

Feature based Sentiment Analysis for Product Reviews

Dr. D. Sivaganesan

Professor, Computer Science and Engineering
PSG Institute of Technology and Applied Research
Coimbatore, Tamil Nadu, India

Sridhar K

Computer Science and Engineering
PSG Institute of Technology and Applied Research
Coimbatore, Tamil Nadu, India

Dhruv Aggarwal

Computer Science and Engineering
PSG Institute of Technology and Applied Research
Coimbatore, Tamil Nadu, India

Arunkumar M

Computer Science and Engineering
PSG Institute of Technology and Applied Research
Coimbatore, Tamil Nadu, India

Abstract—E-commerce is fast becoming one of the most preferred modes of shopping. The products are being reviewed before buying. Reviews play a vital role for the customers to make a choice regarding buying the product or not. Opinions collected from user's experiences regarding specific products directly influence future customer purchase decisions and also cause revenue losses due to negative reviews. Most often the reviews are very generic and doesn't describe the features of the product. If the product is popular, the number of reviews is very high and it's not possible for the customers to go through all the reviews one by one. Also, the reviews are long and it sometimes the customers read a few reviews only and this might lead to a biased view. Most of the existing systems using sentiment analysis classify the polarities as positive or negative only but the reason for the customers to give those comments are not known and hence it is difficult for other customers to know about the features being mentioned and also the product manufacturers won't find it easy to solve the issues. The work suggests feature-based sentiment analysis where the features are extracted from the reviews and corresponding polarities are predicted using the TextBlob and SVC model.

Keywords—Feature-Based sentiment analysis, Aspect Term, Opinion Term, Sentiment Polarity, TextBlob, SVC.

I. INTRODUCTION

E-commerce is fast becoming one of the most preferred modes of shopping. The products are being reviewed before buying. There are a lot of reviews for a product and most of those reviews don't describe about the features of the product and hence it becomes difficult for other customers to get proper honest review of the product they wish to buy. The sentiment analysis based on the features extracted and predicting the corresponding polarities will provide the customers as well as the product manufacturers better clarity on the features mentioned by the customers. The use of trained model with large data set can make it easier to identify the features mentioned in the comments and the description of the product page and the predicting the corresponding polarities. The reviews are web scraped from the e-commerce website using selenium scripts and then pre-processed where the noise such as emoticons, smiles are removed. As a result, the performance of the classifier and the speed of the classification process will be increased. These sentences are

identified and removed by the regex parser in the Natural Language Toolkit (NLTK). This project uses SVM since they are able to classify the feature extraction into positive and negative when trained with train data.

II. RELATED WORK

Many research papers related to product reviews, sentiment analysis and opinion mining have come up recently. In the work [1] Xing Fang and Justin Zhan performed sentiment polarity categorization. They categorized the text into positive, negative or neutral polarity. They have performed sentiment polarity categorization at two levels, namely, sentence-level categorization and review-level categorization. Sentence-level categorization classifies sentences as positive or negative in terms of the sentiment that it conveys. Review-level categorization classifies the reviews based on the ratings given for the reviews. 4-star ratings are classified as positive, 1-star ratings as negative. 2 and 3-star ratings are classified as neutral. The approach implemented is a bag-of-word model that simply counts the appearance of positive or negative (word) tokens for every sentence. If there are more positive tokens than negative ones, the sentence will be tagged as positive, and vice versa. Sentimental polarity categorisation process is performed into 3 phases. In phase 1, the data is reviewed, sentiment sentence extraction takes place and part-of-speech tagging is performed. In phase 2, an algorithm is proposed and implemented for negation phrases identification. Also, a mathematical approach is proposed for sentiment score computation, a feature vector generation method is presented for sentiment polarity categorization. In Phase 3, two sentiment polarity categorization experiments are respectively performed based on sentence level and review level. Finally, the performance of three classification models, namely Naïve Bayesian, Random Forest, Support Vector Machine are evaluated and compared based on their experimental results.

In [2], the authors used Aspect Sentiment Triplet Extraction (ASTE). The triplets (What, How, Why) from the input sentence are extracted which shows WHAT the targeted aspects are, HOW their sentiment polarities are and WHY they have such polarities (or) opinion reasons.

For instance, one triplet from "The battery life is amazing" could be ('battery life', positive, 'amazing').

