IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 23, 2019, accepted August 11, 2019, date of publication August 13, 2019, date of current version August 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935200

Mapping Consumer Sentiment Toward Wireless Services Using Geospatial Twitter Data

WEIJIE QI[1,2], ROB PROCTER[3,5], JIE ZHANG[2,4], AND WEISI GUO[1,5], (Senior Member, IEEE)
[1]School of Engineering, University of Warwick, Coventry CV4 7AL, U.K.
[2]Ranplan Wireless, Papworth Everard, Cambridge CB23 3UY, U.K.
[3]Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.
[4]Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 4BP, U.K.
[5]Alan Turing Institute, London NW1 2DB, U.K.

Corresponding author: Weisi Guo (weisi.guo@warwick.ac.uk)

This work was supported in part by the U.K. Research and Innovation and Ranplan Wireless under Grant KTP10734, and in part by the EC H2020 under Grant 778305.

ABSTRACT Hyper-dense wireless network deployment is one of the popular solutions to meeting high capacity requirement for 5G delivery. However, current operator understanding of consumer satisfaction comes from call centers and base station quality-of-service (QoS) reports with poor geographic accuracy. The dramatic increase in geo-tagged social media posts adds a new potential to understand consumer satisfaction towards target-specific quality-of-experience (QoE) topics. In our paper, we focus on evaluating users' opinion on wireless service-related topics by applying natural language processing (NLP) to geo-tagged Twitter data. Current generalized sentiment detection methods with generalized NLP corpora are not topic specific. Here, we develop a novel wireless service topic-specific sentiment framework, yielding higher targeting accuracy than generalized NLP frameworks. To do so, we first annotate a new sentiment corpus called SignalSentiWord (SSW) and compare its performance with two other popular corpus libraries, AFINN and SentiWordNet. We then apply three established machine learning methods, namely: Naïve Bayes (NB), Support Vector Machine (SVM), and Recurrent Neural Network (RNN) to build our topic-specific sentiment classifier. Furthermore, we discuss the capability of SSW to filter noisy and high-frequency irrelevant words to improve the performance of machine learning algorithms. Finally, the real-world testing results show that our proposed SSW improves the performance of NLP significantly.

INDEX TERMS Wireless, quality of experience, natural language processing, social media data, consumer.

I. INTRODUCTION

A. BACKGROUND

Hyper-density HetNets has been widely recognized as one of the popular solutions for future 5G networks due to their performance in capacity enhancement and blackspot coverage, especially in urban and indoor environments [1]. The constituent small cells are considered as the key to realize millimeter-wave beamforming [2]. However, the practical deployment of small cells is still limited because of the lack of high spatial resolution traffic demand and consumer-centric Quality of Service (QoS) and Quality of Experience (QoE) data. Therefore, due to the small coverage area of small cells, an automated system to mine and analyze high spatial resolution consumer QoE information is needed.

The associate editor coordinating the review of this manuscript and approving it for publication was Mubashir Husain Rehmani.

Twitter is becoming a popular social network platform in recent years – by the year of 2018, the total number of monthly active Twitter users has reached 330 million and total number of tweets sent per day is over 500 million [3]. This penetration has made Twitter a valuable resource for analyzing public opinion on popular daily life topics: the conventional topic may be shopping, politics and marketing. Opinion or sentiment mining is an important area of research and commercial application [4], with topics ranging from political forecasting [5], consumer opinion on new products [6], public order [7], and organization cohesion [8]; but the accuracy is often hindered by ambiguity and the use of non-standard language and orthography in tweets [4].

B. RELATED WORKS

In recent years, statistical analysis of Twitter data has shown that it has strong correlation with actual wireless

