand perception, aspect-level sentiment analysis, Amazon product reviews, summarization

1. Introduction

In the past 5 years, thanks to the progress in communication technology, the act of publishing opinions about topics or products on the Web has become increasingly popular. These opinions are generated by users in the form of reviews, and they are published in places such as specialized websites (including Amazon, Barnes and Noble, and Best Buy), blogs and microblogs, comments in social networks, and critiques in specialized magazines (e.g., *Nature* [1]). For example, in Amazon, the number of reviews published per year in the electronics department has increased from around half a million in 2010 to more than two and a half million in 2013 (data taken from Ref. [2]). Experts in big data predict an increase of around 4300% in the yearly data production rate by 2020 [3].

Although most of these opinions are published in the form of text, their presentation in other formats such as images and videos is growing in popularity (e.g., some people use YouTube to publish videos in which they present their experiences with certain products).

In this chapter, the word *opinion* refers to the assessment some user has on something (a topic or product); and the word *review* refers to the text or, in general, the content from which the opinion can be extracted in whichever format it is presented. Analyzing these opinions is the main objective of opinion mining. The analysis of these opinions can be helpful in contexts as follows:

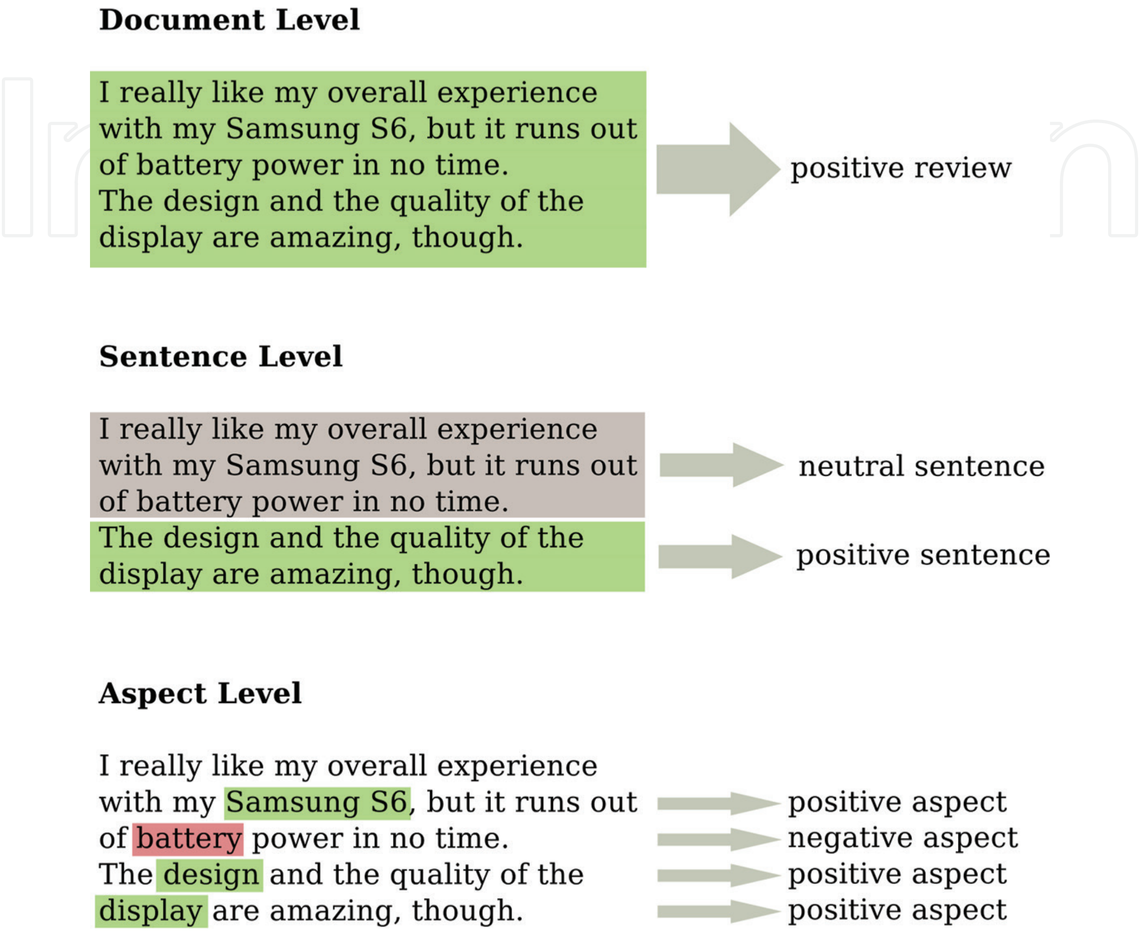
1. Production: here, opinion mining can be used to find defects in a product or aspects that are prone to be enhanced. For instance, a cell phone can be made with a sturdier material if users complain about its fragility.

2. Customer service: here, the satisfaction of users (buyers, tourists, etc.) can be measured using their comments. For example, the selection of entertainment content offered to passengers in an airship can be improved if their tastes in music or movies are inferred from their reviews about previous flights; and the packaging of products for their delivery might be improved if many users report having received their orders in a bad condition (damaged, bent, etc.).

3. Entertainment and sport industries: the impact of advertisement on a certain audience can be increased by using their favorite artists and sportsmen in ads. For example, comments about an artist in social networks can help in predicting the impact of using him or her in an ad about clothing for teenagers.