It proposed two-stage framework. In the first stage, potential aspect terms are extracted together with their sentiment and potential opinion terms are extracted. In the second stage, after obtaining all the possible candidate triplets from the stage one, each triplet is sent to a binary classifier. The classifier was trained on the ground truth aspect and opinion pairs in the training set. The model performing the best on the validation set was used as the stage two classifier for evaluating both our model and baselines. The goal is to pair up aspects with the corresponding opinion expressions. As we observed, for sentences with multiple aspects and opinions, word distance is very indicative for correctly pairing up an aspect and its opinion.

In [3], the authors aimed to automate the process of reading content from thousands of reviews. The redundant data which doesn't assist in sentiment analysis is eliminated. The commonly used words or stop words are carefully removed. To achieve this, we have employed part-of-speech-tagging (POS tagging) to capture the real sentiment of the review. Instead of explicitly eliminating all the stop words, the authors proposed an approach to select only valuable content. To achieve this, they used the functionality of gensim library to perform text lemmatization to filter the adjectives, adverbs, verbs and nouns from the rest of the Corpus (collection of reviews). Then these words are represented in the form of vectors using TF-IDF vectorizer technique. TF-IDF makes calculation easy, helps extract the most descriptive keywords in a document and measures the uniqueness and relevance of the content. Two commonly used classifiers Naïve Bayes and SVM are used to classify the TF-IDF vectors. Also, a Feed-Forward Neural Network is trained and then the results are compared on the basis of their performance. The aim of this work is to provide a novel approach to classify the reviews as negative or positive and give the best suitable model to capture the sentiment of online customer reviews.

In [4], the authors have proposed phrase dependency parsing. Product features usually contain more than one word, such as "LCD screen", "image colour", "Canon PowerShot SD500", and so on. Product Features products, product parts, properties of products, properties of parts, company names and related objects. For example, in consumer electronic domain, "Canon PowerShot", "image quality", "camera", "laptop" are all product features. An incomplete product feature will confuse the successive analysis.

For example, in passage "Image colour is disappointed", the negative sentiment becomes obscure if only "image" or "colour" is picked out. Since a product feature could not be represented by a single word, dependency parsing might not be the best approach here unfortunately, which provides dependency relations only between words. Previous works on relation extraction usually use the head word to represent the whole phrase and extract features from the word level dependency tree. This solution is problematic because the information provided by the phrase itself cannot be used by this kind of methods. And, experimental results show that relation extraction task can benefit from dependencies within a phrase. To solve this issue, the concept of phrase dependency parsing is proposed. Phrase dependency parsing segments an input sentence into "phrases" and links segments with directed arcs. The parsing focuses on the "phrases" and the relations between them, rather than on the single words inside each phrase. Because phrase dependency parsing naturally divides the dependencies into local and global, a

novel tree kernel method has also been proposed. This approach is performed in three steps, (1) constructing phrase dependency tree from results of chunking and dependency parsing; (2) extracting candidate product features and candidate opinion expressions; (3) extracting relations between product features and opinion expressions. A dependency relationship which is an asymmetric binary relationship holds between two phrases. One is called head, which is the central phrase in the relation. The other phrase is called dependent, which modifies the head.

Consider the following example:

"We really enjoyed using the Canon PowerShot SD500."

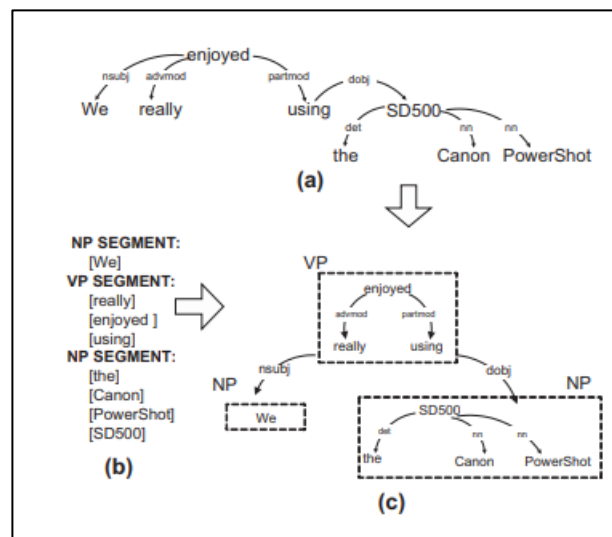


Fig. 1. Example of Phrase Dependency Parsing

Fig.1 shows the procedure of phrase dependency parsing. Fig.1(a) is the result of the lexical dependency parser. Shallow parsers result is shown in Fig.1(b). Chunk phrases "NP(We)", "VP(really enjoyed using)" and "NP(the Canon PowerShot SD500)" are nodes in the output phrase dependency tree. When visiting node "enjoyed" in Fig.1(a), the shallow parser tells that "really" and "using" which are children of "enjoy" are in the same phrase with their parent, then the three nodes are packed. The final phrase dependency parsing tree is shown in the Fig. 1(c).