traffic demand [9], and preliminary NLP analysis has shown blackspots of repetitive complaints [10]. Compared to conventional methods such as customer helplines, drive-by testing, and network analysis tools, Twitter mining has the advantage of providing real-time capability and spatial accuracy monitoring for blackspot detection. Guo and Zhang were able to uncover key service complaints about 3G and 4G [10]. Takeshita *et al.* also implied that network failure could be detected in its early phases through monitoring Twitter [11]. Unlike Guo and Zhang, they mainly focused on how to find suitable mobile network failure tweets among various conversation topics. They suggested that traditional search methods on keywords is insufficient because of high false positive rates. Thus, they demonstrated that applying machine learning methods could suppress the false positive results so that the network failure can be found early and fast. Reference [34] proposed establishing context-aware wireless network by collecting data from three sources: network, user, and external. In which, user data might upload user subscription information and user equipment model information to network, meanwhile receiving the decoded information from base station. Authors applied NLP to classify these text information and cluster texts by K-means or agglomerative algorithm, so that network could fast adapt changing traffic pattern. Reference [35] focused on using NLP to detect Local Area Network (LAN) issue. Authors implied that even minor error in LAN would cause operators consuming plenty of time to check all devices within the network. As a result, they suggested developing a new interface for LAN to parse users' inquiry with NLP, which was designed for establishing the human-machine interaction. These previous works have shown that, through properly pre-processing tweets (remove URL, tokenization, remove stop words and so on) and applying sentiment analysis on selected tweets, it is possible to summarize users' opinions on specific topics at a high spatial resolution, which is wireless network in these works. Whilst existing work has used NLP, the generalized lexicon/corpus databases used cannot be accurately applied to large sets of Twitter data, as they do not have a topic-specific corpus.

The conventional method for extracting consumer opinion is to determine sentiment polarity by considering the content of the tweet with NLP. There are two main categories of sentiment analysis for short text data: Corpus based, and Machine Learning based. A corpus-/lexicon-based method uses dictionary with sentiment words and match them to text content of Tweets. According to the sentiment scores recorded in the dictionary, the algorithm will classify the text into positive, negative or neutral [54]. Machine learning method, however, depends on training data to establish algorithm model and realize classification of texts. Moreover, the corpus based category will require heavy human labor and strict protocol to do annotation of sentiment words from data set. Once the dictionary is established, it will not be affected by testing data. Machine learning methods will take less human labor when labeling training set without complex annotation protocol, and machine learning model may have a better adaption of data set during the learning phase. Therefore, the preparing time of machine learning methods may take much less than the first category, but accuracy may be limited by data size and algorithm characteristics. New data set may also require new labeling which reduce the applicability of generated models [55]. We will review existing related works of sentiment analysis based on these two categories.

Corpus based sentiment analysis has already been studied in many fields (movie reviews, academical paper reviews, journalism) and several popular corpus libraries have been established. Hu and Liu suggested that sentiment lexicons had been proven to be useful also for sentiment analysis on Twitter [12]; Moilanen *et al.* introduced sentiment lexicon corpus to refine and obtain a new context part [13]; For corpus applications, Finn Årup Nielsen created AFINN (name after the author) by using Amazon Mechanical Turk to label several words lists like the original Balanced affective word list and internet slang from Urban Dictionary of obscene words [14]; [36] implied a hypothesis that Tweets' sentiment would have a positive correlation with the Tweet authors' habits. In order to test this hypothesis, he also used AFINN to apply sentiment analysis and calculate the sentiment value of Tweets relating to transportation services. The result evaluated the accuracy of hypothesis and suggested that user habits might influence Tweets sentiments dramatically. Reference [37] proposed a Twitter-driven event detection system to monitor urban emergency events that happened in specific geographic locations, which included both natural and man-made disasters. AFINN was applied to analyze sentiment upon geotagged Tweets within the period of four to six hours during emergency event happens. Another popular corpus used for Sentiment Analysis in Twitter is SentiWordNet (SWN), which was firstly presented by Baccianella *et al.* in 2006 [38], and it has been widely used in Tweets sentiment detection and sentiment score calculation. Reference [39] created an emoticon dictionary based on collected Tweets from Twitter API. This corpus contained 384 emoticons describing positive, neutral and negative sentiment and evaluated sentiment score on each emotion based on the reference score from SWN, which reported high accuracy of 0.74 when using the proposed emoticon corpus. Reference [40] believed that opinion upon specific domain in Twitter might help people to make purchasing decision. They took the case study by collecting Twitter users' opinion on Oman tourism and applied SWN classifier as one of the lexicon used to do sentiment analysis, which suggested that combined lexicon base method might reach high F1 score of 79.43. Reference [19] modified an existing sentiment corpus by adding annotated hashtag names. The corpus is named as Lexicon Based Approach (LBA), which is developed by Wiebe. 233 and 157 hashtags for candidates from different political parties are selected and combined with LBA. The result shows a 7% accuracy gain on predicting users' opinion upon political candidates. Thelwall built the lexicon dictionary SentiStrength, which was annotated by humans and improved with the use of machine learning. The core classification of this work relied

on the Linguistic and Word Count (LIWC) dictionary, which the authors expanded by adding new features for the social network context [15]. Corpus based methods have been proven to be applicable in sentiment analysis upon many fields, especially for AFINN and SWN. However, no specific corpus has been established to focus on SA upon wireless network topic, and the capability of existing corpus upon this topic is not researched either.

