eecjournal

International Journal of Electrical, Electronics and Computers
Vol-6, Issue-6 | Nov-Dec, 2021
Available: https://aipublications.com/ijeec/
Peer-Reviewed Journal

Comparative Study on Lexicon-based sentiment analysers over Negative sentiment

Subhasis Sanyal, Mohit Kumar Barai

Samsung Research Institute, Noida, India.

Received: 25 Sep 2021; Accepted: 03 Nov 2021; Date of Publication: 16 Nov 2021

Abstract**—***Sentiment Analysis or Opinion Mining is one of the latest trends of social listening, which is presently reshaping Commercial Organisations. It is a significant task of Natural Language Processing (NLP). The vast availability of product review data within Social media like Twitter, Facebook, and e-commerce site like Amazon, Alibaba. An organisation can get insight into a customer's mind based on a product or what type of opinion the product has generated in the market. Accordingly, an organisation can take some reactive preventive measures. While analysing the above, we have found that negative opinion has a strong effect on customers' minds than the positive one. Also, negative opinions are more viral in terms of diffusion. Our present work is based on a comparison of two available rule-based Sentiment analysers, VADER, and TextBlob on domain-specific product review data from Amazon.co.in. It investigates, which has higher accuracy in terms of classifying negative opinions. Our research has found out that VADER's negative polarity sentiment classification accuracy is more elevated than TextBlob.*

Keywords**—*Crowdsourcing, RAKE (Rapid Automatic Keyword Extraction), TextBlob, VADER (Valance Aware Dictionary and sEntiment Reasoner), WOC (Wisdom of Crowd.***

I. INTRODUCTION

Social listening is a technique to analyse a domain-specific conversation or review provided by Social Media users based on that domain. In this regard, the domain can be a product, a political subject, a living being, etc. Any entity which has a subjective value with the miscellaneous aspect or features can be considered as a domain. Formerly we have mentioned conversation or review; these are nothing but a wealth of knowledge provided in Natural Language [1]. This type of knowledge can provide sustainability [2]. Sentiment analysis or opinion mining is a sub-part of social listening which analyses people's opinion, sentiment, evaluation, attitude, and emotion from a written language perspective [3] [4]. With the advent of cloud-like disruptive technology, the paradigm of commercial business has concocted a novel field of e-commerce [5]. The growth and far-reaching capability of this field are assertive that e-commerce organisations like Amazon, Alibaba, etc. have given a unique status quo. Within this e-commerce medium, not only a seller can sell their items, but also they can view the opinion of buyers on their products and can take a reactive measure. A buyer can share his or her feedback (review) of a product. Each of these feedbacks is nothing but valuable information about the concerned domain (product), which exhibits the sentiment polarity of a consumer on that certain domain (product). Reviews can be good or bad, but seldom neutral [6]. 'Bad' can be considered as negative sentiment, and 'Good' can be considered as a positive sentiment. Our study has found out that negative opinions on a subject have a strong impact when compared to positive [7] [8] [9] and also more viral in terms of proliferation. Tsugawa, Ohsaki has said [7] that "diffusion volume of negative tweets was 1.2–1.6-fold that of positive and neutral tweets, and that the diffusion speed of negative tweets was 1.25-fold that of positive and neutral tweets when the diffusion volume was large". The same impact can be inferred within e-commerce medium like Amazon. Consumers are confident when information is highly diagnostic [10]. Tversky and Kahneman (1974) [33] have found that the increased availability of reasons for a decision increases the decision maker's confidence. Negative review creates a lack of confidence and it hinders a consumer's purchasing decision. Product maker needs to address this predicament as early as possible so that they can maintain a competitive edge. We termed this kind of congregation of negative sentiment based on a product or the feature of a product as Early Warning Symptoms (EWS). If a product development organisation can find an early workaround within a finite small amount of time, they can find a solution for the created negative sentiment.

Our research work is based on two available rule-based sentiment analyser-VADER and TextBlob. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media [12]. TextBlob is a python based library that does a sentiment analysis based on some predefined rules.

Both of them provide a sentiment score based on sentence sentiment polarity. We will discuss both in the later part.

We are trying to find out which rule-based algorithm has higher accuracy in classifying the negative sentiment. To analyse this, we have extracted more than 66000 mobile phone product review data from Amazon. And applied the feature extraction algorithm like RAKE (Rapid Automatic Keyword Extraction Algorithm) [11] and tuned it in such a way so that it can extract the highest probable uttered phrases. From here, we have extracted the primary keyword features like 'Camera', 'Battery', etc. We did not stop there; we went into a deeper level to find out the most uttered sub-features. We have used the Word2Vec algorithm with CBOW to find out cosine similarities between sub-features and main features. We have picked the sub-feature based on higher cosine similarities. We went into our main corpus of 66000 reviews and tried to find the sentence where this features and sub-features has uttered and extracted those sentences. After that, we did a sentiment analysis of those features and sub-feature associated sentences using TextBlob and VADER. We were able to find out there is a substantial gap between VADER and TextBlob's classification of sentiment polarities based on given sentences. Especially with negative polarity sentiments.

To validate both of their performances, we had used the 'Wisdom of Crowd' [13] philosophy; after using crowdsourcing, we have randomly picked 150 sentences (feature and sub-feature specific) and did crowdsourcing based on 20 individuals. Among these 20 individuals, 10 of them were taken as domain experts, who have prior knowledge of the domain and 10 common people, who are less aware to the technicality of the domain. During our research work, we have seen that the sentiment polarity of a particular sentence varies from person to person. So, we did aggregation of sentiment polarities from each of the predefined 20 members, for each sentence and extracted the final polarity, based on the highest polarity support. As we are following the Wisdom of Crowd model [13] we have considered it as an actual or gold standard. We had evaluated VADER and TextBlob results on these 150 data. The outcome was very interesting. Initially, we wanted to find out relative entropy based on Kullback Leibler (KL) divergence.

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i).\,log\,\frac{p(x_i)}{q(x_i)} \qquad (1)$$

KL divergence represents the distribution of predicted data over actual data. We are considering the outcome from WOC as actual data. The measure q(x) typically represents the outcome of VADER or TextBlob on the approximation of p(x). We had found

DKL(p(R) || q(Result of VADER on X)) = 0.1256

DKL(p(R) || q(Result of TextBlob on X)) = 0.1838

Where X = 150 chosen feedback data, R = Result of WOC on X.

Lower the value of DKL proves that better the class approximation is made by the algorithm based on True distribution. This heuristic demonstrates that VADER performs better in the context of classification than TextBlob. Also, the Confusion matrix shows to find out negative polarity VADER has an f1-score of 0.80, whereas TextBlob has 0.56. It proves VADER can classify negative polarity sentiment based on True data better than TextBlob. The overall accuracy of both Rule-based models are VADER, TextBlob respectively 63.3%, and 41.3% on feedback data.