In [5], the authors have mentioned about different approaches used for opinion mining and sentiment analysis, namely, (1) Machine learning based, (2) Lexicon based.

The machine learning based approach uses various supervised and unsupervised learning algorithms for sentiment classification. While the Lexicon based techniques use a lexicon dictionary with sentiment words related to specific domain for sentiment classification. The dictionary contains the information about the polarity of each word, whether the words are positive or negative. The word in the sentence can be matched with the words in dictionary to determine their polarity.

The machine learning approach can be classified in two categories, supervised and unsupervised. Most of the researchers focused on supervised classification for sentiment analysis. In a machine learning based approach, two sets of documents are needed: training set and test set. Training and testing data used are mostly product reviews. An automated classifier uses a training set to learn the differentiating

characteristics of documents. A test data set tells how well the classifier performs. The Machine learning approach first collects the training dataset. After collecting a training dataset, the classifier will be trained on the training data. The supervised machine learning techniques give more accuracy of sentiment classification.

Lexicon based approach does not require a prior training of dataset. This approach is basically an unsupervised approach, as it involves the creation of sentiment lexicon through the unsupervised labelling of words with their sentiment polarity or subjectivity status. It is also known as semantic orientation approach to opinion mining. Lexicon based approaches do not require time for training data. Most domains lack labelled training data, in these cases unsupervised methods is very useful for developing applications. This study concludes that machine learning techniques have performed better in terms of accuracy than lexicon-based methods in sentiment analysis. However, the performance of machine learning approaches is heavily dependent on the selected features, quality and quantity of training data and the domain of the dataset. The lexicon based approach do not require time for training. Therefore, the combined approach of machine learning and lexicon based techniques should be used to avoid the weaknesses of both the techniques and to preserve their strength. The combined approach of machine learning and lexicon based techniques gave the best accuracy of the sentiment classification as compared to the machine learning and lexicon based approaches. It is inheriting high accuracy from machine learning approach and achieving stability from lexicon based approach to enhance the performance of sentiment classification.

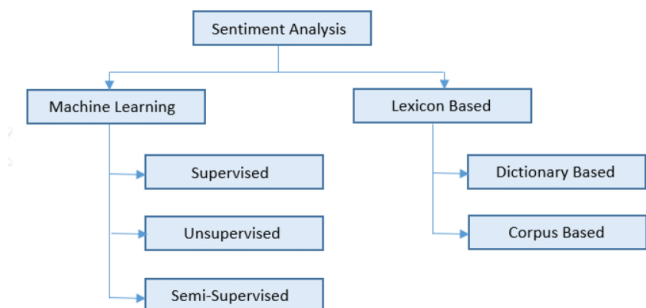


Fig. 2. Different Approaches in Sentiment Analysis

III. PROPOSED METHODOLOGY

In the existing models, no pre-processing has been done on the dataset. Pre-processing refers to the process of data being cleaned before applying algorithms to it. Online reviews contain many irrelevant data such as white spaces, emojis. Higher accuracies can be achieved by removing these stop words.

Some models have used Bag of Words model which doesn't take into account the positioning of the words in the reviews which can have a different impact on the semantic of a review.

For example,

"The battery life is great, the camera quality is superb and sound quality is amazing. However, it has heating issues."

The overall sentiment of the above mentioned review is negative, the model would classify the review as positive because of the number of the positive words it contains.

Another challenge which existing models face is the identification of the negation and its effect on the semantic understanding of the sentences.

For example,

"The battery life is not good"

The overall polarity of this sentence has become negative due to the negation word – 'not'. The existing models would classify this review as positive ignoring the effect of the negation word. In order to overcome the above mentioned challenges, we have proposed a system.

