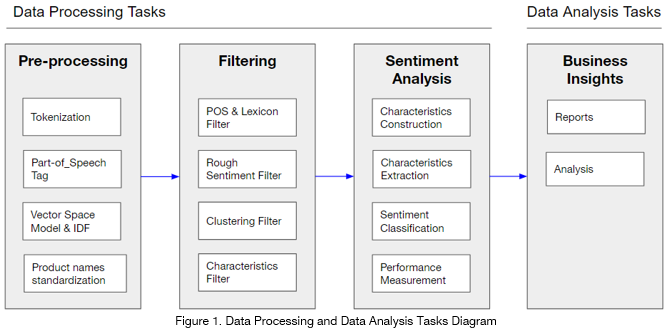
**INTRODUCTION:**

Merchants selling products through ecommerce often received a high amount of customers reviews too large in scale for human processing. These reviews often have important business insights that can be leveraged to perform actions that can improve profits. In this project we analyze ~400,000 mobile phone reviews from Amazon.com aiming to find trends and patterns to determine which product characteristics are mentioned most by customers and with what sentiment. Our task is performed in six steps: (1) pre-processing to prepare the data for analysis including tokenization and part-of-speech tagging(2)productnamesstandardization,(3)characteristics extraction, (4) reviews filtering to remove reviews considered as outliers, unbalanced or meaningless, (5) sentimentextractionforeachproductcharacteristicand(6)performanceanalysistodeterminethe accuracy of the model where we evaluate characteristic extraction separately from sentiment scores.

## METHODOLOGY:

A flowchart of the project, including the approach, performance and final business analysis is presented below:



## 1. Pre-procesing

This part includes:

**1.1 Tokenization**

Applied to both product names and reviews. It involves removal of stopwords, treating stemming of words, case-folding, removing characters that are not alphanumeric and breaking at whitespace.

**Synonyms**

Synonyms were grouped together as a means of dimensionality reduction, with manually inputted gazetteer with most common synonyms (for example the words “camera”, “video”, “display” are all transformed into “camera”).

**Negation**

It was important to handle negation for sentiment analysis so that negated opinion words could be reversed when computing its score. This method comes from Das and Chen 2001 - basically appending the suffix 'NEG' to every word appearing between a negation and a clause-level punctuation mark (such as comma). The built-in function sentiment.util.marknegation from NLTK package was used without considering double negation.

**Spelling correction** Because reviews are hand-typed the function 'spell' from the 'autocorrect package' was used to treat misspellings but also considering a manually inputted gazetteer to ignore special cases (for example the word “microsd” was incorrectly being transformed into “micros”).

### **1.2 Part of Speech tagging**

**POS tagging** was critical for three reasons.(1) To find adjectives which were all considered as opinion words (as well as others exceptions that will be discussed in next sections),

(2) to extract it’s sentiment score since words have different polarity depending on their POS tag and

(3) to extract products characteristics where Nouns (NN) and Noun-phrases (NNP) were considered as potential candidates.

The function pos\_tag from NLTK package was used for this task.

### 1.3 Vector Space Model and TF \* IDF transformation

**Vector Space Model**

A vector space model was created based on a normalized (by euclidean distance) Term-Document-Matrix via bags-of-words for both product names as well as reviews in preparation for clustering purposes. For the first to standardize product names and for the latter to filter reviews.

**Inverse Document Frequency**

Another normalized TDM was constructed this time using TF\*IDF weightings for each product name term. Its purpose was to determine which potential terms could be considered as standardized product names. The higher the IDF value the more important to be a potential part of the standardized name since the most commons words such as “unlocked”, “black” or “dual-core” should be avoided (and they have low IDF scores).

### 1.4 Product Names Standardization

Merchants often name their products in different ways, for example “iPhone 4 32GB Black, AT&T” and “iPhone 4 16GB Gold, Verizon”. This tasks objective was to add a new standard name that for this case should simply be “iPhone 4”.

Three different approaches were combined,

(1) manually inputted set/gazetteer with words to be removed,

(2) IDF importance score and

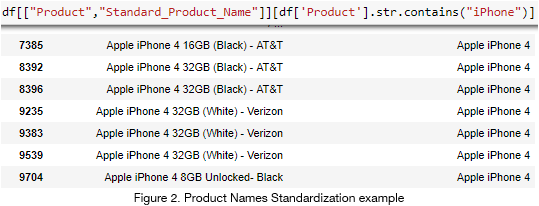
(3) Clustering.

The first step cleaned the names through the (1) gazetteer, removing colors names and common terms (such as “unlocked”).

With the remaining terms (2) selected only the first 5 terms with the highest IDF in the group (null common terms).

