university of groningen

faculty of science and engineering

Gaining Insights from EV Charging Reviews Using Natural Language Processing

Hector Quaicoe

university of groningen

faculty of science
and engineering

University of Groningen

Gaining Insights from EV Charging Reviews Using Natural Language Processing

Bachelor's Thesis

Hector Quaicoe (s4079183)

Primary Supervisor : V. (Viktoriya) Degeler PhD

Secondary Supervisor : M.A. (Andrés) Tello Guerrero, MSc

External Supervisor : Ding Luo PhD

July 7, 2022

Contents

Acknowledgments

The completion of this project could not have been possible without the expertise of Dr. Viktoriya Degeler, my thesis supervisor. I would also like to thank my amazing friends, Kwabena Darkwa and Chelsea Azumah, for taking their time to proofread my thesis and provide feedback where needed thoroughly.

A debt of gratitude is also owed to Dr. Ding Luo for setting up this collaboration thesis with Shell. This opportunity has helped expand my knowledge of NLP and opened doors to possible career advancement.

Last but not least, I would like to thank my parents, Mr. and Mrs Quaicoe; without their support and prayers, none of this would indeed be possible

Abstract

Customer satisfaction plays an imperative role in the business' success. One way to measure customer satisfaction level is by utilizing customer reviews. This thesis analyzes customer reviews of EV charging stations owned by Shell using a text mining approach, including sentiment analysis and topic modeling. Vader Lexicon is the classification method utilized to aggregate positive or negative sentiments in each review. Moreover, Latent Dirichlet Allocation is used to cluster reviews into various topics. In addition, a Support vector machine classifier is used to identify positive or negative sentiments in the review sentence to compare the unsupervised and supervised approaches taken. The classification results show that the supervised technique performs better at classifying sentiments. Ultimately, this thesis aims to help Shell use their data efficiently by improving the quality of its EV charging solutions and staying ahead of its competitors.

1 Introduction

User feedback or reviews are one of the ways businesses monitor the performance of their products or services. Through these mediums, companies can gain a general overview of a product's popularity among their customers. Customer satisfaction is an opinion or feeling between expectation and reality obtained by consumers [1]. Reasonable customer satisfaction affects the profitability of nearly every business. For example, when customers perceive a good product/service, each will typically tell nine to ten people [1].

Shell is one of the world's leading energy providers. Shell operates over 80,000 charge points for electric cars at homes, businesses, Shell retail sites, and destinations. In addition, Shell presently offers access to over 300,000 additional charge points through its roaming networks. Therefore, to ensure customers worldwide are satisfied with Shell's products, Shell has identified sentiment analysis as one of the ways to improve customer satisfaction.

Sentiment analysis involves looking at a text and classifying the text into a positive, neutral, or negative sentiment. We can clean the text by removing stopwords, punctuations, and numbers. Moreover, by using the lexical-based approach, we can classify the sentiments in unlabelled data accordingly. This research aims to gain insights from user feedback using sentiment analysis and use these results to understand customers' sentiments accurately. On top of that, after classifying each review in our data, we will look to cluster the various topics within the reviews. Therefore, incorporating topic modeling will allow us to gain additional insights into customers' sentiment and better understand it.

1.1 Research Questions

To summarize, this thesis focuses on the following problems:

Q1. **What insights can we retrieve from customer feedback using sentiment analysis?**

Q2. **Can we understand whether customer's opinion is positive or negative using sentiment analysis?**

2 Background Literature

Different techniques exist that can be used to classify customer reviews' sentiment polarity (pos, neu, neg). These include Sentiment Analysis, Topic Modeling, Text Generation, etc. All these techniques fall under the umbrella term Natural Language Processing also known as NLP.

2.1 General Overview of NLP

Natural language processing is the process of dealing with or extracting data from text. As mentioned above, several techniques exist, such as sentiment analysis, topic modeling, text classification, lemmatization & stemming. However, for this project, we shall concentrate our efforts on sentiment analysis, also known as opinion mining and topic modeling.

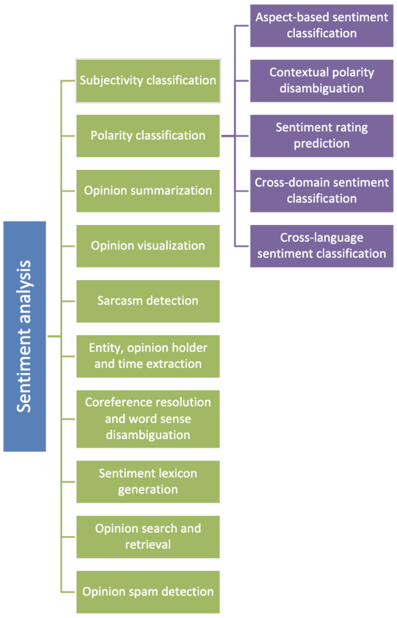

Sentiment analysis
Subjectivity classification
Polarity classification
Opinion summarization
Opinion visualization
Sarcasm detection
Entity, opinion holder and time extraction
Coreference resolution and word sense disambiguation
Sentiment lexicon generation
Opinion search and retrieval
Opinion spam detection
Aspect-based sentiment classification
Contextual polarity disambiguation
Sentiment rating prediction
Cross-domain sentiment classification
Cross-language sentiment classification

Figure 1: Taxonomy of common tasks in SA

Sentiment analysis aims to define automatic tools able to extract subjective information from texts in natural languages, such as opinions and sentiments, to create structured and actionable knowledge to be used by either a decision support system or a decision maker [2]. Sentiment analysis has numerous world applications, and for this research, we will concentrate on its use in customer or user feedback to drive business metrics. Moreover, sentiment analysis on customer feedback is an application rarely used, and it is a growing field. Therefore, it is prone to be misused for certain tasks. Sentiment analysis is often improperly used when referring to polarity classification, which instead is a subtask aimed at extracting positive, negative, or neutral sentiments (also called polarities) from texts [2]. Although an opinion could also have a neutral polarity (eg, "I don't know if I liked the movie or not. I should watch it quietly."), most work in sentiment analysis usually assumes only positive and negative sentiments for simplicity [2]. Fig.1 illustrates some common use cases of sentiment analysis

An important task in sentiment analysis is to cluster customer reviews into topics to capture the various sentiments shared among customers. In this research, we will aim to summarize the different topics within the data through Topic Modeling. Topic modeling is a method that draws from a large number of documentation, possibly useful topics based on a process probability distribution model [3]. Topic modeling approaches can be divided into two categories: the probabilistic model, including the LDA model [4], and the non-probabilistic model, including Non-negative Matrix Factorization [5]. Probabilistic models are more popular, assuming that there exist latent spaces [6] between related parameter systems. The probabilistic model generally produces better results at the cost of stability [7]. The two categories of approaches have a series of similar factors. For example, input parameters should include the topic number and keyword vectors [8]. Recently, a stream of research has contributed to mining the topics of customers from online reviews based on the LDA model [4]. Tirunillai and Tellis [9] presented an architecture combining a new, improved LDA for mining topics from online reviews. The preprocessed data are fed into the model, and then the label and heterogeneity of each topic can be identified through the model's outputs.