Sentiment analysis is the core of opinion mining. The main goal of sentiment analysis is to classify the sentiments expressed in the opinions of users with regard to something (e.g., some topic, some product, or an aspect of a product), i.e., to find the polarity of the opinions. Sentiment analysis can be carried out at different granularity levels (**Figure 1**): in the lowest level of granularity, the opinion contained in the entire document (review) is extracted and classified as a single sentiment; in higher levels of granularity, the opinions contained in individual paragraphs or sentences (or video segments in the case of video reviews) can be extracted and classified; in the next level, the opinions regarding particular aspects, i.e., properties of the product in question, are analyzed. For example, consider a mobile phone, its screen and its battery are aspects of the phone. As the level of granularity of sentiment analysis increases, the extracted sentiments become more detailed and more relevant as a source of information; however, the complexity of the sentiment analysis algorithms also increases.

Once the classification stage is over, the next step is a process known as *summarization*. In this process, the opinions contained in massive sets of reviews are summarized. This task can be approached from different perspectives depending on the kind of information that is of interest to the users. The simplest forms of summarization produce naive aggregate results for sets of reviews, such as a count of total positive opinions versus total negative opinions. Other approaches extract representative sentences or show the words that are most frequently mentioned by users in the reviews. Other approaches, such as the one proposed here, reveal the relationships between the products or topics and extract additional relevant information from the reviews. An image depicting the opinion mining stages can be found below (**Figure 2**).



Document Level

I really like my overall experience with my Samsung S6, but it runs out of battery power in no time.
The design and the quality of the display are amazing, though.

Sentence Level

I really like my overall experience with my Samsung S6, but it runs out of battery power in no time.
The design and the quality of the display are amazing, though.

Aspect Level

I really like my overall experience with my Samsung S6, but it runs out of battery power in no time.
The design and the quality of the display are amazing, though.

Figure 1. Levels of granularity of opinion mining.

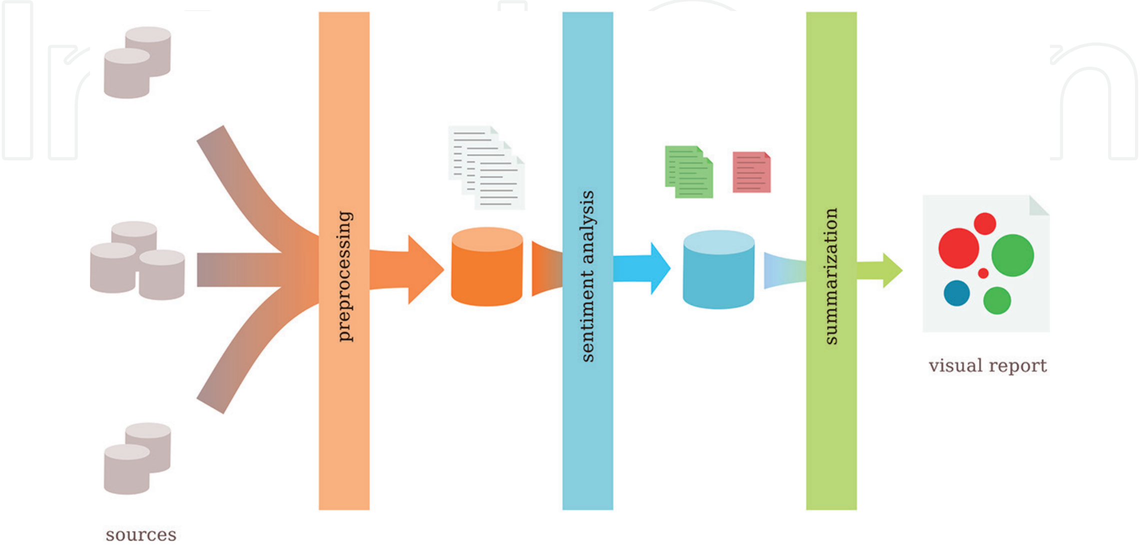


Figure 2. Opinion mining process.

2. Problem statement

Knowing the opinions of users about a brand and its products is useful in many situations; e.g., people often search for information regarding the strong and weak points of some brand in a group of products (department) before actually purchasing them; companies (brands) seek data about the aspects of their products that should be improved in order to satisfy their customers and increase their market share in that product department. The opinions of users can be used as a measurement of the quality and the experience of a company in a set of products; this might be of interest for potential employees pursuing professional growth in a certain field. In these situations, the need for tools to summarize high volumes of product reviews is evident.

Product reviews are a key in identifying the aspects of a product from which opinions originate and in establishing a comparison between products, product departments, and brands. In order for these comparisons to be valid, the sentiment analysis has to be executed over either the same product, or a set of products in the same department, or a group of brands with some product departments in common. This enables the use of domain-specific classifiers, which usually yield better classification results than their cross-domain counterparts [4].

For the sake of illustration, a product department (e.g., mobile devices) common to two brands is considered (e.g., Sony and Samsung). Given a list of product reviews and a set of aspects shared by all the products in this department (e.g., their battery and their display), we like to find, for each brand, the opinions with regard to each particular aspect. Moreover, in order to facilitate the analysis of the evolution of opinions in this product department, the user perception in different time intervals is aggregated and displayed. This enables, for instance, the discovery of periods of time in which a radical change in the public perception of some brand occurred. This information can be used to recognize aspects that caused the sudden opinion changes.