For machine learning based approaches, [41] discussed that reviews on Twitter had enabled users to connect with each other and share opinions on specific topic or people. They applied Python library Tweepy to extract Tweets relating to five famous people and applied Naïve Bayes (NB) to classify Tweets based on users' sentiment upon them. Reference [42] focused on predicting stock market behavior based on Twitter users' sentiment analysis. In order to raise classification accuracy, [42] proposed to use Hybrid Naïve Bayes Classifier, which requires four phases – data collection, transformation, labeling and classifying. The result showed that the accuracy may reach 90.38%. Reference [43] discussed the combination of NLP and machine learning to evaluate users' online opinion of restaurants, which included online discussion forums or social media. Within the paper, NB, Supported Vector Machine (SVM), Decision Tree, Random Forest and K-Nearest Neighbor (KNN) have been adopted. The result implied that the first four algorithms may reach accuracy up to 90% with N-gram model. Moreover, neural network application on NLP has drawn attention recently. Reference [44] have applied Recurrent Neural Network (RNN) with word2vec word embedding to classify Tweets relating to US political parties' competition in mid-term elections. Through locating Tweets in 68 most competitive districts, system could predict the winner with 60% accuracy. Meanwhile, [48] and [49] discussed applying one-hot word embedding method to establish their RNN language model. Reference [18] adopted target-dependent model to consider words around target word only. In which, top 20 nouns having strongest relation with topic were selected according to their pointwise mutual information as target words. And then graph based clustering algorithm were used to classify Tweets according to the Tweets published by same author, retweets and replying relations. Reference [16] tried to analyze tweets about electronic products to predict public opinion on specific commercial brands, such as a new phone release or laptop. Reference [17] introduced a novel approach for automatically classifying the sentiment of tweets with the help of NB and SVM. These previous papers have been focusing on applying Machine learning methods to do sentiment analysis. However, they either ignored filtering topic when collecting data, or directly applied ML methods on data without data pre-preparing with existing corpus.

C. NOVELTY AND CONTRIBUTION

Despite rapid advances in NLP in other fields, NLP analysis about Twitter which focuses on mobile wireless network experience quality is relatively new, with only a few papers [10]. Yet, it is crucial for the accurate rollout of hyper-dense 5G small cells. Furthermore, conventional corpora only indicate sentiment polarity but seldom provide detailed sentiment score, which will also limit the usage of corpus for further machine learning based classification. Finally, the word sparsity and huge word vector dimensions of Tweets have made machine learning training extremely noisy. The research of Jiang *et al.* implies that conventional sentiment analysis (SA) model which analyses whole sentences without specified entity words or conventional corpora containing irrelevant sentiment words will bring 40% classification errors [18].

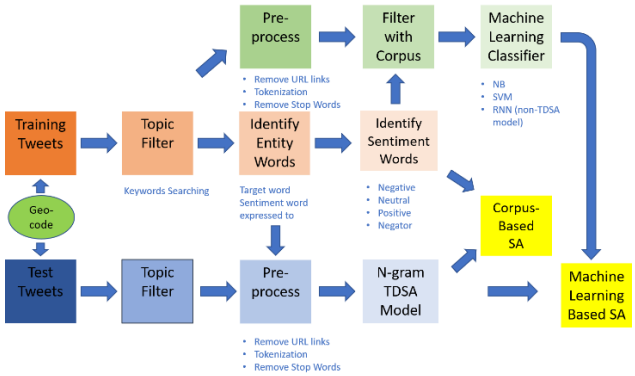


FIGURE 1. Flow chart for two categories of SA used in this paper.

In this paper, to overcome the inaccuracy of generic sentiment corpus, we first build our topic-specific sentiment corpus SignalSentiWord (SSW), which can specialize in Sentiment Analysis of Tweets relating to QoE in mobile networks. Unlike existing sentiment word corpora, we have included online slangs and mobile network related phrases (n-gram) used in Tweets through annotating the SSW; we also establish a detailed sentiment score system with a proprietary weight function instead of simply showing sentiment polarity. The performance of SSW is then compared with two other popular sentiment corpora. Next, we introduce three conventional machine learning algorithms, which are Naïve Bayes (NB), Supported Vector Machines (SVM) and Recurrent Neural Network (RNN) to classify Tweets according to their sentiment. The performance of applying machine learning classifiers only and classifiers which use the corpus as a word filter are comparatively assessed on real world data. The results show that our proposed corpus SSW improves the performance of classification by machine learning. A summary of our work flow in this paper can be found in Fig. 1.