Finally with the remaining terms performed the (3) Clustering using a VSM matrix with k=N°\_reviews/2 clusters. This number was approximated through trial and error validating with visualization since did not know a priori how many product names were in the dataset. The number of clusters was a trade-off between having different products in the same standardized names (low number of clusters) which was highly undesirable and having too many standardized names that couldn’t standardize properly (for example having “iPhone 4 32GB” and “iPhone 4 16GB”).

A sample output can be seen in figure 2.



## 2.1 Characteristics Extraction

Two steps were taken to extract the main characteristics from reviews:

(1) manually inputted set/gazetteer with words to be removed or included and

(2) identifying NN/NNP POS tagged terms that exceeded a specific threshold (set in 1%\*N°\_reviews) of reviews occurrences.

## 2.2 Filtering

To effectively reduce the dataset size and improve performance, we need to filter out unusable, misleading and noisy reviews through 4 methods described below. In the end the dataset was reduced by 77% from an initial ~1500 reviews.

**POS Filter** Reviews without an adjective POS tag are removed since sentiment orientation is extracted only from adjectives.

**Wordnet Filter** Reviews with descriptive words not recognised by Wordnet or other sentiment lexicons are also pruned.

**Rough Sentiment Analysis Filter** To filter misleading reviews, we first conduct a rough sentiment analysis on individual opinion words, giving them a score of -1 or 1, and the overall score of a review, which is the sum of scores of all adjectives in the review. If there are less than three times the number of positive adjectives than that of negative reviews, or vice versa, then we assume the review is noisy and filter it out. Additionally, we assume that the sum of all reviews being positive, zero, or negative corresponds to a rating of >=3, 3 and below 3 respectively. Thus reviews not satisfying this equality condition against the rating are pruned.

**Clustering Filter** Performing clustering through a raw and normalized (VSM) TDM, the best results were obtained through VSM since it managed to obtain more diverse clusters - TDM was biased to create clusters based on the amount/frequency of words (clustering almost based exclusively on the lengths of the reviews).

**Characteristics Filter** The last step aims to keep only the reviews that have at least one characteristic. Since the objective of the project is determining why a product is good or bad through their characteristics sentiment instead of just computing the sentiment score of the review which can already be derived by the rating.

**2.3 Characteristics Sentiment Extraction**

This task was approached with five combined methods.

(1) Manually inputting set/gazetteer to fix wordnet sentiments that should be positive/negative instead, to ignore certain opinion words (for example “unlocked”, “old”, “normal”, “yellow”, etc.) and to include words that are not tagged as adjectives (i.e. opinion words) such as “broken”, “love” or “cool”.

(2) Inverting the polarity of words when they were negated.

(3) Using “Minqing Hu and Bing Liu” lexicon when opinion words were not supported by wordnet (either missing or neutral).

(4) Extracting the nearests opinion words (with a maximum set at 2) considering token distance (with a maximum set at 5).

(5) Computing the final characteristic sentiment score weighting by their distance (the further apart the lesser its weight).

For (4) the procedure considered looking at opinion words before and after the characteristic found and always keeping the closest ones first (for example taking the opinion word at distance -1 before the one at distance +2) where distance refers to the numbers of tokens from the characteristic word. The maximum amount of opinion words was set at two since usually when a third is found it is because the application misses a new characteristic that was there and which that third opinion word was referring to, hence avoiding assigning incorrectly an opinion word to a characteristic. Furthermore the procedure also considers if an opinion word has already been assigned to a characteristic, which is an advantage (for example avoiding to assign an opinion word twice) as well as a limitation (an opinion word between two characteristic might end up being incorrectly assigned to the first characteristic found) and should be handled in further improvements through Relationship Extraction (RE).

Another limitation and challenge of this task was the fact that customers usually give a review comparing the product with another one. This is problematic since for example they could be talking positively about the screen of the product of importance and negatively about another one they previously had, giving a neutral sentiment in the end. This was not handled in the current project and should be further investigated using RE as well.

## 3. Performance

To determine how effective the application is performing we separated the measurements in two steps.

(1) Measure effectiveness of the mobile phones characteristics extraction and

(2) over the corrected characteristics extracted, measure how effective the sentiments were recorded.

For both steps we created a manually annotated test set with ~150 reviews chosen at random. The format of the test set is as follows:

Where the third column corresponds to manually inputted results.

For step (1) we compared the characteristics extracted by the application for the reviews annotated in the test set. The measurements computed for each review were:

● True\_Positives: Correctly extracted characteristics

● True\_Negatives: All potential characteristics (NN/NNPs) that were not considered and are not in test set

● False\_Positives: Incorrectly extracted characteristics (i.e. not in the test set)

● False\_Negatives: Missed characteristics that were considered in the test set

Based on those metrics aggregated on all reviews we calculated Specificity (0.773), Recall (0.070), F1\_score (0.036) and Accuracy (0.720).