For the proposed analysis, which includes using a machine learning model trained with real-world data, both the domain and the format of the reviews must be known in advance. Regarding the first point, only reviews of products that belong to the same department should be used. Regarding the second point, only texts written in natural language (which is the usual format of product reviews in websites such as Amazon, Barnes & Noble, and Best Buy) are considered. The use of this format requires some additional processing before execution of the sentiment analysis, including the extraction of the linguistic elements of each review, which can be handled using existing tools.

The algorithm proposed in this chapter yields these key pieces of information:

a. A summary, in the aspect level, for a set of reviews about products in some target department.

b. A comparison of the summaries obtained for different brands.

This analysis can be used by both companies and clients: to the former, it is a reliable way to find aspects in their products that can be leveraged to obtain competitive advantage over other companies; to the latter, it is a useful source of advice in the search for high-quality products.

3. Related work

In recent years, several authors have proposed different approaches for opinion mining. Some of them focus on sentiment analysis [5–7], others in summarization [8], and some others in the observation of the evolution of opinions [9, 10], among other topics. Some terms of high relevance in related research are defined in the next section.

3.1. Preliminary concepts

3.1.1. Dependency grammar

Dependency grammar is an approach for the analysis of natural language sentences. It is based on the idea that each word in a sentence, except for the finite verb, directly depends on some other words through a grammatical relation. These relations link together words that are structurally related even if they are not adjacent in the sentence. For instance, in the sentence "I really like my overall experience with my Samsung S6," there is a relation (called nominal subject) between the words "I" and "like" [11]. The Stanford Parser API can generate a dependency graph for any input sentence as a means to represent these dependency relations; however, since it uses a machine learning model to identify the dependencies, the results are not always accurate and are sensitive to punctuation and spelling mistakes [5].

3.1.2. Direct neighbor relation

Two words are said to be connected through a direct neighbor relation if they are adjacent and neither of them is a *stop word*, i. e., frequently used words without a meaning of their own, such as articles, pronouns, and some prepositions [6].

3.1.3. Machine learning and classifiers

Machine learning facilitates the adaption of models to different domains and datasets. Automatic classifiers are an example of this kind of application. Once the model of the classifier has been trained, it can take an object as input and then output a prediction of the class to which the object belongs.

3.1.4. Clustering

A cluster is a collection of elements that are related according to some criteria (e.g., closeness). In sentiment analysis, clustering can be used to abstract and simplify opinions in order to facilitate their classification.

3.2. Aspect-level sentiment analysis

In Ref. [6], a method is presented for obtaining the polarity of opinions at the aspect level by leveraging dependency grammar and clustering. Their work is the base for the sentiment analysis method proposed in this chapter.

3.2.1. Clustering

The clusters proposed in Ref. [6] consist of a *head* (the target aspect or, in their terms, *the target feature*) and a set of words that describe the head, called *opinion words*. In their approach, to find the cluster of opinion words for some target aspects, the following procedure is used: a graph is built using the dependency relations (which can be found using the Stanford Parser API [5]) and direct neighbor relations; then, a cluster is created for each noun, and then each word is attached to the cluster of the closest noun in the relations graph; finally, the clusters that are close to the cluster of the target aspect are merged with it. The clusters are close if their heads are separated by less than some threshold distance θ in the relations graph; making the threshold too small would result in an important loss of information; on the other hand, making it too large would result in the cluster including (almost) all the words in the sentence, thus making clustering pointless. The tests presented in Ref. [6] showed that the optimal θ for aspect-level sentiment analysis of product reviews is 3.

3.2.2. Summarization of opinions by brand

In Ref. [12], a method is proposed to detect events linked to some brand within a period of time. The authors use the term *event* to refer to topics that rise in popularity on microblogs during short-time intervals (a day, a few hours, etc.), e.g., the final match of a football tournament or the new song of a famous artist. Although their work can be manually applied to several periods of time, the temporal evolution of the opinions is not explicitly shown by their system. Moreover, the information extracted by their model is more closely related to the brand itself than to the aspects of products of that brand.

3.2.3. Temporal evolution of the opinion

This topic has been approached in Refs. [9, 10]. Understanding which aspects are the most influential in the change of opinion polarity is of utmost importance. Systems aimed at presenting the temporal evolution of opinions are often accompanied by a visual aid as a way to give end users a more intuitive representation of the results, as opposed to raw numbers. The approaches proposed by Cao et al. [9] and Schouten et al. [10] take this into account; however, there are important differences between their system and ours in both the domain (in their case, politics and hotel and catering industry, respectively) and conceptual aspects. In Ref. [9], e.g., the system focuses on politics and on the detection of events (such as recent political speeches or voting results announcements), and the reviews under analysis come from microblogs. On the other hand, in [10], the domain is the hotel and catering industry, and the main contribution of their approach lies in the visualization of changes of the general opinion in several dimensions (spatial, temporal); for instance, one of their charts compares different kinds of trips and groups the results by country of origin of the reviewer.

3.2.4. Classifiers

Machine learning-based classifiers are more versatile than their deterministic counterparts in practice due to the possibility of training them automatically for any domain as long as an appropriate dataset is available, i.e., a dataset that is large enough and is correctly labeled. Among the applications of machine learning for data mining, the most frequently used models are Naive Bayes, SVM, and Logistic Regression. The first one is renowned for its precision in dealing with small datasets [13]; the second one, on the contrary, is very precise for large datasets because of its low bias [13], but in order for it to work correctly, the objects with different labels must be clearly divided into two separate clusters in the space given by their features, i.e., there must be a clear gap between the classes; the third classifier has a low bias and a high variance, and it can be found in different variants such as MaxEnt and Softmax. The Logistic Regression classifier is very useful in the case of opinion mining since it can adapt to the most common kind of datasets in sentiment analysis: those with blurred borders between classes.