2.2 Business Impact of Sentiment Analysis

Businesses are continuously trying to find different avenues to increase their revenues. One of the ways to achieve that is through customer satisfaction and online reviews of products. First, online review ratings have been controversial for their objectivity. Consumers who are most likely to leave a product review are either the ones who are extremely satisfied or the ones who are extremely dissatisfied [10]. Thus, the average online review ratings are prone to extremity bias as extreme values like the highest or the lowest ratings prevail in the distribution of online review ratings. Unsurprisingly, the distribution of online review ratings for innovative products often does not feature a normal distribution [11]. Sentiment analysis is an alternative option for analyzing customer sentiment rather than relying on online reviews. Text sentiment metrics are more straightforward for their values to indicate sentiment in customer opinions. Sentiment analysis compares the number of positive leaning or negative leaning (or both) words to calculate overall positivity in the text by subtracting the number of negative leaning words from positive leaning words [11]. There are some limitations with these methods, such as the accuracy of lexicon dictionaries used to classify sentiments and the uncertainty that some sentences might express stronger sentiments with fewer words [12].

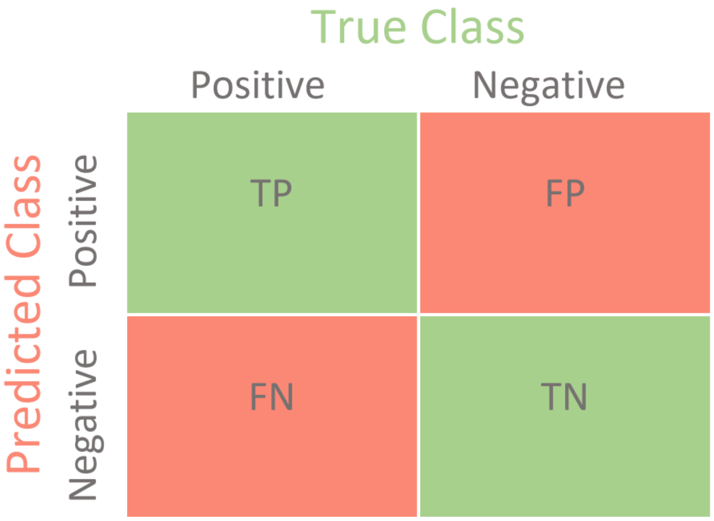
2.3 Sentiment Classification

2.3.1 Machine Learning Approach

Sentiment classification is a unique text classification technique that aims to classify a text according to the sentimental polarities of opinion it contains [13]. This paper will experiment with the algorithm Support Vector Machine (SVM) for sentiment classification. SVM is a regression and classification method for analyzing, recognizing data patterns and making predictions that require pre-classified documents as training data. Classification performed by SVM is to find a hyperplane that separates positive class data from negative class data by maximizing the distance between positive class data and the closest negative class data [14]. SVM has a high dimensional input space, so SVM is suitable for large amounts of data.

Methods of precision, recall, and accuracy is used to check the accuracy of the results of the process [15]. A confusion matrix is created to provide performance classification data. Elements of the confusion matrix in Fig.2 are True positive (TP) when both human and method predict are positive and True Negative (TN) when both human and method predict are negative. False-negative (FN) is used when the human prediction is positive while the method prediction is negative, and False Positive (FP) is used when the human prediction is negative while the method prediction is positive [16].

The level of accuracy between what the user wants and the results of the system process is called Precision, which can be seen in Equation 1. In contrast, Recall is the system's average success in finding information, listed in Equation 2. Precision and Recall calculations are used to avoid measurement errors for deviation values, as shown in Equation 3. Accuracy is the degree of truth between the predictive value and the actual value shown in Equation 5. Precision is the ratio between true positives and all the positives, and since this research analyzes customer reviews, the positive results are the reviews classified as positive reviews. The Recall value is obtained by dividing TP by positive results based on the real data. The F-Score value uses the Recall value and Precision value. The accuracy is the division of the amount of TP and TN to the amount of data [15].



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 2: Confusion matrix of a two class problem

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$F-Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{3}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

2.3.2 Lexicon Based Approach

A primary task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence, or an entity feature/aspect is positive, negative, or neutral [17]. When performing sentiment analysis on product reviews, it is best to do phrase-level sentiment analysis because most reviews tend to contain positive words to describe a negative feeling. Consequently, when we analyze each word, the polarity assigned to the phrase, and the context in which it is being used, we can retrieve the overall polarity of the review. Turney presented an unsupervised algorithm for the classification of reviews into two classes: recommended or not recommended [18]. He gave phrase extraction patterns; then, the semantic orientation of a phrase is computed using the PMI-IR algorithm. PMI-IR is Pointwise Mutual Information (PMI) and Information Retrieval (IR), measuring the similarity of pairs of words or phrases. Reference Word Pairs are used for predicting the sentiments of phrases, and the average semantic orientation of the review is used to classify the review [19].

3 Methodology

3.1 Data

We will be using data provided by Shell's E-Mobility analytics group. Moreover, since the data is proprietary, any sensitive information will not be shared. The data is a set of charge station reviews collected from the end of December 2021 to January 2022. Each feedback includes the following information: 1) Network, 2) Name, 3) Postcode, 4) LocNum, 5) DeviceNum, 5) ChargeDeviceRef, 6) Created, 7) StatusID, 8) Handle, 9) Status, 10) Comment.

Moreover, we shall briefly discuss three columns to give some context to what they mean. Throughout the thesis, we will mention them, so it is essential to understand what they mean.

- Network : Contains the value of different charging operators. In the data, the two distinct values are Ubitricity and Shell Recharge
- Status : Contains five distinct values Successful charge, Issue Reported, Comment, Reply and Device working. Before users input their review, they choose one of these values to express their experience during charging sessions
- Comment : This is the review text. Moreover, in the thesis, we will use comment or review to describe this column interchangeably, referring to the same thing.

3.2 Preprocessing & Analysis

Our data must be cleaned before performing any explanatory data analysis. Some standard ways to clean our data are trimming all the texts to lowercase and removing punctuation, numbers, and stopwords. Stopword removal eliminates words that often appear but do not have meaning in languages, such as “the”, “a”, “an”, “in” [15]. Additionally, since we are performing sentiment analysis, it is important not to remove negating stopwords such as “no” and “not” as they can alter the polarity of a given review. Also, we performed lemmatization on the text. That way, we keep the lemma of each word. The choice to rather lemmatize instead of stem each word in a review is because it gives an accurate meaning of the word for the given context. Generally, stemming might reduce the word to an incomprehensible form. In the lemmatization process, each word is given the appropriate POS tag and then tokenized. Therefore, each token will be a tuple of a word and a respective POS tag, and from this, we lemmatize each word. Lastly, we remove every entry with no review.