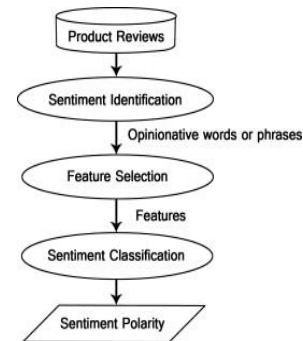


Fig. 3. Flow of the proposed methodology

Due to lack of available dataset that specifically focuses on features and correlated opinion for mobiles, it becomes difficult to create a system for Feature based sentiment analysis. The proposed system makes use of Yake (Yet Another Keyword Extractor) for extracting keywords from product details. These keywords are the most relevant words with respect to the product. Dependency parsing is used to analyse the customer reviews and provide aspect (feature) term and opinion (feedback) term. Using these aspect and opinion terms, we can determine user's opinion and sentiment for the given review. In order to analyse the sentiment of the opinion term, Support Vector Machine based classifier is used. This classifier is trained with feature and opinion terms dataset created manually after analysing approximately 3000 reviews for different mobiles.

IV. IMPLEMENTATION

A. Web Scrapping and Pre-Processing

Due to limitation on the usage of free APIs for Amazon product data, web scrapping using Selenium is chosen. When provided with a website, first the ASIN number is extracted. ASIN is Amazon Standard Identification Number which is used to identify every product listed on Amazon. Other important is also extracted such as product name, product features details and reviews.

For each product, Amazon has over 10000 reviews. But many of reviews are not authentic and may convey false information. In order to improve the authenticity of reviews, only reviews from **Verified Purchase** are taken for a period of 6 months from current data as shown in Fig. 4.

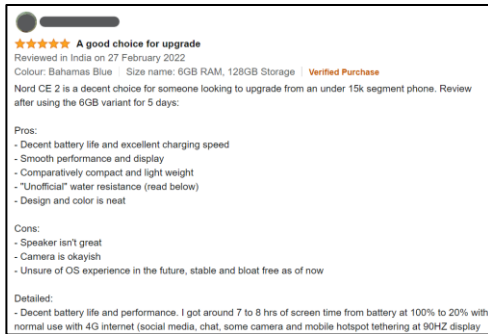


Fig. 4. Sample review considered for sentiment analysis

B. Feature Extraction

In order to extract key features of a product, product description is used. Product description conveys important information regarding the product as a whole and features offered such as camera and display in case of a smart phone. These keywords will act as an easy way for user to view all the features of the product. These keywords also signify the most talked about features due to its nature of extraction.

C. TF-IDF based keyword extraction

TF-IDF is a very efficient technique to find out meaningful words from a paragraph or group of paragraphs and cancel out the incapable or useless words. TF-IDF stands for Term frequency – Inverse Document Frequency. It is calculated based on 2 main terms i.e., Term Frequency and Inverse Document Frequency. Term frequency is ratio of occurrence of term t in document d to total words in document d as show in (1).

$$tf(t, d) = \frac{\text{count of term } t \text{ in document } d}{\text{number of words in } d} \quad (1)$$

Document Frequency $df(t)$ is termed as occurrence of t in all documents as shown in (2). Based on (2), we can simplify and write inverse document frequency as logarithm of ratio of total number documents d in corpus D to document frequency as shown in (3). This can alternatively be written as in (4).

$$df(t) = \text{occurrence of } t \text{ in documents} \quad (2)$$

$$idf(t) = \log \frac{N}{df(t)} \quad (3)$$

$$idf(t, D) = \log \frac{N}{\text{count}(d \in D: t \in d)} \quad (4)$$

TF-IDF is a very efficient and easy technique to extract keywords but suffers from a main issue of evaluating each word separately. Many words like *operating system* and *gorilla glass* are critical to smartphones. When TF-IDF is used, these words are treated separately instead of as a single group of words, hence increasing the possibility of these words appearing as a keyword separately. Words like *operating* makes sense only when added in front of a noun such as *system*. The system works well in presence of multiple available documents d for a given term t in corpus D . In case of a smart phone, there is only one source of information like product details on Amazon website making available number of documents limited to 1. This proves to be a challenge while using TF-IDF as a keyword extraction mechanism.