II. RELATED WORK

Research work related to sentiment analysis on Natural Language Processing has been burgeoning from the recent past, and it will continue to do so. We will try to cover some canonical works pertinent to our research work. There are two ways to extract the sentiment polarities from Natural Language.

A. Lexicon based approach
B. Machine Learning approach

Lexicon based approach involves calculating the sentiment from the semantic orientation of words or phrases that occur in a text [15].

With this approach, a predefined dictionary is required based on positive and negative polarities.

Different methods have been propounded like manual [16] or automatic methods. In a Lexicon-based approach for a sentence, each word has been given a polarity value. These values are predefined in a dictionary. A combination function for calculating the overall average can be used. As per Jurek et al. [17], “Apart from a sentiment value, the aspect of the local context of a word is usually taken into consideration, such as negation or intensification.” This type of Lexicon-based approach also can be called ‘Rule-based’ sentiment analysis.

Jurek et al. [17] had proposed a similar kind of Lexicon-based approach. The central focusing point of the algorithm was sentiment normalisation and an evidence-based combination function, which is mainly used to estimate the intensity of the sentiment rather than positive/negative labels and to support the mixed sentiment classification process.

Gilbert et al. [12] has developed VADER based on this approach. They have combined qualitative and quantitative method to produce an empirically validated, a gold standard sentiment lexicon that is attenuated based on a microblog. After this, they have considered five heuristics like punctuation, capitalisation, degree modifiers, polarity shift due to conjunction, catching polarity negation. These are syntactical conventions that humans use to express sentiment intensity. They have compared its effectiveness with 11 typical states of practices benchmark like LWIC (Linguistic Inquiry and Word Count), ANEW (Affective Norms for English Words), SentiWordNet, General Inquirer, and Machine Learning technique that rely on Naive Bayes, Maximum Entropy, and SVM (Support Vector Machine). As per Gilbert et al. [12], VADER outperforms (f1=0.96) individual human rater (f1=0.84) at correctly classifying the sentiment of the tweet into positive, negative, and neutral polarity classes. VADER uses valance score. Valance score is nothing but a score which is assigned to the word under the experiment employing observation and experiences rather than pure logic. The valence score of VADER is measured on a scale from -4 to +4, where -4 denotes the most ‘Negative’ sentiment, and +4 denotes the most ‘Positive’ sentiment whereas 0 as a neutral sentiment. Apart from the positive, negative, and neutral polarity score, it uses another score, i.e. compound. The compound score of VADER is computed by adding the valence scores of each word in the lexicon, which is adjusted according to the rules. Then it is normalized between -1 (extreme negative) and +1 (extremely positive). As per explanation they have used below formula

$$\left\{ x = \frac{x}{\sqrt{x^2+\alpha}} \begin{matrix} x{=}sum\ of\ valence\ scores \\ of\ constituent\ word \\ \alpha{=}normalization\ constant \\ default\ value\ 15 \end{matrix} \right\} (2)$$

An example of polarity classification by using VADER looks like below :

Sentence= *"The Camera is super cool"*

The polarity score given by VADER for above sentence {'neg': 0.0, 'neu': 0.0 , 'pos': 0.674 , 'compound' : 0.7351}.

Bouazizi and Ohtsuki have proposed SENTA [18] a multiclass based sentiment analysis which was based on the twitter data. Their study was based on multiclass seven sentiments. As per their research, The accuracy achieved for multiclass sentiment analysis was 60.2%.

Another available lexicon-based sentiment analysis model is TextBlob [14]. TextBlob is a package for python. TextBlob uses two parameters; polarity and subjectivity. Polarity exhibits the actual sentiment polarity of the sentence. It varies from -1 to 1 where 0 is neutral. Also, it has a field called subjectivity. Subjectivity refers to how someone’s judgment is shaped by personal opinions and feelings instead of outside influences—the greater the subjectivity stronger the polarity score. When calculating sentiment for a single word, TextBlob uses a sophisticated technique of “averaging”.

An example of polarity classification by using TextBlob looks like below :

Sentence= *"The Camera is super cool”*

The polarity score is given by TextBlob:

sentiment(polarity=0.38542, subjectivity=0.6583333)

In their paper, ‘A comprehensive study on lexicon-based approaches for sentiment analysis’ [19]. Bonta, Kumaresh, and Janardhan have proved VADER is better than TextBlob while classifying sentences based on the gold standard concerning microblogs.

One of the significant advantages of this rule-based or Lexicon-based approach is that they are not domain-specific. Socategorisation of sentences based on polarity is independent. Our research endeavour is based on VADER and Text blob, out of which lexicon-based models can approximate the negative sentiments more accurately.

A manually created lexicon-based approached is laborious and time-consuming. As per Boiy et al. [20], “Machine learning techniques for sentiment classification gain interest because of their capability to model many features and in doing so, capturing context, their more easyadaptability to changing input, and the possibility to measure the degree of uncertainty by which a classification is made”. Naive Bayes Classifier [22] [23], Maximum Entropy, Support Vector Machine [24] are different machine learning algorithm which can be espoused to perform sentiment analysis.

To apply these techniques, we need to train our model on domain-specific data. So a domain-specific algorithm won't work well with asymmetric domains. Also, it can be computationally expensive when we are training with extensive data set to get the best result. While doing a multi-domain sentiment analysis using Machine Learning, A resource domain which has plentiful of domain sentiment knowledge can transfer its intelligence to a domain which will have less resource, related to conducting sentiment analysis but this provokes few problems like feature mismatch, polarity divergence, polysemy, and sparsity [25] [26].

III. METHODOLOGY

As we have said, we have analysed mobile phone product review data from amazon india for our experiment. So, to prove our hypothesis that vader can classify negative polarity sentiment better then textblob, we took the below methodology.

1. Data Collection
2. Preprocessing and Feature Extraction
3. Sentence extraction based on features and sub-features
4. Classifying sentences sentiment polarity using VADER or TextBlob

Our data collection process is divided into two sections as we are not only extracting the key features from the mobile phone review database of Amazon India but also we have created the Gold standard by crowdsourcing based on 'Wisdom of Crowd'. Let us first discuss Gold standard creation by extracting value from a group of people.