After completing preprocessing our data, it was adequate to summarize the main characteristics of our data through visual methods. We created a word cloud visualization that effectively illustrated the word counts in our data.

3.3 Vader Lexicon

3.3.1 Quantifying the Emotion of a Word

After cleaning our data, we are ready to perform sentiment analysis on the reviews. For this section, we used a pre-existing library called VADER (Valence Aware Dictionary for sEntiment Reasoning), a model used for text sentiment analysis that is sensitive to both polarity (positive/negative/neutral) and intensity (strength) of emotion [20]. The VADER sentiment analysis depends on a dictionary

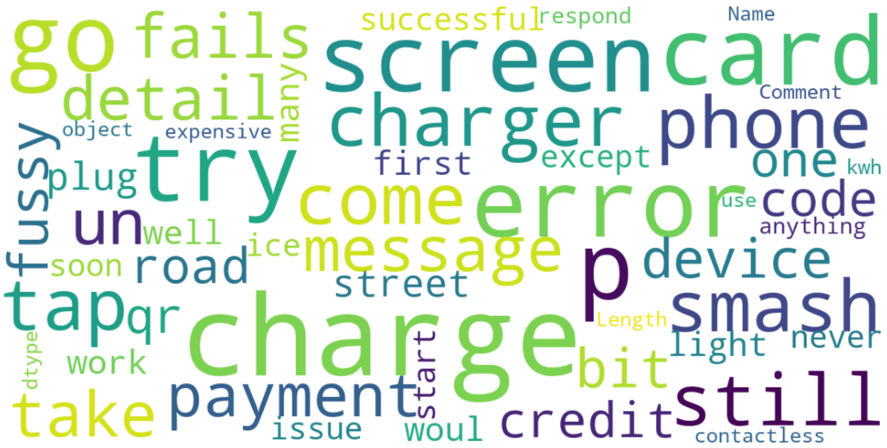


Figure 3: Word counts

that maps lexical features to emotion intensities called sentiment scores. All the lexical features are rated for the polarity and intensity on a scale from (-4, +4), where -4 is very negative, and +4 is very positive. A lexical feature is anything that is used in textual communication. Therefore, anything from emoticons to slang words such as “WTF” and “LOL” is mapped to intensity values. The average score is then used as the dictionary’s sentiment indicator for each lexical feature. For example, in Vader, the word “okay” has a positive rating of 0.9, “good” is “great” is 3.1, whereas “horrible” is -2.5, and “sucks” is -1.5. Lastly, any word excluded from the dictionary will be scored 0, which is neutral.

3.3.2 Quantifying the Emotion of a Sentence

The sentimental score of the entire text can be obtained by summing up the intensity of each word in the text and normalizing the final score between (-1, +1) from most negative to most positive using the function:

$$\frac{x}{\sqrt{x^2+\alpha}}$$

where x is the sum of the sentiment scores of each word within the sentence and alpha is set to be 15, approximating the maximum expected value of x.

3.3.3 Four heuristics

In Vader, other elements exist that affect the sentiment of an entire text. Below, we discuss the four heuristics incorporated in Vader, which have an impact on emotions and feeling [21].

1. Punctuation : **“love”** and **“love!!!”** convey two different emotions. Vader understands that the exclamation mark’s presence should positively or negatively intensify the text’s emotion. The same applies to a question mark instead of an exclamation mark. If the score of the text is positive, Vader adds a constant empirically-obtained value for every exclamation point (0.292) and question mark (0.18). If the score is negative, Vader subtracts.

2. Capitalization: Just like punctuations, Vader understands to increment or decrement the score of a capitalized word by 0.733, depending on whether the word is positive or negative.

3. Degree Modifiers: The effect of a degree modifier happens in two ways. One way is to increase the intensity of a base word, for example, " very good." Another way is to decrease the intensity of the base word. Words that increase the intensity of the base word are known as boosters, and the opposite is known as dampeners. Vader has a record of a dictionary that keeps track of these types of modifying words. The modifier's effect also depends on how far it is from the word it is modifying. Therefore, the further away from the modifier from the base word, the smaller the intensifying effect.

4. The shift in Polarity Due to "but": For example, "I love the movie, but I do not like the plot and how long it was." The first clause, "I love the movie," is positive, but the second one, "I do not like the plot and how long it was." is negative and the more dominant in terms of sentiment. Vader implements a "but" checker. All sentiment-bearing words before the "but" have their valence reduced to 50% of their values, while those after the "but" increase to 150% of their values.

3.3.4 Status column

In subsection 3.1, we described the variables within our data. The column contains five distinct values, namely, "Successful charge," "Issue Reported," "Comment," "Reply," and "Device Working." After the "Status" column, we have the "Comment" column, which is the review left by customers. It is important not to confuse the "Comment" column and the "Comment" value found in the "Status" column. One is a variable in our data, and one is a value that exists in another variable. From analyzing the reviews, we decided to use the "Status" as our label column and use the assumption that entries with "Successful charge" should have a positive sentiment and entries with "Issue Reported" should have negative sentiment, respectively.

3.3.5 Combining Vader lexicon with Status

Applying the Vader lexicon to our review data returns a dictionary of scores in four categories: positive, neutral, negative and compound. The compound score determines the review's sentiment by comparing it against a threshold of ±0.05. Below is the simple if and else code used to determine the sentiment polarity using the compound score for each review.

```
if compound score is >= 0.05
    sentiment is positive
else if compound score >= -0.05 and <= 0.05
    sentiment is neutral
else
    sentiment is negative
```

After aggregating each review with the Vader lexicon, we realized that it did not correctly classify many reviews. This is because most of the words used in the reviews do not exist in the pre-existing dictionary of Vader; subsequently, they are automatically set to have a compound score of 0. To fix this, we used an iterative process of updating the polarity of certain words. Now, going through every review and changing each word's polarity intensity is a cumbersome task. Since we are using the Status column in our dataset as a label, precisely the Successful charge and Issue Reported status values. We assumed that every review with a Successful charge status should result in positive sentiment and that every review with an Issue Reported status should consequently result in negative sentiment, as already mentioned in 3.3.5. Based on this assumption, we filtered every Successful charge review with a negative classified sentiment. We investigated why they were negative, and if the lexicon incorrectly classified them, we would change the polarity scores of some words that did not exist in the Vader dictionary and vice versa for the Issue Reported status. Through this iterative process, we observe its effect on other reviews and how the lexicon classified reviews better than before. Fig.4 illustrates the changes in sentiment for each status as we change the polarity of certain words.