Our main focus is to have a high Recall, that is, to correctly extract characteristics which represent the main output of the business objective.

Currently it’s extremely low failing to produce relevant insights. Since in contrast Specificity is relatively high it further proves that the model is missing characteristics.

For step (2) using only the characteristics correctly extracted (Recall results) we compared their sentiment scored against those from the test set. The measurements computed for each review were:

● True\_Positives: Characteristic correctly classified with positive score

● True\_Negatives: Characteristic correctly classified with negative score

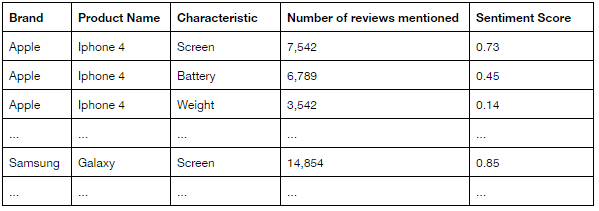
● False\_Positive: Characteristic incorrectly classified with positive score

● False\_Negatives: Characteristic incorrectly classified with negative score

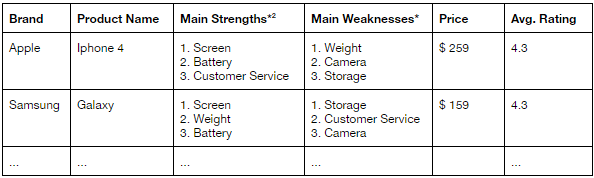
Based on those metrics aggregated on all reviews we calculated Specificity (0.8), Recall (0.666), F1\_score (0.666) and Accuracy (0.75). However, results are not statistically significant since the test set on this part was extremely low with only 7 reviews considered that had the correct characteristic extraction. nulltheless it gives insights that assigning correct sentiment scores is performing better than characteristic extraction with higher Specificity and Recall.

## 4. Business Insights

By extracting the main characteristics that customers are reviewing and which rating (i.e sentiment score) they are giving to them the business will be able to understand what positively or negatively affects product reviews and what specifically users choose as highlights or pain points. From the output table with the sentiments scores assigned to each product name characteristics and simple reporting transformation a the following table can be obtained:



Flexible enough allowing to create further reports such as:



Which can then be used by manufacturers (i.e. Apple or Samsung) to improve the quality of their products based on a specific characteristic they are getting negative reviews, and also by sellers who can use this information to diversify their products (for example have one which is strong in screen quality and another in battery) or to stop buying products that have critical issues.

## 5. Discussion

#### **5.1 Further Improvements**

Businesses do not necessarily need to have a sentiment score for reviews, especially for ecommerce sites such as Amazon where a rating is also available. For manufactures in particular even if they have a score they would not know exactly where to prioritize their efforts to improve their products. Instead, by giving them the specifics characteristics where their products are failing or not they get valuable insights to tackle problems as they arise. Hence, the challenge of correctly extracting the products characteristics is of major importance.This application underperforms in the capabilities of extracting the characteristics and seems to perform fairly well in assigning the correct sentiments to them (although some exceptions need to be adjusted through gazetteers by using the domain knowledge of the business and industry). Tofurther improve characteristics extraction an approach using topic modelling could be implemented, where assumptions are made on the probabilistic distribution of topics inside documents. An example of this would be the Latent Dirichlet Allocation that outputs word clusters. By extending the basic model of identifying topics, we can separate sentiment and features from each topic. As mentioned before, opinion word can be incorrectly assigned to characteristic when multiple characteristics are present, a task that could be tackled and improved with the usage of Name Entity Recognition (NER) and Relationship extraction (RE). Because of computational limitations we worked only on a subsample of the ~400,000 reviews. In the future using cloud computing as well as parallelization and improving the algorithm will allow to process an even larger amount of reviews. Finally to have statistically significant results a larger test set should be created with roughly at least 10% of the data (for this project only ~150 reviews were created).

#### **5.2 Conclusion**

In this project we analyzed the performance of measuring sentiment analysis on specific characteristics of mobile phones mentioned in customer reviews to provide manufacturers with actionable insights to improve their products and for sellers to improve their offerings. Results shows the worst performance on characteristic extraction where Recall is critically low. This topic is also the main challenge which could be further improved by implementing topic modelling. Sentiment scores on characteristics extraction revealed a good but not great performance suggesting that further improvements could be made using Relationship Extraction. However the test set was too small to have a clear statistical significance on the results.