II. CORPUS IMPLEMENTATION

In this section, we will describe the target word recognition, word annotation and sentiment score system of our proposed corpus SSW.

A. TARGET WORD RECOGNITION AND DATA COLLECTION

In order to establish a mobile network quality related corpus, we may want to collect tweets that express sentiment

towards mobile devices as training data. Rezapour *et al.* collected tweets discussing political issues from Twitter's streaming API by tracking a topic-related hashtag [19]. However, this may not be suitable for our scenario because the QoS of mobile network is not a topic as popular as, for example, politics, thus tweets containing relevant hashtags may not be sufficient in volume. Hence, we choose to track topic-related entity words (e.g. target word) instead of hashtags as the specific target in each tweet and detect sentiment words around this entity word only, which is also known as the Target-Dependent Sentiment Analysis (TDSA) model [18].

Two issues need to be addressed that would otherwise reduce the searching accuracy and efficiency for this method:

1) Users may prefer to use plain words ("signal", "phone") instead of technical terms ("cellular", "QoS"), so that a conventional entity word list for mobile network related document is not always fully applicable [20];
2) Unlike technical terms, which have a specified meaning, target words used in microblogs usually contain multiple meanings that drift with time and location, creating ambiguities during the NLP training process [21].

We have adopted Word Sense Disambiguation to solve these issues. Word Sense Disambiguation (WSD) is the task of identifying the intended meaning (sense) of words in context [22]. The over-arching goal is to find the most frequently used context word for a specific meaning to describe QoE. Signal strength is the crucial parameter to evaluate QoE: Peak-Signal-Noise-Ratio (PSNR) has been widely used by researchers to assess mobile network QoE [51], [52]; Received Signal Strength (RSS) is also the standards to determine the other important QoE parameter – coverage [53]. Since terms such as PSNR may not be highly used by Twitter users, we have selected the plain word 'signal' as the root word and search string for collecting tweets. Then, we use Pearson Correlation [23] to calculate the correlation between the most frequent used words in the tweets data with the root entity word "signal" and find out which words have a high correlation with its interpretation as "mobile network signal". The formula is as follows: n_{11} represents the number of Tweets where both word A and B appear, n_{00} is the number where neither appears, n_{10} and n_{01} are the cases where one appears without the other. The correlation coefficient is given by:

$$c = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1x}n_{0x}n_{x0}n_{x1}}} \tag{1}$$

Therefore, if the root word 'signal' and its highly correlated word appear in one tweet, we will have high confidence that this tweet is related to our scenario. Table 1 shows the accuracy when searching combined entity words with top correlation words for 1000 collected tweets on each search in New York, which demonstrates high searching accuracy.

TABLE 1. Searching accuracy for entity phrase.

Searching Keywords	Percentage of Related Tweets
Phone Signal	89.3%
Network Signal	74.5%
Signal Strength	71.6%
Cell signal	87.6%
Signal bar	86.6%
Phone Service	79.1%

B. ANNOTATION PROTOCOL AND DEFINITION

Generally speaking, annotation is "a note by way of explanation or comment added to a text or diagram", which is the first step to establish a new corpus. Specifically, we need to annotate mobile network related tweets, noting down the target words and corresponding sentiment words [19]. In order to ensure agreement between annotators, a protocol of annotating must be formulated. The protocol contains instructions, definitions and examples for the annotation process, and the detailed definition of annotated words are as follows:

Positive Sentiment Word: Positive sentiment word which describes feelings of pleasure, satisfaction, compliment or recommendation.

Negative Sentiment Word: Negative sentiment word describes feelings of disagreement, disapprove, complaint or hate. Some technical terms, which may not be typical adjectives, such as "busy signal", "call drops" or "technical issues", are also considered as negative as long as they are referring to the target word. Typical negator included in special phrases to express negative in our scenario will be considered as part of sentiment word instead of negator, such as "no signal".

Neutral Sentiment Word: Neutral sentiment tweets are those annotators cannot identify as an opinion (positive or negative). Some tweets are just describing a fact, or asking a question, while some tweets may be a mobile network related advertisement. The annotator will label these tweets as neutral and sometimes there are no typical neutral sentiment words in these tweets.