3.4 Supervised Learning

Most machine learning algorithms are unable to process raw text. Instead, we have to apply text feature extraction, which allows us to pass our raw text to numerical features, which a machine learning algorithm can use. In the example below, we have three different texts in the list named *messages*. Every unique word or term in each text is transformed into a feature, and for each feature, we give them a value of 0 or 1 depending on whether it occurs in any of the documents. Documents are the three different texts that we have. If it appears in a document, we assign it a value of 1; else, 0. This results in a matrix format called a document-term matrix. The values in the document-term matrix are the token counts or the term frequency (tf).

However, instead of filling the document-term matrix with token counts, we will use the term frequency-inverse document frequency value for the term. The term frequency is the raw count of a term in a document. An inverse document frequency is the logarithmically scaled inverse fraction of the documents that contain the word. It is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of this quotient. The inverse document frequency factor is incorporated to diminish the weight of terms frequently occurring in the document set and increases the weight of the term that occurs rarely. The formula for this can be seen in equation 5, 6 and 7 respectively.

```
messages = ["Hey, lets go to the game today!",
            "Call your sister.",
            "Want to go walk your dogs?"]

from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer()
```

Table 1: Document term matrix with token counts

call	dogs	game	go	hey	lets	sister	the	to	today	walk	want	your
0	0	1	1	1	1	0	1	1	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	0	1	0	1	1	1

```
messages = ["Hey, lets go to the game today!",
            "Call your sister.",
            "Want to go walk your dogs?"]
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
vect = TfidfVectorizer()
dtm = vect.fit_tranform(messages)
```

Table 2: Document term matrix with TF-IDF value

call	**dogs**	**game**	**go**	**hey**	**lets**	**sister**	**the**	**to**	**today**	**walk**	**want**	**your**
0.00	0.00	0.40	0.31	0.40	0.40	0.00	0.40	0.31	0.40	0.00	0.00	0.00
0.62	0.00	0.00	0.00	0.00	0.00	0.62	0.00	0.00	0.00	0.00	0.00	0.47
0.00	0.46	0.00	0.35	0.00	0.00	0.00	0.00	0.35	0.00	0.46	0.46	0.35

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \cdot \text{idf}_t \tag{5}$$

$$\text{tf}_{t,d} = \text{number of times t appears in d} \tag{6}$$

$$\text{idf}_t = \log \frac{N}{(1+df)} \tag{7}$$

where:

- d is document
- t is term
- N is the total number of documents
- df is the number of documents with term t.

[22]

3.4.1 Support Vector Machine

SVM is a potential classification technology proposed by Vapnik et al., a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis [23]. Its main idea is that each sample is indicated as a point in space for a multi-dimensional sample set. And the system then randomly generates a hyperplane that moves continuously and classifies the samples until the points belonging to the same category distribute precisely on the same side of the hyperplane. Many hyper planes satisfy this condition, and we need to find such a plane to maximize the blank area between sides of it to achieve the optimal classification of these samples. For new data to be classified, we map it to the same space and predict the category based on its location [24].

3.4.2 Classification process

The classification process is done by creating a classification model based on training data and testing data. From the data set, 70% of the data were used as training data, and the remaining 30% were used as testing data. Our feature (X) is the review column, and our label (y) is the status column. Moreover, the label has three distinct values: Issue Reported, Successful charge and Others. After this, we apply TF-IDF to our review column, transforming our text data into a sparse matrix that can be passed to our machine learning algorithm. The algorithm used for the classification process is a support vector machine (SVM). The performance of the classification model is evaluated by its accuracy, precision, recall and f1 measure.

- Accuracy : percentage of correct decisions overall.
- Precision_x : correct decisions over instances assigned to class "x"
- Recall_x : correct assignments to class "x" over all instances of class "x" in test set
- f-score_x : combined measure of precision and recall

3.5 Latent Dirichlet Allocation

3.5.1 High Level Overview of LDA

Topic Modeling is used efficiently analyze large volumes of text by clustering documents into topics. In our scenario, the data is unlabelled; thus, we cannot apply previous supervised learning approaches to create machine learning models for the data. Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [25]. LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D [26]:

- Documents are probability distributions over latent topics
- Topics themselves are probability distributions over words

Morever, LDA assumes that documents are produced in the following fashion [26]:

- Decide on the number of words N the document will have.
- Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics).
- Generate each word in the document by using the topic to generate the word itself(according to the topic's multinomial distribution)
- Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

Given a set of documents. We have chosen some fixed number of K topics to discover, and want to use LDA to learn the topic representation of each document and the words associated to each topic. It is important to note that for LDA to work, as a user you should have some intuition on how many topics will be discovered. Furthermore, we go through each document, and randomly assign each

word in the document to one of the K topics. This random assignment already gives you both topic representations of all the documents and word distributions of all the topics
Now we iterate over every word in every document to improve these topics. For every word in every document and for each topic **t** we calculate:

- p(topic **t** | document **d**) = the proportion of words in document **d** that are currently assigned to topic **t**
- p(word **w** | topic **t**) = the proportion of assignments to topic **t** over all documents that come from this word **w**

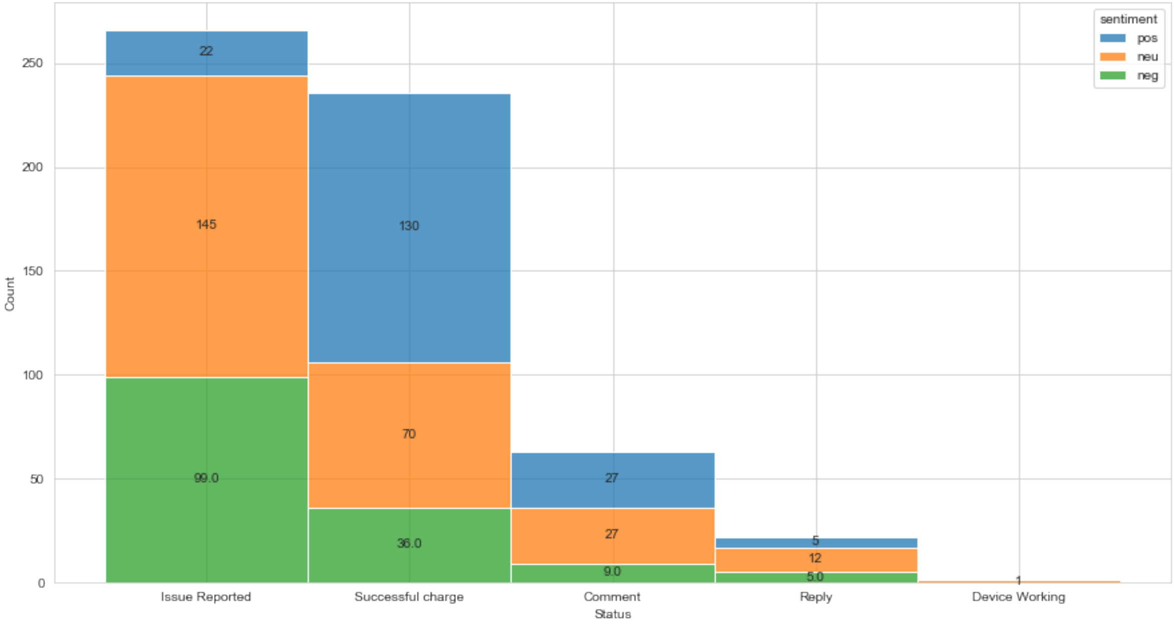
Then we reassign word w a new topic, where we choose topic t with probability **p(topic | document d) * p(word w | topic t)**. This is essentially the probability that topic t generated word w. After repeating the previous step a large number of times, we eventually reach a roughly steady state where the assignments are acceptable. Finally, we have each document assigned to a topic and we can also search for the words that have the highest probability of being assigned to a topic.
Lastly two things to note:

- The user must decide on the amount of topics present in the document
- The user must interpret what the topics are. For each topic, users can view the top ten words or any amount they wish. From looking at these words, the user can make an educated assumption on what the topic is

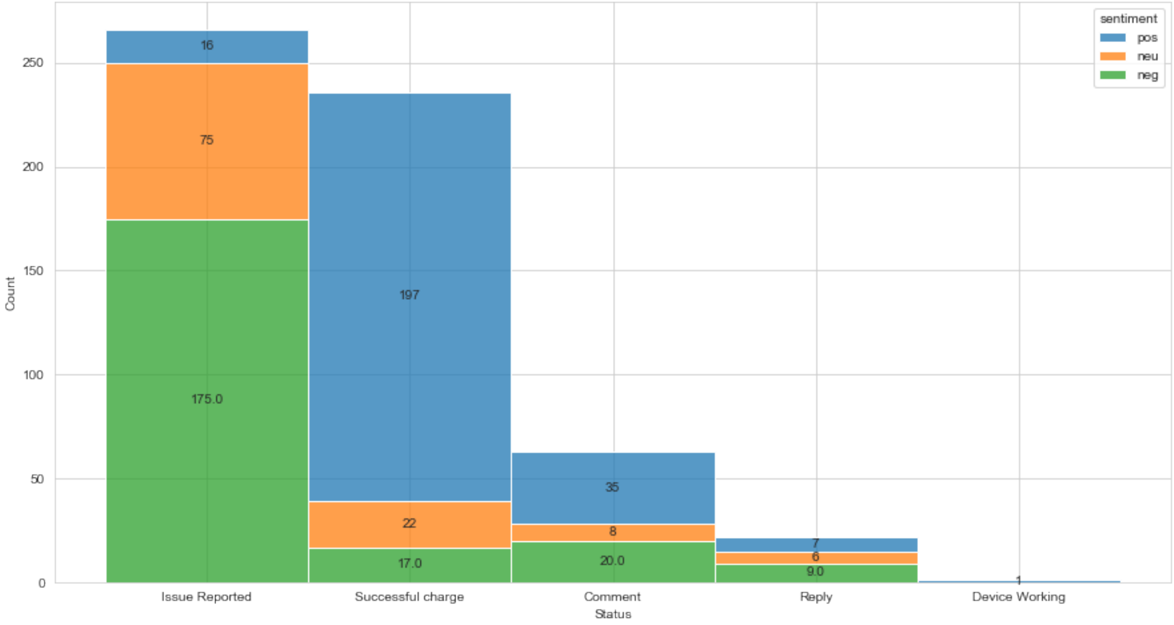
3.5.2 Implementation of LDA

Before we apply LDA, we must perform some preprocessing. Since our data has already been cleaned we will only perform feature extraction in the form term frequency–inverse document frequency which has been explained in section 3.4. Therefore we will have a document-term matrix which has tf-idf applied on it, then fit the document term matrix on our review data. Moreover, we fit the LDA model provided by scikit-learn on our document-term matrix, with the paramaters *n_components* and *max_iter* which is the number of topics and maximum iterations set to 4 and 100 respectively.

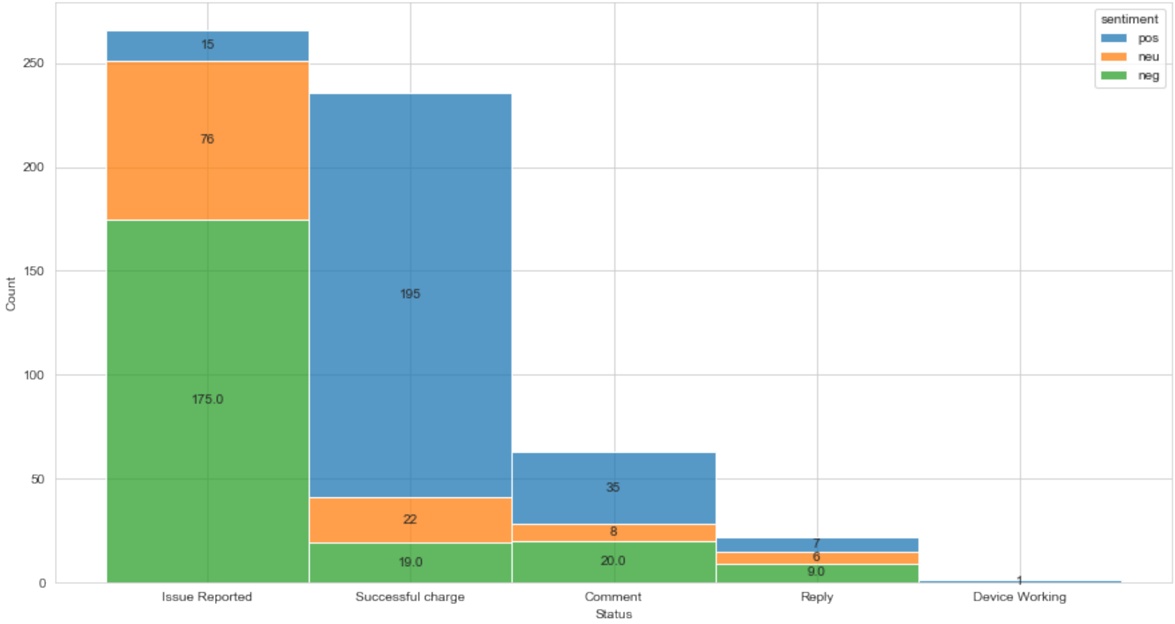
Initially, the results we were getting did not make sense regardless of how many times we increased or decreased the number of topics. Therefore, we attempted to apply the same process again to the data. However, this time the data only contain nouns. This attempt produced better results as the topic discovery began to make more sense. Finally the same process was repeated again but with nouns and adjectives only and this last attempt yielded much better results as well.