3.5. Other related work

Similar systems have been proposed in the past by other authors trying to tackle different parts of the problem our system attacks, but they are focused at specific parts of the opinion mining pipeline and have a more limited scope. A table listing a few representative approaches is presented below (**Table 1**).

Ref.	Granularity level	Polarity extraction technique	Main contribution	Input data	Polarity evolution	Summarization and comparison between brands
Brand-related events detection, classification and summarization on Twitter	Document*	SVM	A technique that can be used to detect events in microblogs; summarization technique that takes into account for three kinds of polarities: positive, neutral, and negative	Twitter	No	No. Only summarization by event is considered
SocialHelix: visual analysis of sentiment divergence in social media	Document*	Multinomial Naive Bayes	Novel visualization technique to show changes of opinion over time for different classes of users (e.g., social groups, political parties, etc.)	Twitter	Yes	Not for brands, but it does for classes (groups) of users
OpinionSeer: interactive visualization of hotel customer feedback	Aspect/feature	Dictionary of words previously marked as positive or negative	Visualization technique for the generated summary using figure segmentation, point scattering, and transparencies	Trip advisor	Yes	No. It focuses on tourism

Ref.	Granularity level	Polarity extraction technique	Main contribution	Input data	Polarity evolution	Summarization and comparison between brands
Feature specific sentiment analysis for product reviews {referencia}	Aspect/feature	Machine learning, rule based	Aspect-level polarity extraction technique with dependency relations	Datasets from "*Exploiting Coherence for the Simultaneous Discovery of Latent Facets and associated Sentiments* "(Lakkaraju et al.) and "Mining and Summarizing Customer Reviews" (Hu et. al.)	No	No

Table 1. Other related work.

4. Proposed system

The proposed system transforms a set of reviews (in the form of natural language) into a summary of the opinions in order to facilitate their comprehension. The process can be broadly described in three steps as follows:

1. A set of reviews for products of different brands in the same department are collected; by constraining them to one department, most of the aspects can be assumed to be shared by the reviewed products, thus reducing the ambiguity that would arise if products from multiple domains were grouped and analyzed together [7].

2. The aspect-level sentiments contained in the reviews are extracted by using a combination of machine learning techniques.

3. A detailed summary is generated for each brand in order to show its strengths and weaknesses in regard to that department.

A more detailed explanation of each step is presented in the next section.

4.1. Auxiliary algorithms

In the first step, reviews about products in the same department are taken from a database of Amazon user reviews, such as [2]. This database contains almost 20 years (1996–2014) worth of reviews extracted from the electronics department, which have been preprocessed for easier handling in opinion mining research. The dataset contains not only the plain text of the reviews but also important metadata such as the brand of the product and the *helpfulness index* of the review. This helpfulness index, which is characteristic of sites such as Amazon and YouTube, is the quotient between the number of positive votes given to the review and the total number of votes it has received [2].

The next step is the extraction of sentiments at the aspect level, which is accomplished by leveraging the Stanford Parser API for natural language processing and the clustering algorithm presented in Ref. [6]. The process for obtaining the clusters will be illustrated with an example:

Let $r = (x, t, p, h, b)$ be a review, where:

- x := review text
- t := time in which it was published
- p := name of the product
- h := helpfulness index = [positive votes, total votes] = [h+, htotal]
- b := brand of the product

An instance of r could be:

{

x: "I really like my overall experience with my Samsung S6, but it runs out of battery power in no time.",

t: "02 10, 2010",

p: "Samsung S6",

h: [3, 4],

b: "Samsung"

}

The next step consists in passing the review text (x) as input to the Stanford Parser API in order to obtain the dependency relation graph, to which the direct neighbor relations will be added. The graph obtained for the example review is shown in **Figure 3**.

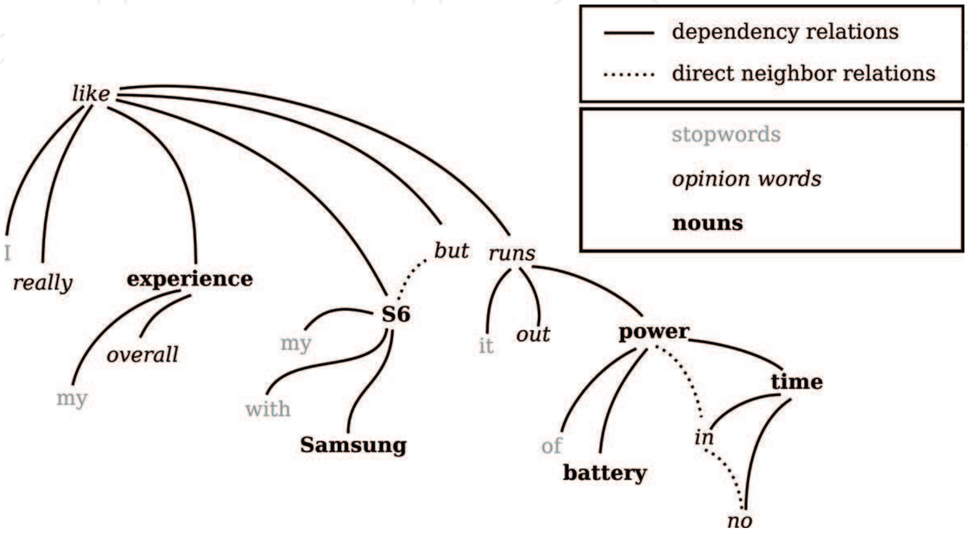


Figure 3. Graph obtained for the example review with direct neighbors and dependency grammar relations.