Negator Word: We noticed that certain users on Twitter prefer to use negation words ("not" is a typical negator) instead of directly using a negative sentiment word. These negators may invert the sentiment polarity of a sentence and generate an opposite expression to the original sentiment word. As a result, we also ask the annotators to note down negator words and combine the word list in our corpus, unlike traditional corpus such as AFINN. Negator word is considered as valence shifter, which may reverse the orientation of sentence. However, itself does not contribute to users' sentiment because it does not have 'content' value. In this paper, we follow the conventional 'reverse' model [45], [46] to mark Tweets with negator word as '-negator' (only consider negator within TDSA window), and then remove negator as stop words because it contains no sentiment value. After sentiment analysis on Tweets (either by corpus or machine

learning algorithm), the sentiment value of Tweets with '-negator' mark will be reversed.

C. ANNOTATION PROCESS

After defining all the annotation terms, the annotation process can be summarized as follows:

1) *Identify if the tweet is relating to the topic*. Since we are using tweets to evaluate users' opinions on phone signal quality or a related mobile topic. Annotator will identify if the tweets are relevant and discard the unrelated tweets without further processing.
2) *Identify the entity word or phrase*. After reading the tweet, annotator will identify the entity word or phrase to which the sentiment is directed.
3) *Identify what best describes the author's attitude, evaluation or judgment towards the primary entity*.
   - Positive: there is an implication in the text suggesting that the speaker's attitude or judgment of the entity is positive (speaker is appreciative, thankful, excited, optimistic, or inspired by the primary entity).
   - Negative: there is an implication in the text suggesting that the speaker's attitude or judgment of the entity is negative (speaker is critical, angry, disappointed in, pessimistic, expressing sarcasm about, or mocking the primary entity).
   - Neutral: there is no implication indicating that the speaker feels positively or negatively.
   - Negator: there is implication indicating that the speaker is using a negator word to invert her opinion on the entity, either positive or negative.
4) *Identify the sentiment word and negator word after determining authors' attitudes towards primary entity.* Once the attitude is determined, annotator should select the most suitable sentiment word that expresses author's opinion and note it down. Once the negation expression is determined, annotator should select the most suitable negator and note it down.

D. SENTIMENT SCORE SYSTEM

As discussed in the introduction, simply indicating sentiment polarity in a corpus may not be enough for our scenario, a sentiment score system is required for the subsequent machine learning based classification. Neviarouskaya *et al.* proposed a sentiment score calculation method based on SentiWordNet, which assumes that each word has different senses and therefore multiple meanings [24]. The sentiment score for each sense of the same word will also be different and each sense will be given three scores: positive, negative and objective. The sum of the three scores for each sense is equal to 1, thus the score for each sense can also be considered as the possibility of this sense's polarity. Alena *et al.* suggest calculating a unique sentiment score instead of multiple ones for each word according to Equation (2), (3) and (4).

$$UniPos = \left(\frac{\sum_{i=1}^{p} Pos(i)}{p}\right) \tag{2}$$

$$UniNeg = \left(\frac{\sum_{i=1}^{q} Neg(i)}{q}\right) \tag{3}$$

$$UniNeu = \left(\frac{\sum_{i=1}^{e} Neu(i)}{e}\right) \tag{4}$$

In equation (2), *UniPos* is the calculated unique positive sentiment score for the word, p is number of positive senses for the word, *Pos(i)* is the specified positive sentiment score for the i_{th} sense. Equation (3) and (4) are similar to (2), which calculates the unique negative sentiment score (*UniNeg*) and neutral sentiment scores (*UniNeu*) for the word separately. q and e are the numbers of negative and neutral senses for the word. After calculating the three scores and finding out the highest one, the unique score for this word will be decided: the final score $FS = 0$ if *UniNeu* is highest; $FS = UniPos$ if *UniPos* is highest; $FS = -UniNeg$ if *UniNeg* is highest. At the end, the score for all words in the corpus will be normalized from -1 to 1. Having this final/unique score instead of separate scores for various senses is quite common for corpus-based sentiment analysis methods, such as AFINN and SentiStrength. One reason is that it is hard for corpus-based methods to detect which sense the current word is referring to. Although it is possible to 'guess' the sense by considering the context of words, the accuracy is low due to word ambiguity and complexity will be increased dramatically for corpus-based methods. In order to compensate for divergent senses in tweets, we introduced a weight function in the final score calculation.

