# Big Data Parallel Programming Project Report
# Amazon Review Term Frequency Analysis

Andreas Häggström

April 30, 2021

# Contents

# 1 Introduction

This report covers the final project of the course *Big Data Parallel Programming, DT8034* at *Halmstad University*. In this project the student were to complete a project of their choice within the field of Big Data using the `Apache Spark framework` and a Cloud platform. The chosen topic for this report is review analysis on Amazon Customer Reviews [1] within the Wireless category. The computations for this report were performed partially on a local device aswell as on a Compute Engine Cluster on Google Cloud Platform. The purpose of this report is to extract and present general opinion of products from reviews. This through the most frequent word-pairs of either verbs, adverbs or nouns. The selection of these word classes will be reasoned in section 3. These word pairs will then be presented together with the mean sentiment and rating value of the reviews in which the word pair is present. Meta-data for the product such as the total mean sentiment value and rating will also be presented. In section 2 the data set is explained together with various information about the methods presented in 3. The final results are then presented in section 4 followed by a discussion in section 5 and a final conclusion in section 6.

# 2 Background

The technologies utilized within the calculating environment are Python 3.7.5, JDK8 and Apache Spark v2.4.5[2] together with the `Spark-NLP`[3] shell developed by John Snow Labs. The purpose utilizing this Spark-NLP is to construct pipelines described in Section 3. The final computations performed for this report will be performed within a computation cloud on Google Cloud Platform as explained in section 3.4

## 2.1 Data set

The data set used for this project is as described in section 1 customer reviews from Amazon within the Wireless category. The data set is constructed as presented in table 1.

| Column | Explanation |
|---|---|
| **marketplace** | Marketplace the product was sold in, in this case "US" for every row |
| **customer_id** | The ID of the customer making the review |
| **review_id** | The ID of the review |
| **product_id** | The ID of the product |
| **product_parent** | Random identifier for product |
| **product_title** | Title of the product |
| **product_category** | Broad product category, in this case "Wireless" for every row |
| **star_rating** | 1-5 star rating of the product |
| **helpful_votes** | Amount of helpful votes |
| **total_votes** | Amount of votes |
| **vine** | Review written as part of the Vine program |
| **verified_purchase** | Indicates if the review is on a verified purchase |
| **review_headline** | The title of the review |
| **review_body** | The review text |
| **review_date** | The date the review was written |

Table 1: Data set structure

Worth remarking regarding the data set structure is that not all columns can be deemed worth for processing. The columns chosen are `review_id`, `product_id`, `star_rating`,`verified_purchase` and `review_body`. The columns not chosen were dropped for various reasons, `marketplace` and `product_category` were dropped since they are the same for every row in the data set. The `vine` column, describing if the review was part of the vine program in which Amazon invites trusted reviewers to post their opinions [4] was removed from the review analysis since only 0.1942% of the 9 million entries were a part of it. The product rating distribution of this column will though be presented in section 4.3 for future discussion. The remaining columns not chosen were dropped since they either were redundant, didn't hold enough informational value or because of project limitations.

## 2.2 Related Work

The subject of this project is review analysis, methods similar to the ones described in section 3 can be found in [5],[6] and [7]. In the work of *Yatani et. al*[5], a restaurant review platform was developed, in which the restaurant rating was displayed together with frequent word pairs. This was performed through extraction of **adjective-noun** word pairs followed by calculating the frequency of the extracted pairs together with their sentimental values. These sentimental values were then used to adjust the font color of the word pairs. In *Zhang et. al*[6] the method was instead to extract word pairs by various patterns e.g. **noun-preposition-noun**, **noun-concept** and **concept-verb-noun**. *Zang et. al* also discussed the concept of including the word **"no"** since it makes a major difference in the review analysis. For instance, if a customer review contains the phrase **"I had no problems"**, the sentimental value of the phrase becomes dishonest if the word **"no"** is removed. It also covers an advanced feature extraction method through Double Propagation to extract relevant features. The work of *Kim and Hovry*[7] covers a more advanced method for the purpose of identifying and analysing opinions. Here not only the opinions were identified but also the opinion *holder* and the *topic* of the opinions. The report also covers a method for identification of synonyms of words using WordNet.