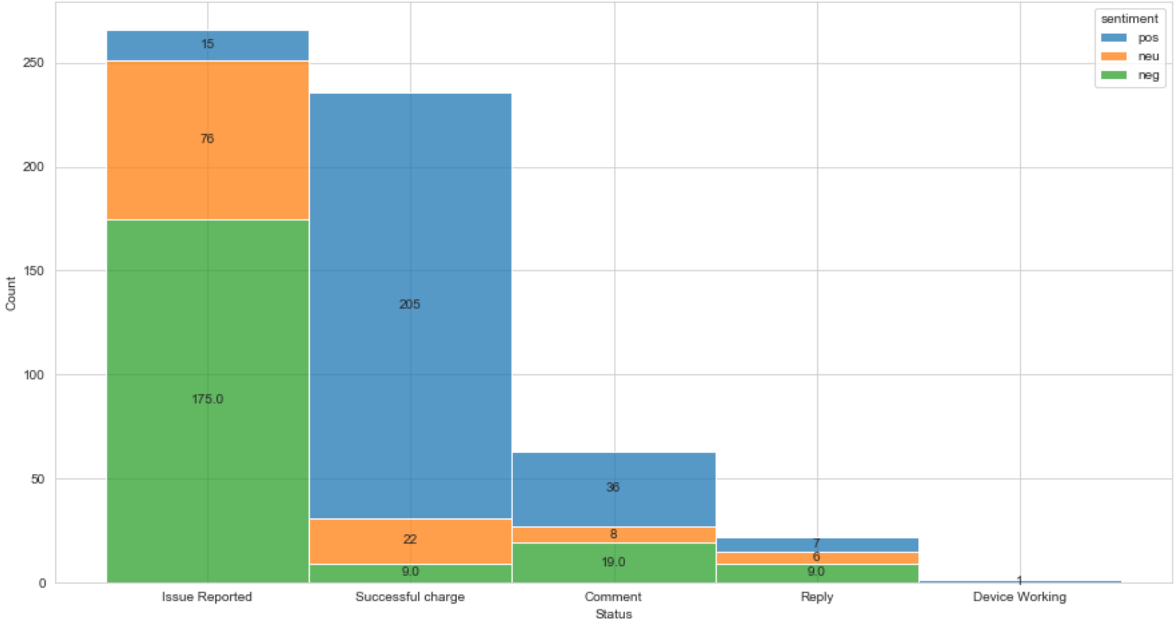
Iteration 0



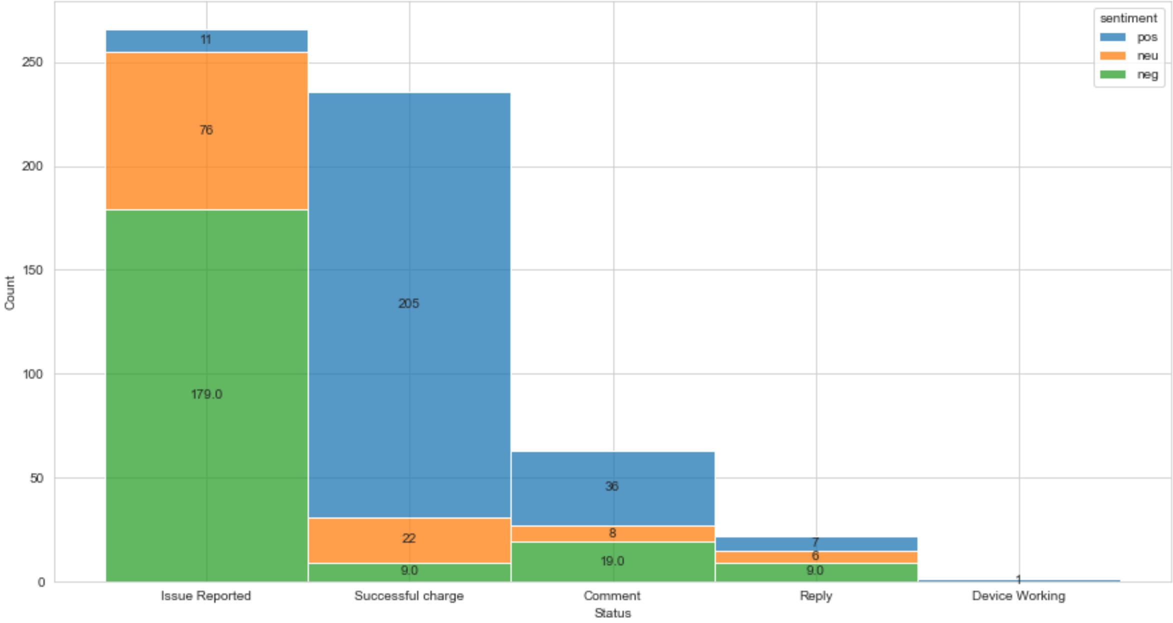
Iteration 1



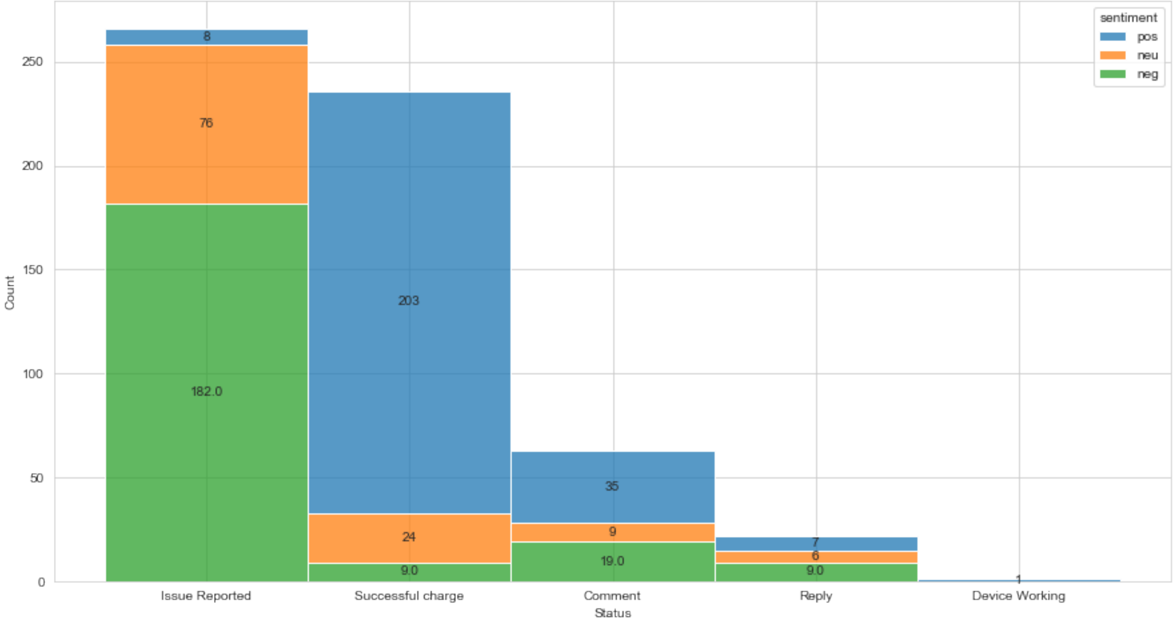
Iteration 2



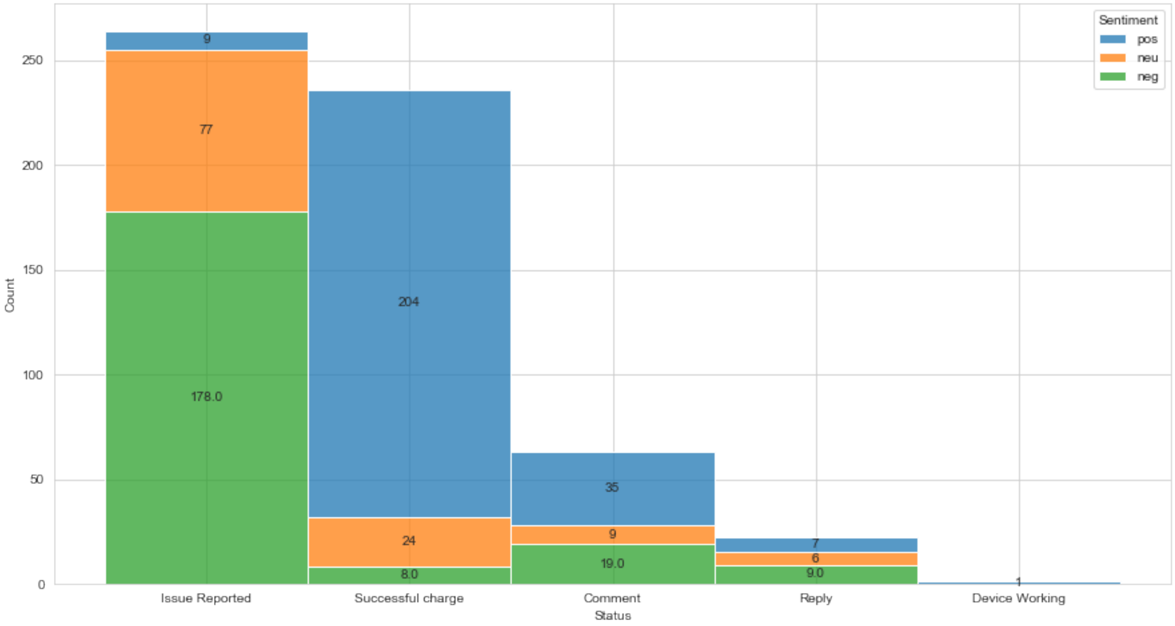
Iteration 3



Iteration 4



Iteration 5



Iteration 6

Figure 4: Sentiment count for each Status

4 Results

Following the completion of our implemented methods, we will aim to summarise and discuss the results found. Moreover, in this section we will discuss the results gain after applying the Vader Lexicon to our reviews and then compare those respective results to results from Supervised Learning approach. Lastly, summarise the various topics that exist within the reviews and also, characteristics discovered regarding sentiments. 3.

4.1 Lexicon Analysis

After aggregating the sentiment polarity for each review, it is important to summarise our results. Initially, we decided to see the difference between the polarity counts between preprocessed and unprocessed data. This is because we want to know if the lexicon performed better on uncleaned or cleaned data. Below you can view the polarity counts per data

Table 3: Polarity counts per data

	Unclean Data	**Clean Data**
pos	183	184
neu	252	256
neg	152	146

We can tell there is not much difference in how the Vader lexicon classifies these results. It is also important to note that these results took place before we performed the iterative sentiment analysis, where we changed the polarities of certain words. Hence, the increased classification in neutral sentiment scores. In addition, we can see how lexicon struggles to classify negative and positive reviews. This is because, the dictionary within the Vader lexicon does not recognise these words and for that reason, they are more likely to result in neutral reviews. This is further proven when we count the number of positive, neutral and negative words in our data, using the lexicon to classify which words it deemed positive, neutral and negative respectively. We see that there are 91 positive words, 829 neutral words, and 61 negative words. Consequently, when we investigate which words appear in the neutral word list, we see that these words are being used to describe a negative sentiment and on the other hand positive sentiment too.

In every classification problem, it is important to illustrate the performance of a method through metrics such as precision, recall, f1 score and accuracy. The definitions of these metrics highlighted in section 2.3.1. Below, we can view the values for each respective measure.