A. Data Collection & Crowd Sourcing

We have done crowdsourcing based on some extracted feature-based sentences (count 150) from our main corpus. From there, we had applied the Wisdom of Crowd principle proposed by Surowiecki [13], and we will consider these outcomes as a Gold standard. As per Surowiecki, "collective knowledge of a group of people as expressed through their aggregated opinions can be trusted as an alternative to an expert's knowledge". He has tried to justify his explanation by stating, "consider Google, the astonishingly successful internet search engine, why does Google work so well?" As per him, 'it's built on the Wisdom of Crowd'. We have picked 20 individual who does not have any connection with each other; 10 of them are from the topic domain field. We termed them as 'Experts' and another 10 randomly chosen who do not have any prior knowledge about the domain. We call them 'Common'. The reason for picking the 'Expert' is, as Brunswik famously argued, that decision-makers need to be presented with representative stimuli to assess how well they use information [28]. We have made sure the response from the 20 individual members is mutually exclusive. As we know that judgement based on sentence sentiment polarity may differ from person to person as the research work of Wilson et al. [27] has suggested, "the contextual polarity of the phrase in which a particular instance of a word appears may be quite different from the word's prior polarity is not easy to classify based on sentiment polarity like positive words are used in phrases expressing negative sentiments, or vice versa. Also, quite often, words that are positive or negative out of context are neutral, meaning they are not even being used to express a sentiment". During our experiment with crowdsourcing, we have observed that below-mentioned sentence:

"But there is some sort of lag in my handset. Perhaps it could be just my phone."

Out of 20 members, 11 has said it conveyed a negative sentiment, and 9 have conveyed its neutral sentiment. This sort of ambiguity of sentiment polarity remains with polarity shifts due to conjunction or punctuation. For our previous example, probably the obscurity of judgement happened because of words like *'but', 'perhaps', 'could be'*. Here is another example that can create convictional ambiguity, *'The weather is not bad but bearable'*. We have made sure that all the decisions on 150 review data remain mutually exclusive because we know that [28] people are egocentric in their use of judgments. They rely too much on their judgments and miss the opportunity to learn from others. So, based on this principle, we can extract exact data from most of the people. So, a greater number of samples with mutual exclusiveness will be helpful to get the near actual data. Also, from crowdsourcing data, we have extracted the Sentiment Polarity Support (SPS).

$$SPS = \frac{\substack{Highest\ number\ of\ polarity\ count\ for \\ particular\ sentence}}{Population\ size} \quad (3)$$

Also, as [30] a variety of feedback gives proper results when we are aggregating sentiment polarity to get the maximum value for the same. Based on this, we have kept 'Common' people who do not have the domain knowledge. A problem we have faced while doing opinion aggregation for some sentences where two different sentiment polarity had the same maximum value, this is a tie situation.

In this case, we have given priority to the 'Expert's' sentiments. Let's consider we have a Crowd of 'A' people, where 'B' is the sub crowd so B ⊆ A. As we have said, earlier ‘Experts’ are blessed with domain knowledge, in this case, which is ‘Mobile phone’ as they are associated with the mobile industry. The selection of experts was made in such a way; each ‘Expert’ will have a minimum of 5 years of experience in the mobile domain. The primary objective to have an ‘Expert’ sub crowd is to get better Accuracy and Efficiency. ‘If collecting a judgment is costly, then alternatives that require fewer judgments are more efficient' [30]. And if smarter sub-crowds or ‘Experts’ exist, it may be possible to attain higher accuracy than conventional WOC aggregation techniques. Efficient and accurate crowdsourcing of judgments should be welcomed in the fields of on-line polling, prediction, and forecasting. In their paper 'The Wisdom of Smaller, Smarter Crowds' [31] Goldstein et al. has proved that improved predictions may be obtained by giving high weights to the opinions of those who have demonstrated skill in the past. Because data sets with individual-level records of human performance are becoming increasingly available, the future may hold more occasions where improvement can be achieved upon crowd predictions by identifying and tapping into the Wisdom of smart sub-crowds. So, complying with it in case of a tie, we have to use a tie-breaker rule by giving priority to the judgment of an expert who has more experience than other experts for respective fields.

B. *Feature Extraction from Amazon Review data*

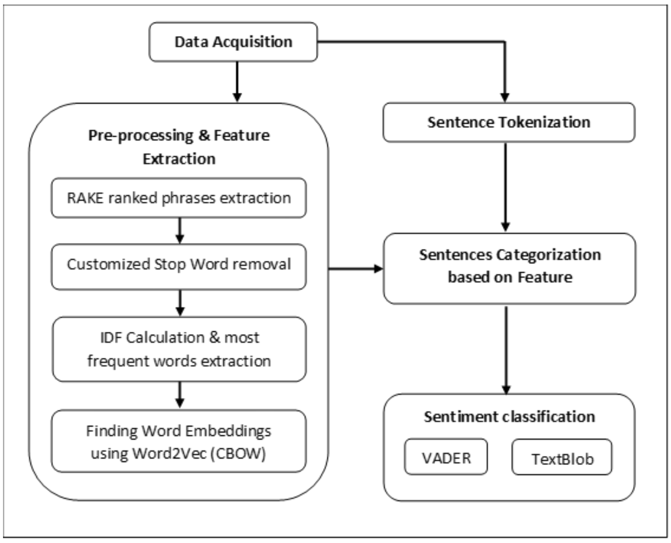

Data Acquisition
Pre-processing & Feature Extraction
RAKE ranked phrases extraction
Customized Stop Word removal
IDF Calculation & most frequent words extraction
Finding Word Embeddings using Word2Vec (CBOW)
Sentence Tokenization
Sentences Categorization based on Feature
Sentiment classification
VADER
TextBlob

Fig.1. Methodology for Key feature extraction from Review Data

We have used the computer programming language Python 3.8 on Windows 10 Home (64 bit) and different libraries for the collection and extraction of the features. Some of the libraries used are Pandas, NumPy, NLTK, Spacy, Gensim, Scikit-learn, etc. The hardware which we have used has Intel i5 processor 2.40 GHz along with 4GB RAM.

For data collection purposes, we had used web scraper to extract mobile product review data. We had collected more than 66000 reviews. Online reviews are immanently temporal, short, and noisy about the evanescent subject that users create in bursts and ultimately faded away, unlike a formal document that is well structured long-lived, less noisy. That is why the cleaning of online review data takes a colossal effort. So here are some steps that we have done. First, we have tried to convert our entire text to lowercase then we have tried to contain misspellings. After that, we have conducted sentence tokenisation followed by a special Key-feature extraction using unsupervised machine learning algorithm RAKE(Rapid Automatic Keyword Extraction). As per the developer of RAKE [11], ‘It is based on our observation that keywords frequently contain multiple words but rarely contain punctuations or stopwords or other words with minimum lexical meanings’. RAKE splits the text at phrases delimiters and stopwords to candidate expression so the candidate key phrases would remain intact. Once the candidate text has been split based on stopwords, delimiter, and content words, the algorithm, creates a word of co-occurrence. Each row will show the number of times a given content word co-occurs with every other content word in the candidate phrase. RAKE followsa logic of ‘n-gram’, so after using RAKE between multiple grams, it will keep only frequent phrases.