# 3 Method

## 3.1 Pre-processing

The method of extracting the most frequent word pairs together with the rating and sentiment data of a product is performed in several steps. The first part was to filter out all products except the 100 products with the most reviews. This shrank the active data set from 9 million to 0.45 million entries. This was done because of limitations and also since working with products having many reviews result in more certain presentations. The next step was to drop the columns deemed uninteresting as explained in Section 2.1. This was followed by filtering out all reviews without a verified purchase for the purpose of ensuring data quality. Then various cleaning was performed such as turning all reviews into lower case and removing chars that weren't normal letters.
For the last part of the pre processing, a pipeline was constructed using `Spark-NLP`. This pipeline performed Lemmatization, removed stop words cleaning and performed Part of Speech tagging. Lemmatization is the process of turning the words of the review into the corresponding base dictionary words, this to reduce the risk of missing patterns due to inflections [8]. Stop words are commonly used words, such as "a", "an" and "the", these words contribute with little or no information of review opinion. Removing stop words also reduces the amount of words the computer has to work with greatly since these are the most commonly used words [9]. Part of Speech tagging marks the words of the review with their corresponding word part of speech tag [10]. This can thereafter be utilized to extract words that are of specific word classes as performed by [5], [6] and [7].

## 3.2 Calculating review sentiment, performing Part-Of-Speech-tagging and extracting N-grams

### 3.2.1 Sentimental Analysis

The sentimental value of a review represents the general tone of the review text [11]. It allows for classification for a review as either negative or positive. This report utilizes a Sentiment Intensity Analyser developed by Natural Language ToolKit, `NLTK` [12]. This sentiment analyser generates four sentimental scores for the review as listed below.

- Positive factor
- Negative factor
- Neutral factor
- Compound factor

Where the **Compound factor** is the normalized overall score constructed from the first three factors through the formula presented in equation 1.

$$Compound = \frac{\sum_i Sentiment(w_i)}{\sqrt{(\sum_i Sentiment(w_i))^2 + \alpha}} \tag{1}$$

Where $Sentiment(w_i)$ is the valence of the word $w_i$ and $\alpha$ is a normalization coefficient.

### 3.2.2 POS-Tagging

After the sentimental value of the reviews are calculated, a filter is performed to remove words that are not of the word class **noun**,**verb** or **adverb**. This is somewhat in contrary to the work of *Yatani et. al*[5] where **adjectives** and **nouns** were extracted. For the purpose of analysing reviews for restaurants as performed by *Yatani et. al*, the combination of **nouns** and **adjectives** are logical, since sentences such as **"burnt steak"** or **"good soup"** are relevant in this case. When analysing reviews for a specific product, the main nouns are already known since these are the product's. This makes the **noun-adjective** pattern mostly irrelevant since the information gained from phrases such as **"good phone"** would reasonably be equivalent to the amount of stars given in the review. This report therefore uses sentence patterns more like the one presented by *Zang et. al*[6], where product reviews are analysed. Figure 1 shows a sample sentence from the data set together with the difference in information gained by extracting the different sets of word classes. Here the words colored in blue are the extracted words and the ones overlined are excluded. Here it's clear that no information is gained from the review by extracting nouns and adjectives but when extracting verbs, adverbs and nouns the word pair **"battery drained"** can be gained.

**Original Sentence:**
Once I turned on the circle apps and installed this case, my battery drained twice as fast as usual.

**Sentence with Nouns and Adjectives marked:**
~~Once I turned on the~~ circle apps ~~and installed this~~ case, ~~my~~ battery ~~drained~~ twice ~~as~~ fast ~~as~~ usual.

**Sentence with Verbs, Adverbs and Nouns marked:**
~~Once I~~ turned ~~on the~~ circle apps ~~and~~ installed ~~this~~ case, ~~my~~ battery drained twice ~~as~~ fast ~~as usual~~.

Figure 1: Sample sentence with marked word classes

As explained in section 2.2, the word **"no"** gives increased information gain when not excluded. This is shown in figure 2 where it's been overlined and excluded. To overcome this, an exception was made for this specific word to keep it for further processing.

**Original Sentence:**
This case worked just fine, and using their yootech wireless charging stand, I had no problems charging my phone in less than 2 hours.

**Sentence with Nouns and Adjectives marked:**
~~This~~ case ~~worked just~~ fine, ~~and using their~~ yootech wireless ~~charging~~ stand, ~~I had no~~ problems ~~charging my~~ phone ~~in~~ less ~~than 2~~ hours.

**Sentence with Verbs, Adverbs and Nouns marked:**
~~This~~ case worked ~~just~~ fine, ~~and using their~~ yootech wireless charging stand, ~~I had no~~ problems charging ~~my~~ phone ~~in less than 2~~ hours.