If the review contains more than one sentence, the Stanford Parser returns a separate graph for each one. Thus, for each text x, the API will return a set of graphs $G = \{g_1, \ldots, g_k\}$, where k is the number of sentences x contains. Each relations graph g will be used to build the clusters that are sent as input to the classifier for sentiment analysis. The algorithm proposed in [6] for the construction of the clusters will be described next, along with an example of its results.

Let A = {screen, battery, design, signal, and camera} be the set of target aspects for the sentiment analysis, which must be nouns; this set is defined by the analyst. For each $a \in A$ in g, an opinion word cluster will be produced through the getCluster algorithm, which is described in Algorithm #1 (**Table 2**).

```
Begin
 Let N be the list of nouns in the graph g .
 For each $n_i \in N$:
  Make $n_i$ the head of the ith cluster, $c_i$ .
 For each word $w \in (g - (N \cup \{s \mid s \text{ is a stopword}\}))$:
  $k \leftarrow \arg\min n_i \in N \{dist(w, n_i)\}$,
  where $dist(w_1, w_2)$ is the number of edges in the shortest path between $w_1$ and $w_2$.
  Add w to $c_k$.
 For each $c_i$:
  For each $c_j$ with $i \neq j$ and $n_j \neq a$:
  Merge $c_i$ with $c_j$ if $dist(n_i, n_j) < \theta$
 Return the cluster
End
```

Algorithm: getCluster(a, g, θ)

Input:
a: target aspect
g: relations graph for the sentence
θ: closeness threshold
**the relations graph used in this example was based on the sentence "I really like my overall experience with my Samsung S6, but it runs out of battery power in no time."*

Output:
Opinion words cluster for *a*. Example:
battery:{runs, out, power, time}

Table 2. Algorithm #1: getCluster.

In our example, with a = "battery" and $\theta = 3$, this cluster is obtained: battery: {runs, out, power, time, and no} (**Figure 4**). All the clusters for this graph are presented in **Figure 5**.

Notice that the example sentence contains two opinions, one about the battery and the other about the product in general; however, this global opinion is not captured by the aspect-specific algorithm unless “Samsung S6” is treated as target aspect. With this in mind, the original algorithm was modified by adding a preprocessing step before the construction of the graphs as suggested by [12]. During this process, a family of sets $A^* = (A_i^*)$ $i \in A$ is defined. This family contains sets of synonyms and common misspellings of the nouns in A. The elements in these sets will be replaced by the corresponding element in F in order to increase the accuracy of the analysis. In addition, nouns such as product, buy, and the name of the product are replaced by the pseudo noun #General, which is also given a set of words in A^* (the number sign is used to avoid ambiguities when the actual word *General* is present in a review); this can be seen as a particular case of synonym replacement, but it serves a different purpose: it intends to capture the opinions that are not linked to specific target aspects but are still expressed by users and thus should not be ignored. For the example given above, one possible A^* would look like this: A^* = {#General : {product, buy, purchase}, screen : {display, scren, sreen}, battery : {battery power, power, batery, charge}, design : {appearance, desin}, signal : {reach}, camera : {lens, photograph, camra}}. After preprocessing the original x, the result would be:

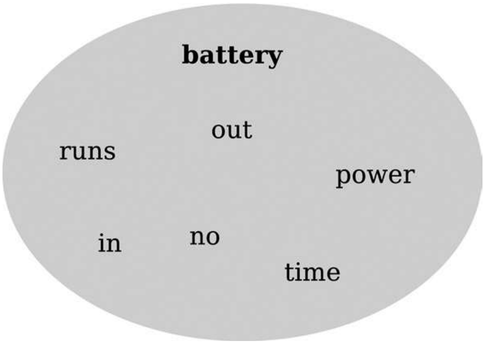


Figure 4. Cluster obtained for the battery aspect.

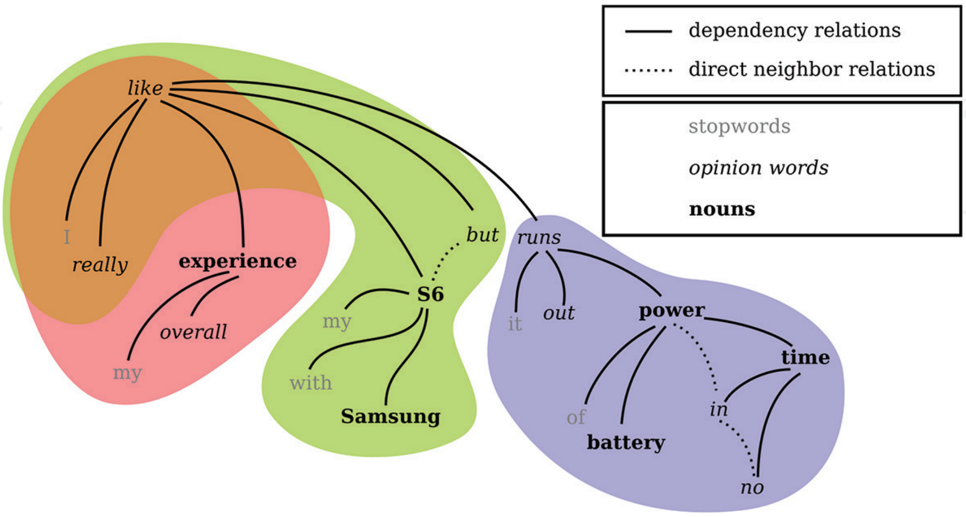


Figure 5. Resulting clusters for all the nouns in the relations graph.