After that, we have tokenised the phrases by word. We have defined a customised stopword list that contains words that are not features on a specific domain like *‘Samsung’, ‘Oppo’, ‘Vivo’,* etc.. From our word tokenised corpus, we have removed those customised stopwords. Then, we havecalculated the IDF (Inverse Document Frequency) value of the word tokens.IDF value is nothing but

$$idf_j = log\left[\frac{n}{df_j}\right] \quad (4)$$

Where $idf_j$is Inverse Document Frequency of the word *j*, *n* is the number of document &$df_j$is the number of documents with the term *j* in it.

IDF is based on Zipf's law, which states that frequencies of certain events are inversely proportional to their rank.Thus, the most common word in English *‘the’*occurs about one-tenth of the time in a typical text; the next most *‘of’*, occurs about one-twentieth of the time; and so forth.Now *‘The’, ‘of’* are the stopwords which are being eliminated by RAKE. So whatever is left, if we remove some domain-specific keywords, we will get the most frequently uttered words. This is the way to get the primary features.

Table 1. Some outcomes of Keywords and their respective IDF Values.

Keyword	IDF
camera	3.992705829
battery	4.277344163
display	5.260289819
performance	5.310988583
charging	5.616157986
.	
.	
Zeiss	11.12958673

We went to find out what are the sub-features the reviewers are mostly talking. Here we have used Word2Vec [32] algorithm CBOW model to find out what are the secondary words associated with our primary words are having the highest cosine similarities. Cosine similarities measure the similarity between two vectors of an inner product space.

$$similarity(A,B) = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} (5)$$

Where A & B are vectors, and with this parameter, we are finding out how close they are. The greater the value, they are more relative to each other and vice versa.

Table 2. Primary words with their relative secondary words

Primary Keyword	Secondary Keywords	Cosine Similarity
camera	selfie	0.5531975
camera	photo	0.6227054
camera	macro	0.6767875
display	resolution	0.67543
performance	ram	0.8571528
performance	speed	0.7564032
fingerprint	recognition	0.8924087

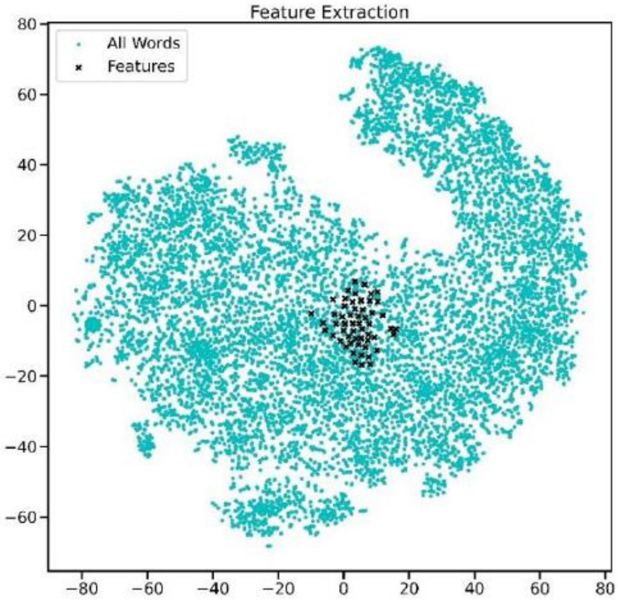


Fig.2. Two-dimensional representation of all Words and Keywords (t-SNE normalisation of Word2Vec)

t-SNE means t distribution-Stochastic Neighbourhood Embedding. To visualise multi-dimensional nonlinear vector in smaller dimensions (mostly 2D or 3D), we use t-SNE.

C. Sentence extraction based on features and sub-features

Once we extracted the primary and secondary features, we went back to our main corpus to find out what are sentences associated with those extracted features. These sentences are probable sentences for which we are looking to perform sentiment analysis based on our Lexicon-based algorithms VADER and TextBlob. We have extracted those sentences.

D. Sentiment polarity detection using VADER or TextBlob

On those extracted sentences, we had performed Lexicon-based sentiment analysis using TextBlob and VADER. In the later part, we will show how we are comparing TextBlob result with VADER's.

IV. RESULT DISCUSSION

We are experimenting with two Lexicon-based sentiment analyser VADER and TextBlob, based on negative sentiment in product review data. We have derived a mathematical representation of our work.

As per Bing Liu [3], we can describe the opinion of a person on a product as below:

$O: (e_i, a_{ij}, s_{ijkl}, h_k, t_l)(6)$

Where O is an opinion, e_i is the name of the entity for which the opinion has been given, a_{ij} is the features of the entity. A person can give an opinion on the features of an entity. h_k is the opinion holder. t_l is the time when the opinion has been given. s_{ijkl} is the sentiment on a_{ij} for entity $e$igiven by h_k on time t_l. We are assuming some extra parameters like T_{ijkl} is nothing but text or document on which h_k will provide the opinion. As per Bing Liu [3] , we can consider three levels while doing sentiment analysis like:

Document-level: The task at this level is to classify a whole opinion based on the document. We have observed that while giving an opinion, people tend to provide a collective opinion based on different entities of a domain. Like while providing an opinion on a mobile, people tend to share an opinion on a distinct entity like Camera, Fingerprint Sensor, Sound, etc. But cumulative sentiment polarity display on Document-level polarity like Amazon overall review for a feed which is categorised by Amazon as 1 star, 2 stars…5 stars.

Sentence Level: The task of this level goes to the sentences and determine the subjective classification, which distinguished sentences (called objective) that express the factual information from the sentences that express subjective views and opinion. Subjectivity is not analogous to sentiment, as many objective sentences can imply the opinion.

Entity and Aspect level: Document and Aspect level are what exactly people like and what they don't. The aspect level performs a finer-grained analysis. The aspect level can be called the feature level.

While doing the analysis, we have decided to consider the aspect level; for this reason, we went through the secondary feature category while extracting the features. So that we can get the perfect view of what the customer is speaking, we termed it as $T_{ijkl,}$ which implies that text or document on which h_k will provide its fine-grained statement about feature a_{ij}. Now a reviewer won't directly say whether this statement conveys a positive or negative or neutral sentiment. So, we can consider this sentiment polarity as a latent variable s_{ijkl}.

We can say that $s_{ijkl}(T_{ijkl}) \in \{1, 0, -1, \Phi\}$ Where 1 conveys positive sentiment, -1 conveys negative, 0 neutral, and Φ are those for which the reviewer has not commented anything during his review.