In table 4, the overall performance of the lexicon is 73%. Given that the lexicon finds it difficult to detect sarcasm and moreover, situations in which positive/negative words are used to describe negative/positive sentiments this result is relatively good. Furthermore, we are able to see that the lexicon classifier performs excellently for the positive and negative class in terms of precision. Also, the recall for the positive class is high which means that the vader lexicon is much better at classifying actual positive sentiments compared to negative sentiments. However, from the table the classifier struggles with respect to the neutral class, this is because the lexicon will classify reviews which have

Table 4: Classification metrics

	precision	**recall**	**f1 score**
pos	0.89	0.86	0.87
neu	0.01	0.11	0.02
neg	0.94	0.64	0.76
accuracy			0.73

a compound score of 0 as neutral or reviews which have an equal number of positive and negative words as neutral when actually the review could be positive or negative.

Furthermore, in figure 4, the effect of iterative sentiment analysis allowed us to classify most reviews correctly. We can view the change from iteration 0 to iteration 6. Initially, in iteration 0, we had 22, 145, and 99 positive, neutral and negative values for Issue Reported status, respectively. In iteration 6, we saw a performance increase in the classification of reviews, with 9, 77, and 178 positive, neutral and negative values, respectively, which is the goal we aimed for. For the Issue Reported status, we wanted to decrease the number of positive reviews and increase the number of negative reviews exponentially.

Additionally, for the Successful charge status, we achieved the same performance goal. Initially, in iteration 0, we had 130, 70, and 36 positive, neutral and negative reviews, respectively. Then, in iteration 6, we had 204, 24, and 8 positive, neutral and negative reviews, respectively.

Finally, with sentiment analysis, it is essential to summarise the common characteristics that are associated with certain sentiments. Some of the observations that were discovered are listed below.

- Charging stations from the Ubitricity network seem to be iced 19% of the time
- 38% of positive reviews come from the Shell Recharge Network
- 20% of positive reviews occur in the afternoon

4.2 Machine Learning Analysis

The same classification metrics previously discussed in section 4.1, are applied on our model to measure its performance. Table 5 illustrates the results found.
The accuracy of the model is 73% which is the same as the Vader lexicon hence, both methods have the same performance. We believe with more data points the accuracy will differ for each method. Furthermore, the model performs adequately in terms of recall with respect to Successful charge and Issue Reported. In addition to the recall values, the model performs satisfactory in terms of precision for Successful charge and Issue Reported. However, the Others class is not being classified correctly by the model. This is because of the less data points corresponding to that class.