“I really like my overall experience with my #General, but it runs out of battery in no time.”

Had the #General been omitted, an important part of the review, corresponding to overall satisfaction with the product, would have been missed by the system, thus leading to inaccurate understanding of the opinions. The function used to preprocess the review text will be described in Algorithm#2 (**Table 3**) preprocess.

```
Begin
  Add the name of the product, $p$, to the set corresponding to #General in the local $A^*$.
  For each word $w \in x$:
    For each aspect $a \in A$:
      For each $a^* \in A^*_a$:
        If $w = a^*$: replace $w$ with $a$.
End
```

Algorithm: preprocess(x, p, A, A*)

Input:
x: review text
p: name of the product
A: set of target aspects
A: family of synonyms and common misspellings*

Output:
Preprocessed review text. Example:
“I really like my overall experience with my #General, but it runs out of battery in no time.”

Table 3. Algorithm #2: preprocess.

For our example, once *x* is preprocessed, the Stanford Parser API returns the graph, as shown in **Figure 6**:

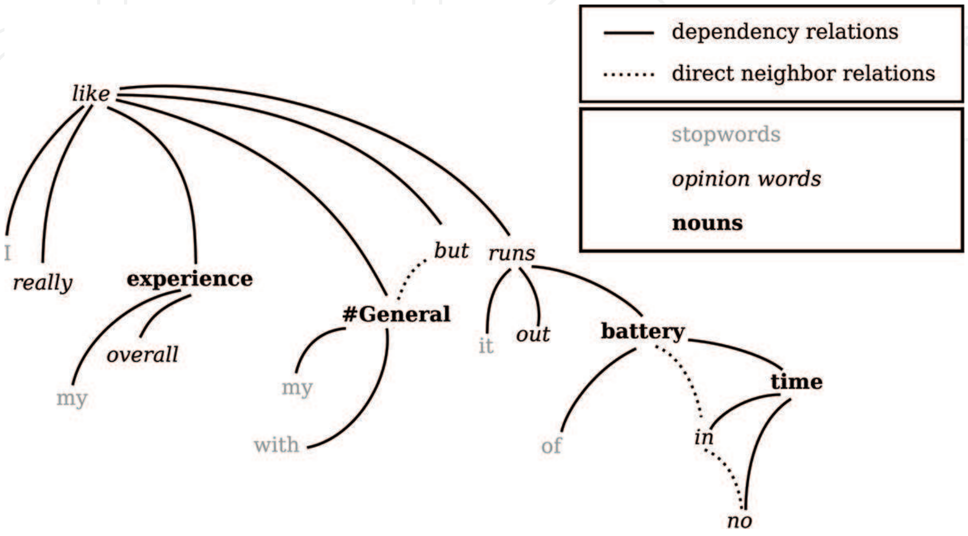


Figure 6. Graph obtained for the example review after running the preprocess algorithm.

Notice that the direct neighbor relations connect words that are intuitively related, such as "but" and "#General" (highlighting contrast in this case), but which are not linked by dependency relations according to the Stanford Parser. Also notice that other intuitive relations such as the one between "battery" and "out" are captured by the dependency rules but not by the direct neighbor relations.

After running getClusters algorithm on this new graph, two clusters are obtained: one for battery (**Figure 7**) and another one for #General (**Figure 8**). All the clusters in the preprocessed graph are presented in **Figure 9**.

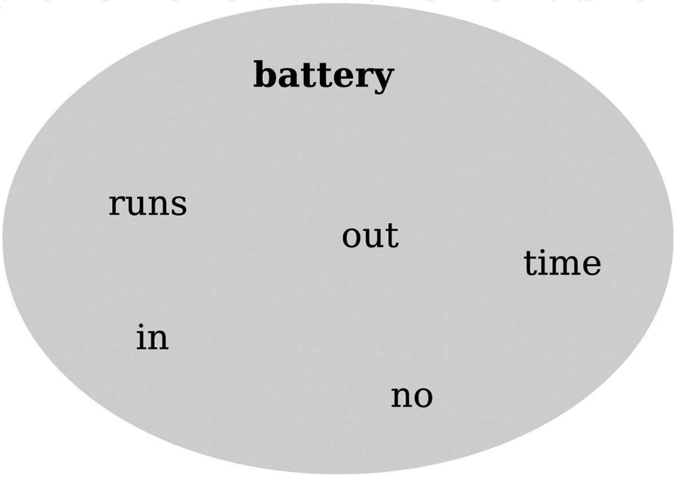


Figure 7. Cluster obtained for the battery aspect after running the preprocess algorithm.

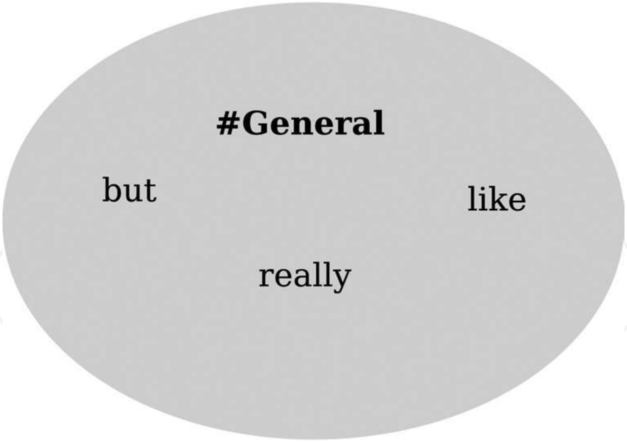


Figure 8. Cluster obtained for the #General aspect.