We can consider e_i, a_{ij} these are the domain-centric, and we can term it as D_g:(e_i, a_{ij}) It states the domain information.

Now, as per our experiment, we are considering the crowdsourcing as a gold standard, so to get the value, we will consider maximum support sentiment after aggregating the result for each sentence. We must emphasise the feelings of the people on a text or product as that determines the sentiment.

As per our research, we can evaluate an opinion as below:

$$O_e : (O, \theta_p, S_{p\theta}, t_{\theta u}) \qquad (7)$$

θ_p is the evaluator. It can be an algorithm or it can be some collective effort of a human being.

θ_p can be described as $\theta_p : (\theta_{pA}, \theta_{pC})$ Where $\theta_{pA}$is nothing but evaluating algorithm and θ_{pC} is nothing but the crowdsourcing gold standard based on 'Wisdom of Crowd'.

$S_{p\theta}$ is sentiment polarity given by θ_p and $S_{p\theta} \in$ {1, 0, -1} Where 1 conveys positive sentiment, -1 conveys negative, 0 neutral. Let's consider $S_{pA\theta}$ is the sentiment polarity obtained by applying the algorithm. $S_{pC\theta}$ is the sentiment polarity obtained from crowdsourcing. So, we can write

$$S_{pA\theta} = \theta_{pA}(T_{ijkl}) \text{ and } S_{pC\theta} = \theta_{pC}(T_{ijkl}) \qquad (8)$$

$T_{\theta u}$is a time factor as we know in the field of machine learning, we always try to approximate the result, so we need to attune our algorithm over time to reach a 100% success rate. Also, the sentiment of people on a product feature changes over time so doing crowdsourcing; time makes an impact while considering the Wisdom of Crowd.

When comparing the success rate of two or more algorithms based on the outcome of 'Wisdom of Crowd' (Gold standard) if we can prove,

$$\theta_{pA}(T_{ijkl}) = arg \max_{S_{pC\theta}} \theta_{pc}(T_{ijkl}) \; \forall \; T_{ijkl} \qquad (9)$$

For overall performance calculation, we can consider the cardinality of True Positive cases from the True Positive Set detected by the algorithm for particular sentiment classes divided by total cardinality of particular sentiment polarity detected by crowdsourcing using Wisdom of Crowd. It is nothing but the accuracy as per the confusion matrix.

For our case,

θ_{pA}= {VADER, TextBlob}

θ_{pC}= {Maximum Collective opinion polarity based on crowdsourcing on a sentence}

We are trying to validate,

θ_{pA}= VADER >θ_{pA}= TextBlob when $S_{pA\theta} \in$ {-1}

OR

θ_{pC}= VADER < θ_{pA}= TextBlob when $S_{pA\theta} \in$ {-1}

Which of the above condition is true?

Algorithm with higher cardinality of the True positive cases will have an edge on others.

We have previously given a few facts about VADER and TextBlob. Now, lets us share some more insight into these two Sentiment analysers.

VADER: As per Gilbert et al. [12], This tool analyses sentence sentiment based on certain factors like Punctuation, Capitalisation, Degree modifier, Conjunction, and Preceding Trigram. There are more than 7500 lexical features with validated valance score that indicates sentiment polarity and sentiment intensity which ranging from -4 to 4. Like the sentiment rating of words, *'good'* is 1.9 and *'sucks'* -2.2.

The compound score is the significant parameter for sentiment polarity detection

Positive Sentiment: Compound score >= 0.05

Neutral Sentiment: Compound Score >-0.05 and <0.05

Negative Sentiment: Compound Score <=-.05

The developer of VADER has built this tool, especially considering the Twitter feed, which will consist maximum of 280 characters. And it has shown an f1 score of 0.96.

TextBlob: Each word in lexicon has score based on

1. polarity: negative vs. positive (-1.0 to +1.0)

2. subjectivity: objective vs. subjective (+0.0 to +1.0)

3. intensity: modifies next word? (x0.5 to x2.0)

4. confidence: looks for correct spelling (0 to 1)

Here are some rules for the subjectivity lexicon for an adjective used by TextBlob:

Adjectives have a polarity (negative/positive, -1.0 to +1.0) and a subjectivity (objective/subjective, +0.0 to +1.0).

The reliability specifies if an adjective was hand-tagged (1.0) or inferred (0.7).

Words are tagged per sense, e.g., ridiculous (pitiful) = negative, ridiculous (humorous) = positive.

The WordNet id refers to the WordNet3 lexical database for English.

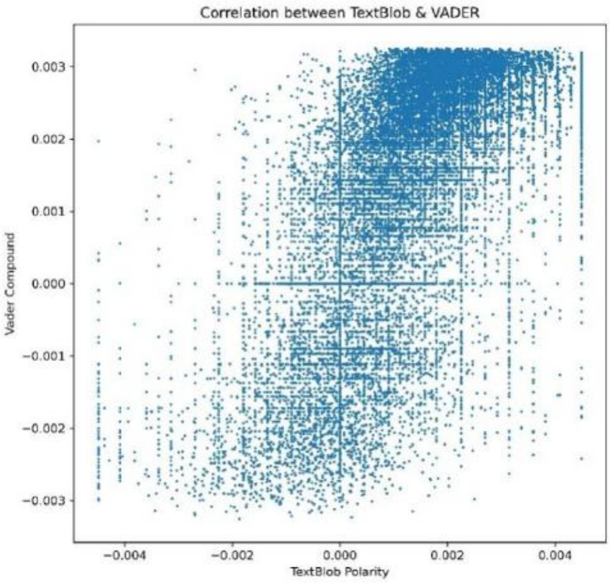
The part-of-speech tags (pos) use the Penn Treebank || tag set: NN = noun, JJ = adjective, ...

For English movie reviews (Pang & Lee polarity dataset v2.0)

Below is how the XML lexicon dictionary looks like:

<word form="slow" cornetto_synset_id="n_a-516764" wordnet_id="a-00980527" pos="JJ" sense="not moving quickly" polarity="-0.2" subjectivity="0.1" intensity="1.0" confidence="0.9" />

Now as we can see that sentiment polarity can be detected by VADER using the parameter 'Compound' and for TextBlob its 'Polarity'. Both of these values are having a range from -1 to 1.So, to check what is the correlation between VADER and TextBlob, we have done a correlation analysis on the same dataset below are the findings:



Fig,3. Correlation between TextBlob & VADER

$$r = 0.63326562$$

$$r^2 = 0.40102534$$

r is the coefficient of correlation, and r^2 is the coefficient of determination. It indicates the extent to which the variation in others explains variation in one variable. So, a 40.10% variation is predictable among VADER and TextBlob and remaining due to other unknown factors. We will review that.