Figure 2: Sample sentence with marked word classes

### 3.2.3 N-gram construction

With the words of the desired word classes extracted, `Apache Spark's` Machine Learning library was used to construct word pairs of two. For the sample sentence shown in figure 1, the resulting N-grams can be seen in figure 3 with the exception of them not being lemmatized. Here one can see noise in form of non informative N-grams are also present, this noise can be mostly ignored through filtering on N-gram frequency as in the work of *Yatani et. al.*

**Extracted Words:**
[turned, circle, apps, installed, case, battery, drained, twice, fast]

**N-Grams constructed:**
[turned circle, circle apps, apps installed, installed case, case battery, battery drained, drained twice, twice fast]

Figure 3: Extracted words and the constructed N-grams

## 3.3 Word pair analysis

After the N-grams were constructed for each review. The importance of each word pair was calculated using Term Frequency - Inverse Document Frequency,**TF-IDF** [13] as shown in equation 2. Additionally, this increases the tolerance against noise shown in figure 3

$$\begin{aligned} TFIDF(p, r, R) &= TF(p, r) \cdot IDF(p, R), \\ IDF(p, R) &= \log \frac{|R|}{DF(p, R)}, \end{aligned} \tag{2}$$

Where $p$ represents a word pair of a review $r$ within the total set of reviews $R$ for a product. $TF(p, r)$ represents number of occurrences of the word pair $p$ within review $p$ and $DF(p, R)$ equals the amount of reviews $r$ within $R$ which contains word pair $p$. This **TF-IDF** value is then normalized for each product for readability.

Finally, every word pair was assigned the mean sentiment value and star rating of the reviews where it is present. The reason to not use the sentimental value for the word pair itself is that one would thereby loose the context of the review.

## 3.4 Google Cloud Platform

The computations explained above will be performed on a Dataproc cluster consisting of three Compute engine VM-nodes where one is a master node and the remaining two are working nodes. The master node is of machine type `n1-standard-8 (8 vCPUs, 30GB memory)` and the worker nodes are of type `n1-standard-4 (4 vCPUs, 15 GB memory)`. These three cluster nodes run on the same image version of `1.4.27-debian9`.

# 4 Results

## 4.1 Product specific presentation

The chosen method of displaying the result of the review analysis is through histograms together with the normal distributions for review sentiment and rating. Here the dotted line represent the mean value of the rating/sentiment. In addition to this a scatter plot of the 15 word pairs with the highest **TF-IDF** score is shown with the axes describing the mean value of the review sentiment and rating of which the word pairs are present. The word pairs are also colorized with a high color intensity representing a high **TF-IDF** score.