Table 5: Classification metrics

	precision	**recall**	**f1 score**
Successful charge	0.79	0.86	0.83
Others	0.20	0.04	0.06
Issue Reported	0.71	0.84	0.77
accuracy			0.73

4.3 Topic Analysis

The optimal solution discovered was K = 4 combined with nouns and adjectives only. Initially, we started with K = 2, then incremented K till K = 5. At K = 5, we realised there were too many similarities between K = 4 and K = 5; to conclude, we discovered five different topics. Besides, since our data set was small, the chances of finding more topics were unlikely. To summarise, the four topics uncovered can be seen below:

- Topic 0 : successful charging sessions, screen
- Topic 1 : payment, kWh received and sentiment about respective price, not able to charge
- Topic 2 : charge point broken, connection issues, comments about the charge point
- Topic 3 : qr code, device offline, charge point dead

On top of that, for each topic we summarise the top 15 words associated with it. Table 6 illustrates that.

Table 6: Topics and their associated words

Topic	**15 words with the highest probability associated to that topic**
0	wait reader slow screen contactless start successful device card charger fine good charge kw work
1	connection cable use month unit power payment today expensive light service easy kwh charge iced
2	bad report car battery max speed rfid charger broken plug charge cc problem kw order
3	fast offline app qr hour station time code point dead available car use charge charger

Moreover, each review (document) was classified for the listed topic above. Below we can view some of the reviews and their assigned topics.

Table 7: Reviews and their associated words

Topic	**Comment**
0	screen still smash
2	rfid broken charge start via app cold battery kw
1	go take payment detail fails
0	device bit fussy credit card would would not accept otherwise ok pay coffee costa machine man till told get free coffee shell app would slightly offset expensive electricity
3	road one not qr code plug street light go would not charge

In summary, the differences in topics are not far apart. They all seem to overlap one another but distinct in their own way. For example, topic 0 and topic 2 raise concerns about charging sessions, in the case of topic 0 there are problems with the screen, in addition, topic 2 issues with charge point persist, however, in topic 0 there are many words that describe successful charging sessions which makes it distinct.

5 Conclusion

5.1 Summary of Main Findings

Customer satisfaction is an essential factor in determining the business success of any company. Amid the increase of EV charging solutions, evaluation of Shell's charging stations based on customer reviews is critical as it provides constructive feedback, so Shell can measure and improve the quality of services to achieve customer satisfaction. This research demonstrates techniques such as sentiment analysis and topic modeling to extract information from customer reviews using a lexicon-based approach and Latent Dirichlet Allocation. It has been observed that the lexicon used had an accuracy of 73%. Moreover, using LDA, we uncovered four topics within the reviews. Overall, there seems to be a correlation between the charge's point potential capacity and sentiment. In cases where customers can charge their vehicles and are content with the service received, it more likely results in positive sentiment, and the opposite applies.

5.2 Limitations & Future Work

In terms of limitation faced during this research is the inability to have access to labeled data. Given this situation, we used unsupervised techniques, topic modeling and a lexicon approach. This leads to our first future work consideration, which is using machine learning algorithms to classify sentiment polarity. This approach, given access to labeled data, will result in better classification of reviews. Another limitation encountered was our data being small. We believe with access to a more extensive data set; we could have explored more topics within the reviews, which also happens to be another future work we wish to look into.

References

[1] K. Ilieska, "Customer satisfaction index – as a base for strategic marketing management,"

[2] P. Federico, F. Elisabetta, M. Enza, and L. Bing, *Sentiment Analysis in Social Networks.* Morgan Kaufmann, 2017.

[3] H.-J. Kwon, H.-J. Ban, J.-K. Jun, and H.-S. Kim, "Topic modeling and sentiment analysis of online review for airlines," *Information*, vol. 12, no. 2, 2021.

[4] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," vol. 3, pp. 601–608, 01 2001.

[5] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, 2003.

[6] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[7] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins, "Progressive learning of topic modeling parameters: A visual analytics framework," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 382–391, 2017.

[8] Z. Wang, P. Gao, and X. Chu, "Sentiment analysis from customer-generated online videos on product review using topic modeling and multi-attention blstm," *Advanced Engineering Informatics*, vol. 52, p. 101588, 2022.

[9] S. Tirunillai and G. J. Tellis, "Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation," *Journal of marketing research*, vol. 51, no. 4, pp. 463–479, 2014.

[10] W. W. Moe and D. A. Schweidel, "Online product opinions: Incidence, evaluation, and evolution," *Marketing Science*, vol. 31, no. 3, pp. 372–386, 2012.

[11] R. Y. Kim, "Using online reviews for customer sentiment analysis," *IEEE Engineering Management Review*, vol. 49, no. 4, pp. 162–168, 2021.

[12] J. Berger, A. Humphreys, S. Ludwig, W. W. Moe, O. Netzer, and D. A. Schweidel, "Uniting the tribes: Using text for marketing insight," *Journal of Marketing*, vol. 84, no. 1, pp. 1–25, 2020.

[13] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *arXiv preprint cs/0205070*, 2002.

[14] G. D. Miner, J. F. Elder, T. Hill, R. A. Nisbet, D. Delen, and A. Fast, "Practical text mining and statistical analysis for non-structured text data applications," 2012.

[15] R. A. Laksono, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Sentiment analysis of restaurant customer reviews on tripadvisor using naïve bayes," in *2019 12th International Conference on Information Communication Technology and System (ICTS)*, pp. 49–54, 2019.

[16] D. Gupta, A. Malviya, and S. P. Singh, “Performance analysis of classification tree learning algorithms,” *International Journal of Computer Applications*, vol. 55, pp. 39–44, 2012.

[17] Wikipedia contributors, “Sentiment analysis — Wikipedia, the free encyclopedia,” 2022. [Online; accessed 24-February-2022].

[18] P. D. Turney, “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews,” 2002.

[19] S. Vashishtha and S. Susan, “Highlighting keyphrases using senti-scoring and fuzzy entropy for unsupervised sentiment analysis,” *Expert Systems with Applications*, vol. 169, p. 114323, 2021.

[20] C. J. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *ICWSM*, 2014.

[21] P. Calderon, “Vader sentiment analysis explained - pio calderon - medium,” Apr 2017.

[22] K. Chen, “Introduction to natural language processing — tf-idf,” May 2021.

[23] B. Xu, S. Chen, H. Zhang, and T. Wu, “Incremental k-nn svm method in intrusion detection,” in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 712–717, 2017.

[24] F. Miao, P. Zhang, L. Jin, and H. Wu, “Chinese news text classification based on machine learning algorithm,” in *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 02, pp. 48–51, 2018.

[25] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55. US Government printing office, 1964.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, p. 993–1022, mar 2003.