In the previous graph (Figure 3), we can observe the neutrally classified statement has created four segments. In the case of the 1st, & 3rd quadrant both the algorithms have reached an agreement. But in the case of the 2nd& 4th quadrant, there is a mismatch especially for the 4th quadrant, which has more contradictory data, it belongs to Positive as per TextBlob and Negative as per VADER. We will discuss this later why this disparity of data is happening.

Our primary aspiration is to find out which algorithm performs better with negative sentiment detection. During crowdsourcing with our 150 data, we have intentionally kept more than 55% sentences which are having a proclivity towards negative polarity as per us. After doing aggregation based on 'Wisdom of Crowd' we have discovered that still, the number of negative sentences are closure towards our pre-assumption; which is close to 55%. As before we have said based on our 150 samples of data, we have created a gold standard. Now to find out how VADER & TextBlob are performing w.r.t. Our gold standard we have created a confusion matrix; our observation was the overall performance of VADER is better than TextBlob. Whereas for negative sentiment classification VADER is far streets ahead of TextBlob.

The first paragraph under each heading or subheading should be flush left, and subsequent paragraphs should have a five-space indentation. A colon is inserted before an equation is presented, but there is no punctuation following the equation. All equations are numbered and referred to in the text solely by a number enclosed in a round bracket (i.e., (3) reads as "equation 3"). Ensure that any miscellaneous numbering system you use in your paper cannot be confused with a reference [4] or an equation (3) designation.

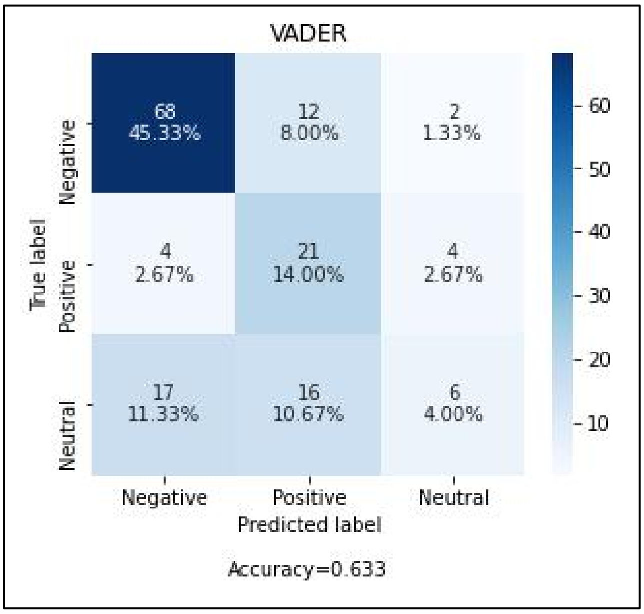


Fig.4. Confusion Matrix (Crowdsourcing & VADER)

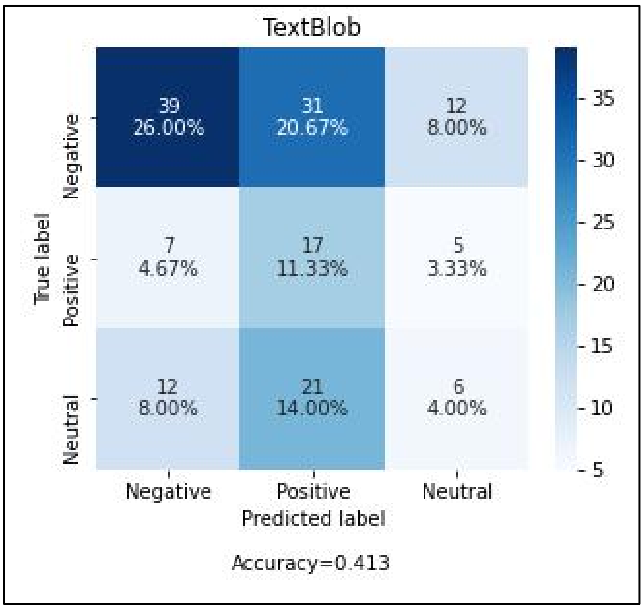


Fig.5. Confusion Matrix (Crowdsourcing & TextBlob)

Table 3. Classification Report of VADER & TextBlob w.r.t. the gold standard (WOC)

		Precision	Recall	f1 score
VADER	**Negative**	0.76	0.83	0.8
	Positive	0.43	0.72	0.54
	Neutral	0.5	0.15	0.24
TextBlob	**Negative**	0.67	0.48	0.56
	Positive	0.25	0.59	0.35
	Neutral	0.26	0.15	0.19

From the above result, we can prove that based on classification VADER outperforms TextBlob and especially while detecting negative polarity sentiments. Based on the f1 score value for negative polarity detection, TextBlob is 0.56, whereas VADER is 0.80, which emphasise VADER’s ascendancy over TextBlob on accurate negative sentiment classification.

	f1 score
Vader	0.80
TextBlob	0.56

We went into further deep to find out, why TextBlob unable to exhibit accuracy with negative sentences? Below are our findings.

let’s consider a word: *'good'*

TextBlob outcome:

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.66.1

Sentiment(polarity=0.7, subjectivity=0.6000000000000001)

let's add a few more words before *'good'*: *"The Camera is good"*

TextBlob outcome:

Sentiment(polarity=0.7, subjectivity=0.6000000000000001)

We can observe there is no change is polarity and subjectivity values for the above cases.

Now let's add a negation before the word good so our sentences will be like: *"The camera is not good"*

TextBlob outcome:

Sentiment(polarity=-0.35, subjectivity=0.6000000000000001)

'Subjectivity' remains the same but 'Polarity' is showing as a negative value. How has it happened? In case of a negation follows by a word which infers some polarity, TextBlob simply multiplies (-0.5) with the polarity score of the next word. So, in our above example, *'good'* has a polarity of 0.7, and when we add *'not'* before *'good'* it will multiply (-0.5) with (0.7) and display the polarity score as (-0.35). There will be no change in subjectivity value. This rule is fixed for TextBlob.

The problem will happen when a word can represent in two-part of speech like *'slow'* it can be a noun that can be an adjective. TextBlob will consider this as an adjective so it will display as

Sentiment(polarity= -0.30000000000000004, subjectivity=0.39999999999999997).

But once we add a negation like *'no'* before *'slow'* and some other words after *'slow'* then The polarity score becomes faulty, like for sentence: *'no slow motion camera'* it simply multiplies (-0.5) with (-0.3) and the polarity score looks like:*Sentiment(polarity=0.15000000000000002, subjectivity=0.39999999999999997).*

So, negative sentiment has been marked as a positive sentiment. As negative signs are cancelled out during multiplication of two negative values. There are more sentences like these. Few are given below:

Sentences

"Network also not so much great"

"Not the best Face Unlock in this price"

"So while placing the finger it doesn't feel good."

with above example let's see how VADER performs for the word *'slow'*, VADER shows:

{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

It termed it as neutral sentiment and which is convenient.