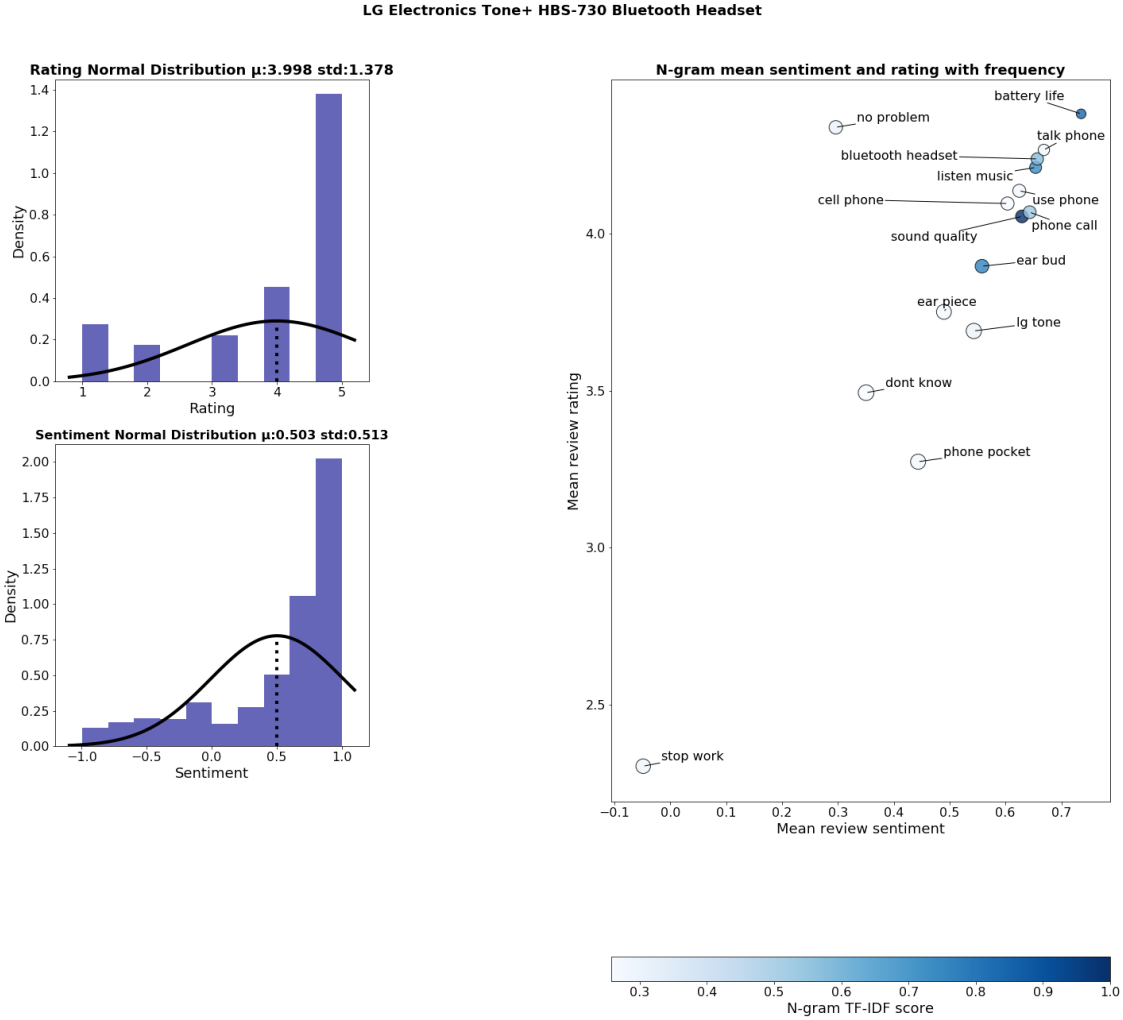


Figure 4: Resulting plot for a Bluetooth headset

For the same product, the 15 word pairs are also described in table 2.

| Word Pair | TF-IDF | Mean Sentiment | Mean Rating |
|---|---|---|---|
| sound quality | 1.0 | 0.6296 | 4.0555 |
| battery life | 0.847 | 0.7355 | 4.3823 |
| ear bud | 0.7662 | 0.558 | 3.8969 |
| listen music | 0.7411 | 0.654 | 4.2115 |
| bluetooth headset | 0.5866 | 0.657 | 4.2387 |
| phone call | 0.541 | 0.6434 | 4.069 |
| no problem | 0.3023 | 0.2964 | 4.3394 |
| lg tone | 0.2997 | 0.5434 | 3.6905 |
| stop work | 0.2889 | -0.0483 | 2.3037 |
| talk phone | 0.2817 | 0.6688 | 4.2674 |
| phone pocket | 0.2755 | 0.4437 | 3.2739 |
| use phone | 0.272 | 0.6248 | 4.1371 |
| ear piece | 0.2649 | 0.4899 | 3.7514 |
| dont know | 0.2589 | 0.3506 | 3.4938 |
| cell phone | 0.2573 | 0.6037 | 4.0966 |

Table 2: Word pair data for a Bluetooth headset

In figure 5, the resulting plot of a product with a lower rating is shown. The tabular of the word pairs are shown can be seen in 3.