4.2. Main procedure

The proposed system uses the procedures presented above to generate a comparative report showing, for each brand, the evolution of opinions regarding each aspect in the set of target aspects. For the sentiment analysis, i.e., polarity extraction, any classification algorithm that is well suited to the problem (as described in previous sections) can be used.

Logistic Regression and a tweaked SVM that supports overlapping classes are both possible approaches. The getPolarity() function used below represents the invocation of one such procedure over a bag of words, which will result in a polarity of either +1 or –1. The following algorithm (**Table 4**) shows the integration of each of these procedures and shows the proposed summarization scheme, which uses online algorithms to compute the mean and an approximation of the variance in a single iteration over the entire dataset (after training). This summarization scheme will ignore the reviews that have no nouns in common with the set of target aspects because all the present nouns cannot be guaranteed to refer to the product itself; reviews with zero positive votes and one or more negative votes are also ignored altogether because they contain information that all the readers considered useless. Unless stated otherwise, all the matrices have initial values of zero in every position.

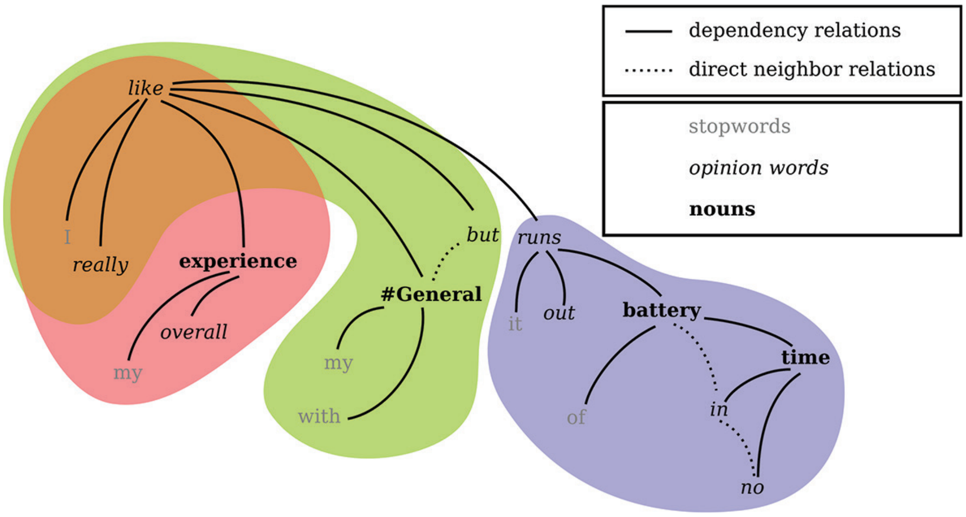


Figure 9. Resulting clusters for all the nouns in the relations graph after running the preprocess algorithm.

Algorithm: summarize()

Input:
d: target department
I: list of time intervals
B: set of brands to compare
R: set of reviews
A: set of target aspects
A: family of sets of synonyms and common misspellings*
These parameters will be described in more detail below.

Output:
M: matrix of mean polarities for each target aspect.
V: matrix of variance in polarities for each target aspect.
These matrices will be described in more detail below.

Table 4. Algorithm #3: summarize.

Let *d* be the target department; e.g., electronics.

Let *TI* be the list of time intervals, which depends on both the time spanned by the reviews set and the length or amount of intervals defined by the user. For example, if the reviews have been published between the 10th of February of 2010 and the 9th of March of 2010, the user may choose to split this time span, e.g., in four intervals (in which case each interval would have 7 days) or in intervals with a length of 2 days (in which case, 14 intervals would be created). For the current example, *TI* contains four intervals of 7 days.

Let *B* be the set of brands, e.g., *M* = {Samsung, Sony}.

Let *R* = (r_i) be the set of reviews, *A*, *A**, and *G* the sets defined in section {4.1}. In our example, *R* is

R = [{*x*: "I really like my overall experience with my Samsung S6, but it runs out of battery power in no time.", *t*: "02 10, 2010", *p*: "Samsung S6", *h*: [3, 4], *b*: "Samsung" },

{*x*: "The charge does not even last for a single day.", *t*: "02 15, 2010", *p*: "Samsung Galaxy Note", *h*: [18, 19], *b*: "Samsung"},

{*x*: "Best product ever!", *t*: "02 18, 2010", *p*: "Xperia Z5", *h*: [13, 20], *b*: "Sony"},

{*x*: "Great improvement with the battery life. The rest of the product remains amazing as always.", *t*: "02 19, 2010", *p*: "Samsung S7", *h*: [45, 45], *b*: "Samsung"},

{*x*: "The Samsung S7 is so much better than the S6.", *t*: "02 25, 2010", *p*: "Samsung S7", *h*: [0, 0], *b*: "Samsung"},

{*x*: "Even though the Xperia is too heavy, I really like the camera and the battery life in this phone!", *t*: "03 08, 2010", *p*: "Xperia Z5", *h*: [50, 52], *b*: "Sony"},

{*x*: "The display is just amazing!", *t*: "03 08, 2010", *p*: "Samsung S7", *h*: [0, 0]}, *b*: "Samsung"}]

The following matrix will be used as an auxiliary variable in our algorithm:

(a) *N*: matrix of extracted polarities count. This stores the total number of extracted polarities indexed by brand, time interval, and target aspect at a given point in the execution of the algorithm.