For sentence *'No slow motion camera'* it displays

{'neg': 0.423, 'neu': 0.577, 'pos': 0.0, 'compound': -0.296}

which is nothing but a negative sentiment. This result is also convenient.

It proves that TextBlob is unable to perform well in some cases whenever a negation is followed by a negative adjective. In contrast, VADER shows consistent performance.So, this is the reason if we compare VADER and TextBlob based on detecting negative sentiment polarity. Vader outperforms TextBlob based on accuracy and f1 score.

During our work, we also did another experiment based on Crowdsourcing result. We had taken those individual sentences where Sentiment Polarity Support (SPS) given by the crowd is 70% or more than that. After that, we have checked the output of VADER and TextBlob. VADER performances improve with the f1 score of detecting negative sentiment is 0.89 and overall accuracy is 78% here still, TextBlob lags at an f1 score of 0.63 and overall accuracy of 51%. It proves that when people are sure about a sentiment polarity. We can expect the best result from VADER.

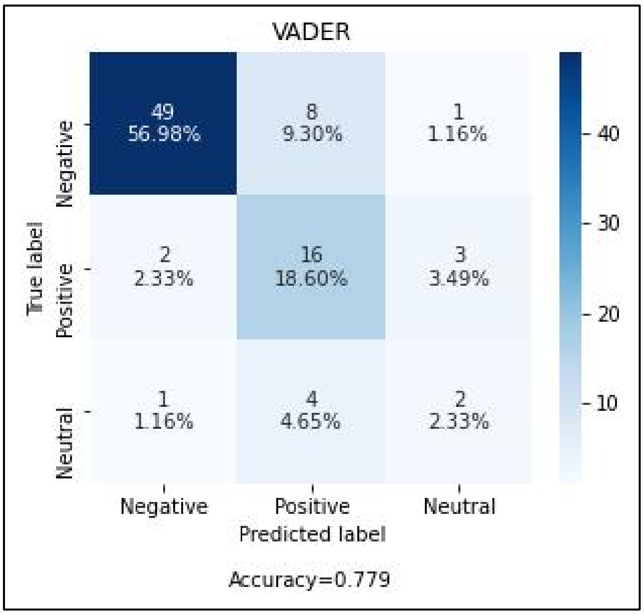


Fig.6. Confusion Matrix considering 70% support from Crowdsourcing with VADER

	f1 score
Vader	0.89
TextBlob	0.63

f1 score of VADER & TextBlob considering negative sentiment.

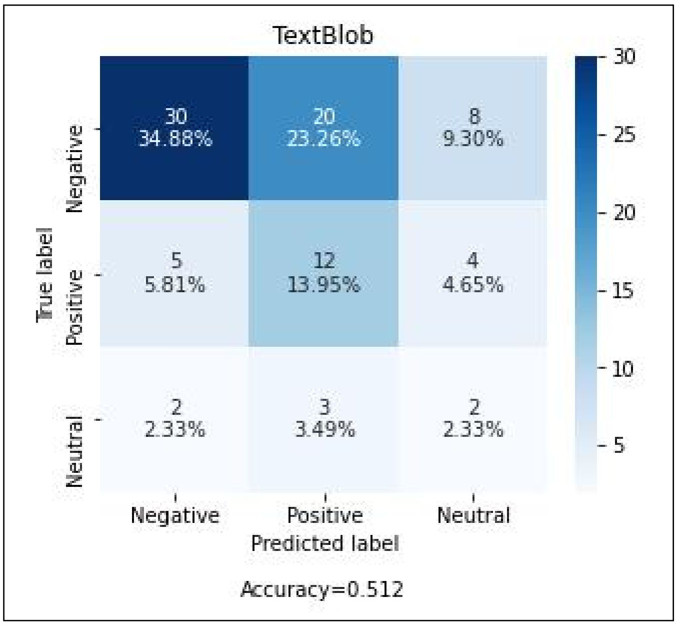


Fig.7. Confusion Matrix considering 70% support from Crowdsourcing with TextBlob

We have sometimes seen VADER and TextBlob outperforms human. Intentionally we have kept a statement that is not complete like *"Stock camera is goo too."* VADER and TextBlob both have classified correctly as 'neutral'. This statement bears no meaning concerning sentiment polarity. But most of the members in our crowdsourcing by their pre-conviction have marked it as a positive sentiment. Even the maximum aggregated sentiment value for this statement was positive. For them, it was *"Stock camera is good too."* Even an expert can do these type of mistakes. But VADER and TextBlob circumvent this challenge. It also proves the necessity of a computer-based algorithm.

During this analysis, we need to keep in mind that VADER has been specially developed for Twitter-like micro Blogs which will have 280 characters. If we can have a sentence with 280 characters, VADER shows good result this is the reason if we extract sentences based on the primary and secondary both keywords we will have sentences closer to 280 characters each. In our case below graph display the character distribution of our main corpus.

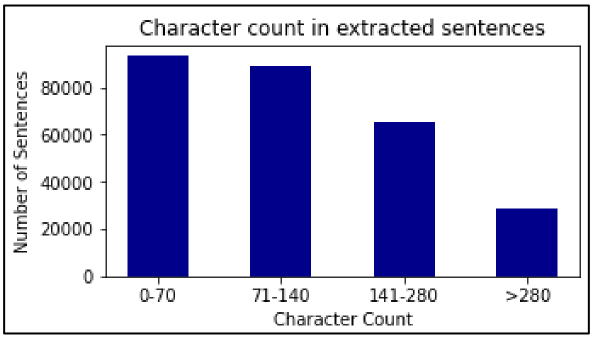


Fig.8. Character count distribution

We have observed that while detecting positive & neutral polarity sentiments, VADER also outperforms TextBlob.

	Positive	Neutral
Vader	0.54	0.24
TextBlob	0.35	0.19

Above is the calculated f1 score of positive & neutral sentences.

But still, for VADER negative sentiment classification not only have a higher edge than its own positive and neutral sentiment classification but also from TextBlob overall.

V. CONCLUSION

Customer review database is a gold mine for an organization. It protrudes paramount value to know the customer and the product. Earlier, we had expounded how negative sentiment can behest not only the product but also the brand value of an organization. Hence, curbing the negative sentiment on a product is cardinal for an organization. From the above explanations, we have demonstrated VADER sovereignty lies in classifying Negative sentiment over TextBlob. Our result displays 0.80 f1 scores in detecting negative sentiment with VADER. In comparison, TextBlob's f1 score lies at 0.56 in detecting negative sentiment. Also, our experiment proclaims that VADER is having an edge in overall sentiment classification then TextBlob. Sentiment classification accuracy for VADER is 63.3%, and for TextBlob it is 41.3%. These data are adequate to prove VADER'S supremacy over TextBlob.