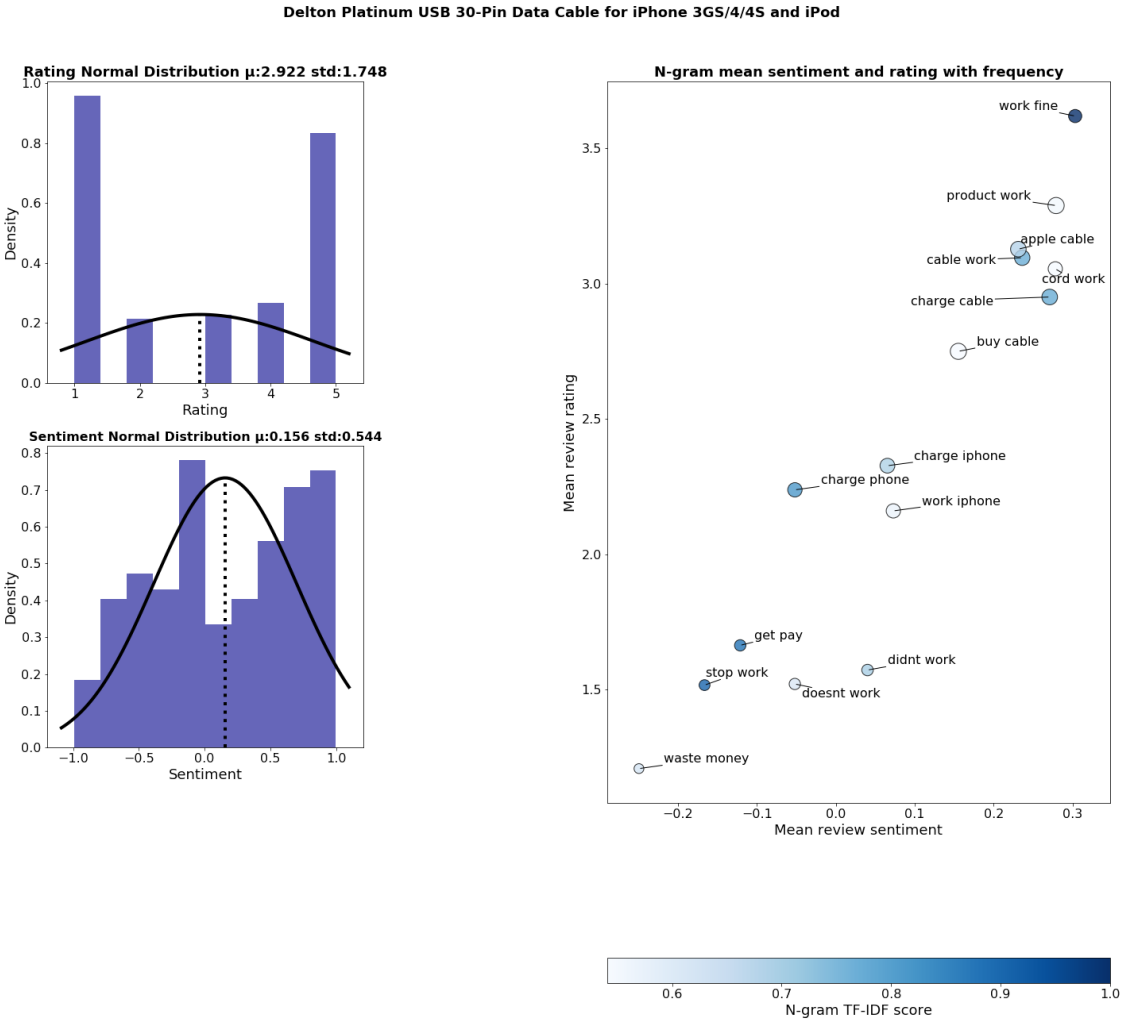


Figure 5: Resulting plot for a Data cable

| Word Pair | TF-IDF | Mean Sentiment | Mean Rating |
|---|---|---|---|
| work fine | 1.0 | 0.3032 | 3.6194 |
| stop work | 0.904 | -0.1664 | 1.5168 |
| get pay | 0.8801 | -0.1212 | 1.6637 |
| charge phone | 0.8131 | -0.0519 | 2.2383 |
| charge cable | 0.7731 | 0.2709 | 2.9508 |
| cable work | 0.7685 | 0.2361 | 3.0959 |
| didnt work | 0.6981 | 0.0401 | 1.5725 |
| charge iphone | 0.6873 | 0.0653 | 2.3275 |
| apple cable | 0.6777 | 0.2312 | 3.1277 |
| waste money | 0.6199 | -0.2495 | 1.2083 |
| doesnt work | 0.6078 | -0.052 | 1.521 |
| work iphone | 0.5689 | 0.0728 | 2.1605 |
| product work | 0.5516 | 0.279 | 3.2887 |
| cord work | 0.5435 | 0.278 | 3.0549 |
| buy cable | 0.5406 | 0.1552 | 2.75 |

Table 3: Word pair data for a Data cable

Additional tables presenting word pair data for products can be seen in the appendix, section 7.

## 4.2 Data set distribution

A plot for the total distribution of the rating of the whole data set can be seen in figure 6 followed by an equal plot representing the distribution of the sentiment values in figure 7.