D. Yake based keyword extraction

The issues faced by TF-IDF based system can be eliminated by Yake. Yake is a novel based multilingual keyword extraction system [6]. The major advantage of Yake is the possibility of using it for a single document. Yake can generate a keyword as long as 3 words which is also known as 3-gram. Initially all phrases with punctuation marks or begin or end with a stop word are removed. Candidate scoring in Yake is done on 5 main parameters: casing, word positional, word frequency, word relatedness to context and word different sentence. In case of casing, words are given higher importance if they are capitalized or acronyms such as "ISRO". It is also observed that words occurring at the beginning of document tend to be more important, hence assigned higher score under word positional. Under word frequency, the frequency of the word is calculated normalized by 1-standard deviation from mean. In order to calculate the relatedness to context, words or terms occurring on the left and right of the candidate word are also seen. Words occurring on left or right of different words frequently tend to be stop words or unwanted words and hence avoided. A word occurring in different sentences are assigned higher score under word different sentence.

The issue with product details is presence of too many technical details which may not be referred frequently by the customer. The issue with customer review is the presence of many unwanted terms and opinion terms such as *amazing*, *superb* or *bad*. These words signify the opinion of a feature and not product feature itself. To avoid such issues, this algorithm is applied on both product details and customer reviews. A combination of keywords from both product details and customer reviews are taken. These keywords are apt to the product and signify the features while avoiding the technical terms too complex for normal customers.

E. Aspect and Opinion term extraction

Aspect terms are attributes or components of an entity (a product or service). Opinion term show the opinion of the user regarding a specific product or service. These terms specify the sentiment of the user. For e.g., in *bad service*, aspect term is service and opinion term bad. This signifies negative sentiment for service offered by the product. Dependency parsing is used to identify the aspect word and associated opinion term. The POS tagging is applied to the sentence to get part of speech each word belongs to. For e.g., for a sentence "*amazing battery and performance*", POS tagging would assign tags as specified in Table I.

TABLE I. POS TAGGING FOR SAMPLE TEXT

Word	Part of Speech	
	POS Tag	Full Form
amazing	JJ	Adjective or Numeral/Ordinal
battery	NN	Noun (Common, Singular or Mass)
and	CC	Coordinating Conjunction
performance	NN	Noun (Common, Singular or Mass)

Initially, the sentence is analysed and adjacent noun terms are identified. In order to simplify these sentences, adjacent noun terms are combined together to form a single noun term. Any noun term separated from each other by, or **and** are also combined to form a single term. This makes the structure of the sentence less complex making it easy to analyse. These

combined words are mapped to actual words using a dictionary for later purpose.

Stanza is a collection of accurate and efficient tools for the linguistic analysis of human languages [7]. It is a Python based natural language analysis package. It can be used to convert a human language text into list of sentences and words along with its corresponding relationship. For e.g., When Stanza Pipeline is used to analyse the text “*amazing battery and performance*”, we can visualize the following text in form of dependency graph as shown in Fig. 5.

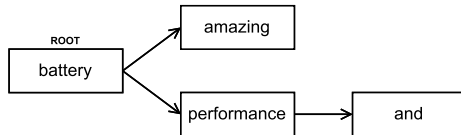


Fig. 5. Dependency graph for text “*amazing battery and performance*”

TABLE II. DEPENDENCY TABLE FOR SAMPLE TEXT

Id	Word	Dependency	
		Head Id	Dependency Relation
1	amazing	2	amod (Adjectival Modifier)
2	battery	0	root (Root word)
3	and	4	cc (Coordinating Conjunction)
4	performance	2	conj (Conjunction)

Using the POS tags assigned to each term, only those terms are considered who are:

- JJ (Adjective);
- JJR (Comparative Adjective);
- JJS (Superlative Adjective);
- NN (Noun-Singular);
- NNS (Noun-Plural);
- NNP (Proper Noun);
- NNPS (Proper Noun-Plural); and
- RB (Adverb). [8]

Using Stanza, the dependency relationship between different terms is found out as shown in Table II. The relations between selected terms are only considered. The most desired relations are:

- acl:relcl (Relative Clause Modifier);
- acomp (Adjectival Complement);
- advmod (Adverbial Modifier);
- agent (Agent);
- amod (Adjectival Modifier);
- compound (Compound);
- dobj (Direct Object);
- neg (Negation modifier);
- nsubj (Nominal Subject);
- obj (Object); and

- xcomp (Open Clausal Complement). [9]

The relations with head term as a Noun form (NN, NNS, NNP or NNPS) are identified. These head terms form the *Aspect term*. The correlated dependent term form the *Opinion term*. In both aspect and opinion terms are cleaned to remove any stop words or any unwanted words like *have, has, the, etc.* For e.g.,

Nice buy. Very costly, but high end built and premium feel. Excellent triple camera with LiDAR. Face unlock works with spectacles and mask on.