The following matrices will be the output of the summarize() function:

(b) *M*: matrix of mean polarities. This is obtained by using an online algorithm to compute the mean of the polarity (sign) times the weight given by the helpfulness.

(c) *V*: matrix of variances for the polarities. These approximate variances are computed by using an online algorithm.

```
Begin.
  Let R be the set of product reviews in the target department.
  Each $r \in R$ is determined by a tuple $(x, t, p, h, b)$ as described above.
  Let T be the set of dates in the time spanned by the reviews.
Let TI be the set of intervals defined by the user
Define dateMap : $T \rightarrow TI$, as a function that maps each date $t \in T$ to the corresponding interval $i \in TI$.
For each $r \in R$, if $(h = [0, 0])$ or $(h_+ > 0)$:
  $x \leftarrow$ preprocess$(x, p, A^*)$.
  Initialize G for x using the dependency and direct neighbor relations.
  $i \leftarrow$ dateMap$(t)$.
  For each $g \in G$:
    $A' \leftarrow \{n \mid (n \in g)$ and ($n$ is a noun) and $(n \in A)\}$.
    If $A'' = \{\}$:
      go to next r
    Else :
      For each a in $A''$:
        $c \leftarrow$ getCluster$(a, g, \theta)$
        $h' \leftarrow$ if$(h = [0, 0])$: 0, 5; Else:$h_+/h_{total}$
        $y \leftarrow$ getPolarity$(c)^* h'$
        $N[b][i][a] \leftarrow N[b][i][a] + 1$
        $delta \leftarrow y - M[b][i][a]$
        $M[b][i][a] \leftarrow (M[b][i][a]) + \frac{delta}{N[b][i][a]}$
        if $N[b][i][a] > 1$:
        $V[b][i][a] \leftarrow \frac{V[b][i][a]^*(N[b][i][a] - 2) + delta^*(y - M[b][i][a])}{N[b][i][a] - 1}$
End.
```

The resulting matrices ***M*** and ***V*** would look like the ones in **Table 5** for the current example.

brand	Interval	Aspect	M	V
Samsung	1	#General	0.75	0
		Battery	–0.848684211	0.019477147
	2	#General	1	0
		Battery	1	0
	3	#General	0.5	0
	4	Screen	0.5	0
Sony	2	#General	0.65	0
	4	**#General**	**–0.961538462**	**0**
		Camera	**0.961538462**	**0**
		Battery	**0.961538462**	**0**

Table 5. Summary matrices: mean and variance.

In this way, our proposal can be considered as the base for a visual representation strategy for giving users a quick and effective understanding of what the strengths and weaknesses of some brands are for a given product department in which all products share a set of common aspects. The temporal evolution of the general user opinion about these aspects could also be represented. This information can be used by companies to create effective marketing campaigns or to improve their products based on the user feedback.

Author details

Francisco Javier Moreno Arboleda*, Gustavo Andrés Angarita Velásquez and Gustavo León Preciado Jiménez

*Address all correspondence to: fjmoreno@unal.edu.co

Universidad Nacional de Colombia, Medellín, Colombia

References

[1] McAuley J., Targett C., Shi J., van den Hengel A. Image-Based Recommendations on Styles and Substitutes. SIGIR. 2015.

[2] de Marneffe M. C., Manning C. D. Stanford Typed Dependencies Manual. 2015.

[3] Chen D., Manning C. D. A Fast and Accurate Dependency Parser using Neural Networks. Proceedings of EMNLP 2014. 2014.

[4] Schouten K., Frasincar F. Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering. 2016;**28**(**3**):813–830.

[5] Macmillan Publishers Limited. Nature Reviews [Internet]. 2016. Available from: http://www.nature.com/reviews/index.html [Accessed: 2016-09-04]

[6] Zhang Y., Hu X., Li P., Li L., Wu X. Cross-Domain Sentiment Classification-FeatureDivergence, Polarity Divergence or Both? Pattern Recognition Letters. 2015

[7] Cao N., Lu L., Lin Y. R., Wang F., Wen Z. SocialHelix: visual analysis of sentiment divergence in social media. Journal of Visualization. 2015;**18**(2):221–235.

[8] Mukherjee S., Bhattacharyya P. Feature specific sentiment analysis for product reviews. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science. 2012;**7181**:457–487.

[9] Wu Y., Wei F., Liu S., Au N., Cui W., Zhou H., Qu H. OpinionSeer: interactive visualization of hotel customer feedback. IEEE Transactions on Visualization and Computer Graphics. 2016;**16**(**6**):1–10.

[10] Hu M., Liu B. Mining and Summarizing Customer Reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2004. pp. 168–177.

[11] Medvet E., Bartoli A. Brand-Related Events Detection, Classification and Summarization on Twitter. In: IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology; 2012. pp. 297–302.

[12] Jurafsky D., Martin J. H. Classification: Naive Bayes, Logistic Regression, Sentiment. Speech and Language Processing. 2015.

[13] Computer Sciences Corporation. Big Data Universe Beginning To Explode [Internet]. 2012. Available from: http://www.csc.com/insights/flxwd/78931-big_data_universe_beginning_to_explode [Accessed: 2016-09-04]