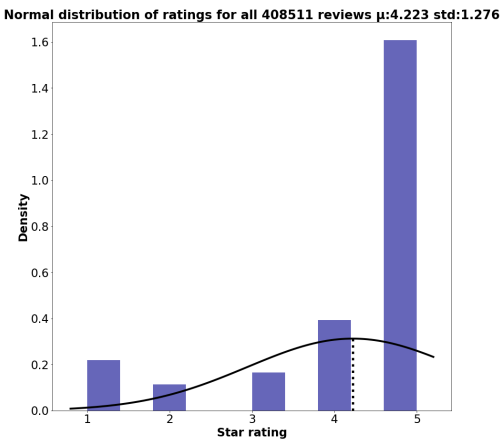


Figure 6: Histogram and normal distribution of review ratings

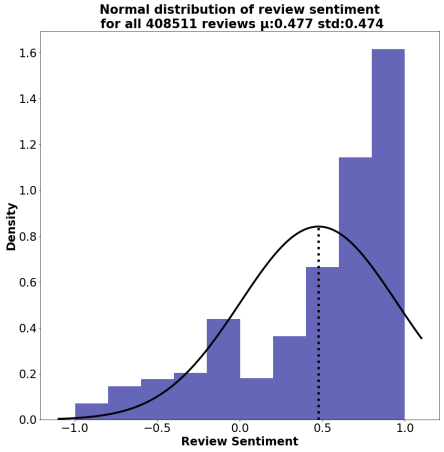


Figure 7: Histogram and normal distribution of review sentiments

## 4.3 Vine program Rating distribution

Finally, as mentioned in section 2, the `vine` column states whether the reviews was written as part of the `Vine program`. The distribution of the rating for these reviews is shown in figure 8.

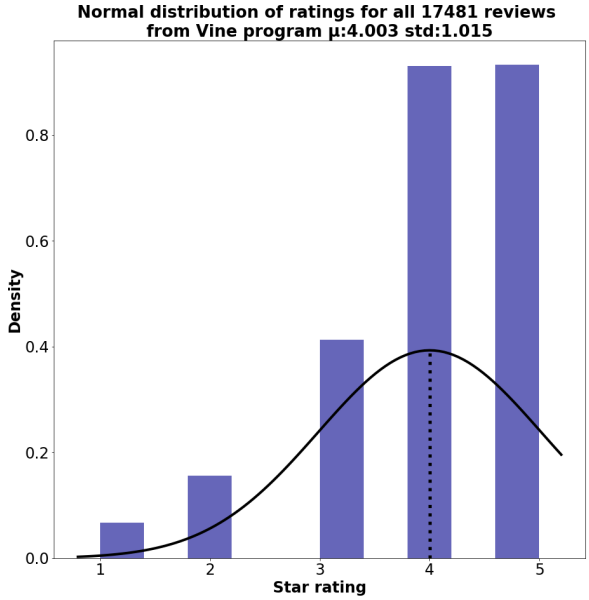


Figure 8: Vine review rating distribution

# 5 Discussion

As seen in the plots and tables in sections 4.1 and 7, noise is still present within the extracted word pairs. This shows the need of more complex POS-tag filtering to remove phrases such as **"bluetooth headset"**, **"ear piece"** and **"dont know"** in figure 4. This could be implemented like the methods of either *Zang et. al* or *Kim* and *Hovry* as explained in section 2.2. The results of the sentiment analysis shown in section 4 were promising since they followed the rating of the product well. This can be compared to the discussion in *Yatani et. al* where their method of analysing the sentiment of each word pair independently of the sentence in which they were present. *Yatani et. al* described how this approach looses the context of the word pairs though the example **"This is not a good restaurant"** where the word pair **"good restaurant"** would be extracted and give an inaccurate result. The method implemented for this report instead assigns a word pair the sentiment of the entire sentence of which it is present as explained in section 3.3. The resulting plots of this report do however display a informational overall representation of the customer review opinions for products.

The Amazon review data set used a rating scale of one to five stars. As explained by *Hu et. al* [14], collection of product reviews from recent buyers are shown to be overwhelmingly positive. With the review distribution following a `"J-shaped"` curve. This can be seen in this data set for both the ratings and review sentiments as shown in figures 6 and 7. *Hu et. al* describes how this is caused partially by customers being very satisfied/unsatisfied are more likely to review a product. Resulting in values between two and four underrepresented within product review ratings. One method presented by *Hu et. al* to overcome these problems within product review analysis are to rely not only on the mean value of product rating but also the standard deviation. Additionally, the analysis should also take the product price in consideration to overcome purchasing bias. Finally, to overcome the under reporting of product reviews a bimodal distribution of the ratings should be generated.