For the given customer review, when we analyse it using dependency parser, we obtain a dictionary of all aspect words and corresponding opinion terms as shown in Fig. 6.

```

{
  "buy": ["nice"],
  "end": ["high", "built"],
  "feel": ["premium"],
  "triple camera": ["excellent"],
  "face unlock": ["works"]
}
  
```

Fig. 6. Aspect-Opinion terms for sample text in form of dictionary

F. Sentiment Analysis using TextBlob

TextBlob is used for sentiment analysis of opinion word. TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part of speech tagging, noun phrase extraction, sentiment analysis, classification, translation and more [10]. For lexicon-based approaches, a sentiment is defined by its semantic orientation and the intensity of each word in the sentence. This requires a pre-defined dictionary classifying negative and positive words. Generally, a text message will be represented by bag of words. After assigning individual scores to all the words, final sentiment is calculated by some pooling operation like taking an average of all the sentiments.

TextBlob returns polarity and subjectivity of a sentence. Polarity lies in between -1 and +1. A polarity of -1 shows negative sentiment while polarity of +1 shows positive sentiment. A higher value of polarity signifies more positive opinion. For e.g., *good* is assigned a polarity of 0.7 while *best* is assigned a polarity of 1.0. Using TextBlob, sentiment level of each opinion word is determined. The major issue with TextBlob is that it is unable to determine the polarity specifically for the feature. For e.g., *light weight slim* is positive feedback while *light lagging* is negative feedback. TextBlob classifies both the statement as positive feedback. This issue can be solved by using SVC model trained with feature specific data.

G. Sentiment Analysis using Support Vector Classifier

Due to lack of available datasets specifically focusing on features and their corresponding opinion, it becomes difficult to train a model. Based on the data extracted from Amazon website, a dataset of 2325 Aspect-Opinion words is created with 1695 positive opinions and 630 negative opinions. To train the model, the data is split into 80 % train data and 20 % test data.

In Model 1, the words are vectorized using TF-IDF vectorizer available in Scikit Learn library. The feature and opinion are considered as a single text. The model is trained

to fit the training vector and tested later for test vector. In Model 2, the column transformer is used to combine feature and its associated opinion into a single transformer. The TF-IDF vectorizer is applied on both feature and opinion.

Major issue with this model is the fact that words such as *no* or *not* vary the meaning of the entire word but is not considered by the model. It is found that words like *issue* and *no issue* are classified as negative sentiment when they may belong to different class. This issue can be resolved by considering negative words like *no* and *not* separately and shifting the polarity of word based on their occurrence in the sentence. Another issue is that dataset is biased towards positive opinion. It has 3 times more positive opinion than negative ones. This can be solved with use of better dataset.

V. EXPERIMENTAL RESULTS

Using three different algorithms to identify the sentiment using the feature and opinion term, different precision and recall is obtained. The best precision and recall for all the three models are shown in Table III. A sample dashboard showing analytics regarding the feature *battery* is shown in Fig. 7. Google Charts are used to make graphs in HTML template. This dashboard is for the product *OnPlus Nord CE 2*.

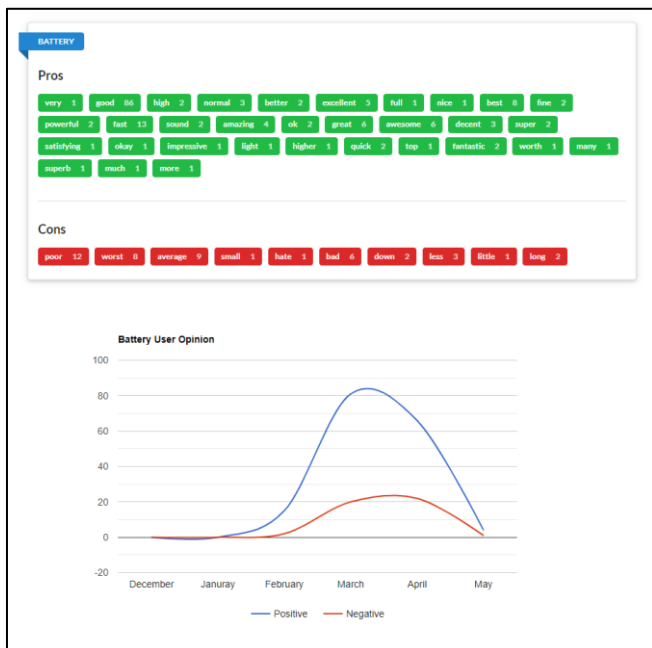


Fig. 7. Analysis Dashboard for feature *battery* for the product *OnePlus Nord CE 2*