VI. FUTURE WORK

During crowdsourcing, we have seen that there is a possibility of having some spam feedback classification provided by a small group of people. If we can detect those sentiments, we may expect an improvement in overall performance. We will try to contemplate the above cases and will try to come out with some solution in the future. Also, we will try to have a more extensive database for our crowdsourcing for better performance evaluation.

REFERENCES

[1] Z. Hu, J. Hu, W. Ding and X. Zheng, "Review Sentiment Analysis Based on Deep Learning," 2015 IEEE 12th International Conference on e-Business Engineering, Beijing, 2015, pp. 87-94, doi: 10.1109/ICEBE.2015.24.

[2] Ballestar, M.T.; Cuerdo-Mir, M.; Freire-Rubio, M.T. The Concept of Sustainability on Social Media: A Social Listening Approach. Sustainability 2020, 12, 2122.

[3] Sentiment Analysis and Opinion Mining Bing Liu
Synthesis Lectures on Human Language Technologies 2012 5:1, 1-167

[4] R. K. Bakshi, N. Kaur, R. Kaur and G. Kaur, "Opinion mining and sentiment analysis," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 452-455.

[5] Müller, Sune & Holm, S.R. & Søndergaard, Jens. (2015). Benefits of Cloud Computing: Literature Review in a Maturity Model Perspective. Communications of the Association for Information Systems. 37. 10.17705/1CAIS.03742.

[6] Godbole, Namrata & Srinivasaiah, Manjunath & Skiena, Steven. (2007). Large-Scale Sentiment Analysis for News and Blogs. ICWSM 2007 - International Conference on Weblogs and Social Media.

[7] Tsugawa, Sho & Ohsaki, Hiroyuki. (2015). Negative Messages Spread Rapidly and Widely on Social Media. 151-160. 10.1145/2817946.2817962.

[8] Baumeister, R. F., Bratslavsky, E., Finkenauer, C., & Vohs, K. D. (2001). Bad Is Stronger Than Good. Review of General Psychology, 5(4), 323-370. https://doi.org/10.1037/1089-2680.5.4.323

[9] Rozin, Paul & Royzman, Edward. (2001). Negativity Bias, Negativity Dominance, and Contagion. Personality and Social Psychology Review. 5. 10.1207/S15327957PSPR0504_2.

[10] Mudambi, Susan & Schuff, David. (2010). What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com.. MIS Quarterly. 34. 185-200. 10.2307/20721420.

[11] Rose, Stuart & Engel, Dave & Cramer, Nick & Cowley, Wendy. (2010). Automatic Keyword Extraction from Individual Documents. 10.1002/9780470689646.ch1.

[12] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[13] James Surowiecki. 2005. The Wisdom of Crowds. Anchor.

[14] Steven Loria,'textblob DocumentationRelease 0.16.0'

[15] Taboada, Maite & Brooke, Julian & Tofiloski, Milan & Voll, Kimberly & Stede, Manfred. (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics. 37. 267-307. 10.1162/COLI_a_00049.

[16] Tong, R.M. (2001) An Operational System for Detecting and Tracking Opinions in On-Line Discussion. Proceedings of SIGIR Workshop on Operational Text Classification.

[17] Jurek, A., Mulvenna, M.D. & Bi, Y. Improved lexicon-based sentiment analysis for social media analytics. Secur Inform 4, 9 (2015).

[18] Bouazizi, Mondher & Ohtsuki, Tomoaki. (2017). A Pattern-Based Approach for Multiclass Sentiment Analysis in Twitter. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2740982.

[19] Bonta, Venkateswarlu & Kumaresh, Nandhini & Janardhan, N.. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. 1-6.

[20] Boiy, Erik & Moens, Marie-Francine. (2009). A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts. Inf. Retr.. 12. 526-558. 10.1007/s10791-008-9070-z.

[21] Polanyi L., Zaenen A. (2006) Contextual Valence Shifters. In: Shanahan J.G., Qu Y., Wiebe J. (eds) Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series, vol 20. Springer, Dordrecht. https://doi.org/10.1007/1-4020-4102-0_1

[22] H. Parveen and S. Pandey, "Sentiment analysis on Twitter Data-set using Naive Bayes algorithm," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, 2016, pp. 416-419, doi: 10.1109/ICATCCT.2016.7912034.

[23] Kang, Hanhoon & Yoo, Seong & Han, Dongil. (2012). Senti-lexicon and improved Na??ve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications. 39. 6000-6010. 10.1016/j.eswa.2011.11.107.

[24] K. Mouthami, K. N. Devi and V. M. Bhaskaran, "Sentiment analysis and classification based on textual reviews," 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2013, pp. 271-276, doi: 10.1109/ICICES.2013.6508366.

[25] Bollegala, T. Mu and J. Y. Goulermas, "Cross-Domain Sentiment Classification Using Sentiment Sensitive Embeddings," in IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 2, pp. 398-410, 1 Feb. 2016, doi: 10.1109/TKDE.2015.2475761.

[26] Parvati Kadli and Vidyavathi B M.. Cross Domain Sentiment Classification Techniques: A Review. International Journal of Computer Applications 181(37):13-20, January 2019.

[27] Recognising Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis Theresa Wilson author Janyce Wiebe author Paul Hoffmann author 2009 text Computational Linguistics continuing periodical academic journal wilson-etal-2009-articles 10.1162/coli.08-012-R1-06-90

[28] Mannes, A. E., Larrick, R. P., & Soll, J. B. (2012). The social psychology of the Wisdom of crowds. In J. I. Krueger (Ed.), Frontiers of social psychology. Social judgment and decision making (p. 227–242). Psychology Press.

[29] Durward, D., Blohm, I. & Leimeister, J.M. Crowd Work. Bus Inf Syst Eng 58, 281–286 (2016). https://doi.org/10.1007/s12599-016-0438-0

[30] Herzog, S. M., & Hertwig, R. (2014). Think twice and then: Combining or choosing in dialectical bootstrapping? Journal of Experimental Psychology: Learning, Memory, and Cognition, 40(1), 218–232. https://doi.org/10.1037/a0034054

[31] Goldstein, Daniel & McAfee, Randolph & Suri, Siddharth. (2014). The Wisdom of smaller, smarter crowds. EC 2014 - Proceedings of the 15th ACM Conference on Economics and Computation. 10.1145/2600057.2602886.

[32] arXiv:1301.3781 [cs.CL]

[33] Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. Science, 185, 1124-1131.