In section 4.3, figure 8 shows the distribution of the review ratings performed as part of the **Vine program**. This distribution can be compared to the one of the the ratings for the 100 most reviewed products shown in figure 6. Here one can see that in the `vine` distribution, the `J-curve` is not as present if at all. What is most notable is the fact that the density of the rating value 4 is equal to the one of 5. This accompanied with the fact that the density for the value of 3 fairly high might indicate that the `vine` ratings are more reliable even though both distributions have similar mean values.

# 6 Conclusion

This report explained a method of analysing Amazon customer review data and presented the results of these methods. The imperfections of the results were like the ones of reports presented in section 2.2. These were a lack och dealing with uneven distribution of review ratings, inadequate grammar in customer reviews and noise filtering. Despite these shortcomings the methods resulted in what seems to be an informative presentation of the general opinion of products. The fact that the sentimental distribution shown in 7 is similar to the rating distribution in 6 indicates that the method of using the whole sentence when analysing sentiment for word pairs promising and that the `J-curve` presented by *Hu et. al* applies to sentiment as well. One aspect to consider about this report is also that, since the data chosen was the collected reviews of the 100 most reviewed products, the reportage bias might be even greater.

The improvement fields for the performed methods would therefore be the imperfections presented previously in this section. Future work could also include practicing supervised learning where the reviews from the `Vine program` can be seen as the true data. This to learn how to find patterns between the standard reviews from customers and the ones from the invited trusted reviewers in the `Vine program`.

# 7 Appendix

| Word Pair | TF-IDF | $\mu$ Sentiment | $\mu$ Rating |
|---|---|---|---|
| charge ipad | 1.0 | 0.3208 | 4.0095 |
| charge device | 0.8591 | 0.332 | 4.2657 |
| charge iphone | 0.8221 | 0.3204 | 4.1855 |
| wall charger | 0.7531 | 0.4289 | 4.4907 |
| charge phone | 0.6667 | 0.4204 | 4.2991 |
| iphone ipad | 0.6532 | 0.364 | 4.2061 |
| ipad iphone | 0.6308 | 0.3924 | 4.3396 |
| charger work | 0.6153 | 0.4922 | 4.4667 |
| apple product | 0.5794 | 0.3591 | 4.1583 |
| work charge | 0.5744 | 0.1363 | 3.5663 |
| work fine | 0.5678 | 0.2865 | 3.9727 |
| use charge | 0.5423 | 0.3744 | 4.4044 |
| port charge | 0.5183 | 0.322 | 4.1148 |
| charger charge | 0.5152 | 0.2828 | 4.2391 |
| no problem | 0.5076 | -0.0444 | 4.764 |

Table 4: Word pair data of a portable charger for Android and Iphone

| Word Pair | TF-IDF | $\mu$ Sentiment | $\mu$ Rating |
|---|---|---|---|
| dash cam | 1.0 | 0.3322 | 3.1804 |
| sd card | 0.9629 | 0.29 | 3.0583 |
| video quality | 0.6189 | 0.2605 | 3.1141 |
| suction cup | 0.4777 | 0.3931 | 3.375 |
| night vision | 0.4587 | 0.2857 | 3.0865 |
| memory card | 0.4513 | 0.3728 | 3.4468 |
| get pay | 0.421 | 0.0406 | 2.0722 |
| work fine | 0.4057 | 0.3416 | 3.3627 |
| license plate | 0.399 | 0.1909 | 3.3411 |
| camera work | 0.3595 | 0.3879 | 3.0351 |
| stop work | 0.3581 | -0.1483 | 1.7013 |
| record video | 0.3494 | 0.1417 | 2.8462 |
| waste money | 0.3082 | -0.182 | 1.1094 |
| quality video | 0.3045 | 0.3817 | 3.1607 |
| date time | 0.3011 | 0.3537 | 3.0851 |