TABLE III. MODEL PERFORMANCE COMPARISON IN PERCENTAGE

	Sentiment Models		
	TextBlob	Model – 1 Basic SVM	Model – 2 Column Transformer
Accuracy	86.02	92.26	92.04
Positive			
Precision	89.44	91.95	91.93
Recall	92.99	97.56	97.26
F1-score	91.18	94.67	94.52
Negative			
Precision	83.33	93.16	92.37
Recall	69.34	79.56	79.56
F1-score	75.70	85.83	85.49

	Sentiment Models		
	TextBlob	Model – 1 Basic SVM	Model – 2 Column Transformer
Weighted Average			
Precision	87.64	92.31	92.06
Recall	86.02	92.26	92.04
F1-score	86.62	92.07	91.86

VI. CONCLUSION AND FUTURE WORK

In this paper, feature based sentiment analysis is performed on the product reviews taken from Amazon. The most talked about features are extracted using Yake, the corresponding opinions about these features are extracted and polarity of the opinions whether positive or negative along with the magnitude of the polarity is calculated.

In the future, a more advanced and powerful dependency parser can be used that could also identify the words adding the value to the opinion term. For e.g., in *very good service*, the word *very* adds more positivity to the existing positive sentiment *good*. A major issue with existing SVC model is the fact that oppositely meaning words are sometimes very closely located and hence classified into opposite sentiment. This issue is also contributed by the fact that existing dataset is highly skewed towards positive feedback or opinion. A larger dataset with more balanced data would help resolve this issue.

REFERENCES

- [1] Xing Fang, Justin Zhan, "Sentiment Analysis using Product Review Data" in Journal of Big Data, SpringerOpen Journal, June-2015
- [2] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, Luo Si, "Knowing What, How and Why: A Near Complete Solution for Aspect-based Sentiment Analysis" in Proceedings of the AAAI Conference on Artificial Intelligence, Volume-34, Issue-5, November-2019
- [3] Neha Bhujbal, Gaurav Baydane, "A Novel Approach for Analyzing Sentiment of Customer Reviews using Gensim and TF-IDF" in International Research Journal of Engineering and Technology, Volume-8, Issue-4, April-2021.
- [4] Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu, "Phrase Dependency Parsing for Opinion Mining" in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Pages 1533-1541, August-2009.
- [5] Ashwini Patil, Shiwani Gupta, "A Review on Sentiment Analysis Approaches", March-2021.
- [6] Campos, R., Mangaravite, V., Pasquali, A., Jatowt, A., Jorge, A., Nunes, C. and Jatowt, A. (2020). YAKE! Keyword Extraction from Single Documents using Multiple Local Features. In Information Sciences Journal. Elsevier, Vol 509, pp. 257-289.
- [7] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Association for Computational Linguistics (ACL) System Demonstrations. 2020.
- [8] Bird, S., Klein, E., & Loper, E. (n.d.). Categorizing and Tagging Words. NLTK. Retrieved May 30, 2022, from https://www.nltk.org/book/ch05
- [9] de Marneffe, M. C., Filip Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Nivre, J., Petrov, S., Pyysalo, S., Schuster, S., Silveira, N., Tsarfaty, R., Tyers, F., & Zeman, D. (n.d.). Universal Dependency Relations. Universal Dependencies. Retrieved May 30, 2022, from https://universaldependencies.org/u/dep/
- [10] Loria, S., Keen, P., Honnibal, M., Yankovsky, R., Karesh, D., Dempsey, E., Childs, W., Schnurr, J., Qalieh, A., Ragnarsson, L., Coe, J., López Calvo, A., Kulshrestha, N., Eslava, J., Albert, E., Harden, T. J., Pavelmalai, Kolb, J., Ong, D., . . . Rachum, R. (n.d.). TextBlob: Simplified Text Processing. TextBlob. Retrieved May 30, 2022, from https://textblob.readthedocs.io/en/dev/.