Table 5: Word pair data for Video recorder

| Word Pair | TF-IDF | $\mu$ Sentiment | $\mu$ Rating |
|---|---|---|---|
| lifeproof case | 1.0 | 0.2367 | 2.6306 |
| phone case | 0.5898 | 0.2931 | 3.0563 |
| life proof | 0.5852 | 0.0541 | 2.3946 |
| buy case | 0.4125 | 0.2361 | 2.4486 |
| water proof | 0.3982 | 0.2415 | 3.1062 |
| drop phone | 0.3786 | 0.386 | 3.88 |
| screen protector | 0.3722 | 0.1907 | 2.8293 |
| proof case | 0.3691 | 0.234 | 2.7093 |
| case phone | 0.3571 | 0.3267 | 2.9875 |
| case case | 0.3368 | 0.2132 | 2.8462 |
| use phone | 0.3346 | 0.1687 | 2.8372 |
| get case | 0.3195 | 0.2192 | 2.7708 |
| headphone jack | 0.3135 | 0.2066 | 3.1538 |
| protect phone | 0.3097 | 0.4856 | 3.5935 |
| case iphone | 0.3078 | 0.5419 | 3.4865 |

Table 6: Word pair data for a Iphone 4 Lifeproof case

| Word Pair | TF-IDF | $\mu$ Sentiment | $\mu$ Rating |
|---|---|---|---|
| charge phone | 1.0 | 0.4696 | 4.3545 |
| charge iphone | 0.5911 | 0.4466 | 4.5655 |
| phone charge | 0.4555 | 0.4334 | 4.4216 |
| get charge | 0.4298 | 0.5414 | 4.4506 |
| customer service | 0.3869 | 0.698 | 4.6035 |
| astro mini | 0.3673 | 0.52 | 4.4167 |
| charge charge | 0.3543 | 0.3742 | 4.1875 |
| work charge | 0.3332 | 0.2651 | 3.7222 |
| phone battery | 0.3286 | 0.5565 | 4.6468 |
| use charge | 0.3239 | 0.4246 | 4.4623 |
| use phone | 0.3195 | 0.5337 | 4.7387 |
| hold charge | 0.3182 | 0.4112 | 3.8155 |
| charge device | 0.3118 | 0.4647 | 4.2586 |
| cell phone | 0.2918 | 0.5261 | 4.5943 |
| fit pocket | 0.2715 | 0.6221 | 4.7661 |

Table 7: Word pair data of a Power bank

# References


[1] Amazon. Amazon Customer Review Data;. Accessed: 2020-05-14 URL: https://registry.opendata.aws/amazon-reviews/.

[2] Foundation AS. Apache Spark;. Accessed: 2020-05-14 URL: https://spark.apache.org/.

[3] Labs JS. Spark NLP: State of the Art Natural Language Processing;. Accessed: 2020-05-14 URL: https://nlp.johnsnowlabs.com/.

[4] Amazon. Amazon Vine Program;. Accessed: 2020-05-14 URL: https://www.amazon.com/gp/vine/help.

[5] Yatani K, Novati M, Trusty A, Truong KN. Review Spotlight: A User Interface for Summarizing User-Generated Reviews Using Adjective-Noun Word Pairs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '11. New York, NY, USA: Association for Computing Machinery; 2011. p. 1541–1550. Available from: https://doi.org/10.1145/1978942.1979167.

[6] Zhang L, Liu B, Lim SH, O'Brien-Strain E. Extracting and ranking product features in opinion documents. In: Proceedings of the 23rd international conference on computational linguistics: Posters. Association for Computational Linguistics; 2010. p. 1462–1470.

[7] Kim SM, Hovy E. Identifying and analyzing judgment opinions. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics; 2006. p. 200–207.

[8] Christopher D Manning PR, Schütze H. An Introduction to Information Retrieval. Cambridge University Press; 2009. p. 32–34. Available from: https://www.informationretrieval.org/.

[9] Christopher D Manning PR, Schütze H. An Introduction to Information Retrieval. Cambridge University Press; 2009. p. 27–28. Available from: https://www.informationretrieval.org/.

[10] Cutting D, Kupiec J, Pedersen J, Sibun P. A practical part-of-speech tagger. In: Third Conference on Applied Natural Language Processing; 1992. p. 133–140.

[11] Christopher D Manning PR, Schütze H. An Introduction to Information Retrieval. Cambridge University Press; 2009. p. 254. Available from: https://www.informationretrieval.org/.

[12] Bonaccorso G. Machine Learning Algorithms: Popular algorithms for data science and machine learning. Packt Publishing Ltd; 2018.

[13] Christopher D Manning PR, Schütze H. An Introduction to Information Retrieval. Cambridge University Press; 2009. p. 118–119. Available from: https://www.informationretrieval.org/.

[14] Hu N, Zhang J, Pavlou PA. Overcoming the J-shaped distribution of product reviews. Communications of the ACM. 2009;52(10):144–147.