Alena *et al.*'s method provided a unique score for each sentiment word by only focusing on its most frequently used polarity. However, the formula to calculate the unique score may not be fair because it does not take the usage frequency into consideration and senses with a low frequency may contribute to the final score as much as those with a high frequency. Therefore, our first modification to the algorithm is to introduce the weight function into the calculation. The weight function and modified *Uni* function for a positive score is given in equations (5) and (6), where p is number of positive senses used in the training data, $n(i)$ is the number of appearances of the current sense in the whole training data, T is the number of appearances of the current word in the training data (the *Uni* score for negative and neutral is similar). The weight function depends on the annotation of training data; therefore, the sentiment score will also adapt to changes in collected data, which will reflect the usage frequency for mobile network related tweets. Our second modification of the algorithm is to observe all three unique scores for each sentiment word, instead of obtaining the final score *FS*.

$$WeightPos\,(i) = \left[\frac{n(i)}{T}\right] \tag{5}$$

$$UniPos' = \sum_{i=1}^{p} Pos\,(i) \cdot WeightPos(i) \tag{6}$$

TABLE 2. Corpus sentiment words.

Type	Number
Negator	10
Neutral Words	25
Negative	366
Positive	109

III. ANALYSIS WITH PROPOSED SSW CORPUS

We collected 93,248 tweets by searching 6 entity phrases on London, New York, San Francisco, Los Angeles and Birmingham (the searching range is 25 miles from the geo-code of each city center) for April – July 2018 and performed annotation of the data set. The final corpus contains 366 negative words, 109 positive words, 25 neutral words and 10 negator words, which contains up to 3-gram phrases.

The most noticeable observation from the corpus is that: negative sentiment words have significant more numbers than positive and neutral words. One reason is that users may prefer to use slang and informal words when expressing negative sentiment, which contributes to more negative words. The other reason is that users are more likely to express negative sentiment (complaints, fault report and so on) on Twitter when talking about mobile signal or service. Users may have a high expectation on signal quality and will less likely express positive sentiment. The training data also supports this reason with a ratio between negative Tweets and positive Tweets = 5.47:1.

We have created our own corpus to mark each target-specific tweet with a sentiment score. The next step is to apply an N-gram Window TDSA model [18] to classify tweets according to their sentiment. In order to analyze how N will affect the result for our scenario, we choose 1000 Tweets with entity words "phone signal" with the searching range as 25 miles from the geo-code of London city center. We then apply the SSW corpus and $N = 2$ to 6 model the test data and record the accuracy by manual annotation, and then plot the PDF of the result and compare. Fig. 2 shows the PDF for different values of N. For $N = 2$, we can see that the peak is located at a sentiment score = 0, which means most tweets are considered as neutral and the user does not express a sentiment on the "phone signal" topic. The second peak is negative. In general, we conclude that the general sentiment for "phone signal" in London is neutral. However, if we set $N = 3$, we find out that the first peak is now negative, while the neutral peak has a significant drop. The results of $N = 3$ and $N = 2$ are inconsistent and the former shows a negative sentiment instead of a neutral one. In order to test the accuracy, we continue to increase number of N from $N = 2$ to 6. The results imply that after $N = 3$, the PDF has a similar tendency, which shows a major negative impact, meanwhile the neutral peak keeps decreasing. As a result, we conclude that $N = 2$, which shows a majority neutral sentiment is not suitable for our scenario. The reason may be: 1) although signal is the primary entity word, it is normally accompanied with "phone" or "network" to form a short phrase so that the sentiment words may sometimes be filtered out by $N = 2$; 2) inclusion of negator words such as "not" "don't", this is also another reason that we decide to omit negator word from N-gram and make '-Negator' mark instead; 3) inclusion of adverbs such as "really" in "the phone signal is really bad" ("is" considered as stop word and will be omitted before analysis so that the processed sentence fed into the classifier is "phone signal really bad").

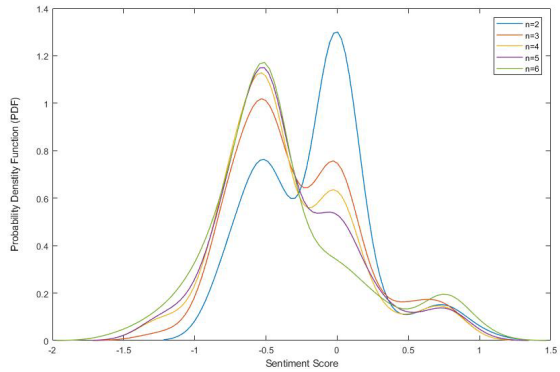


FIGURE 2. PDF for N-gram sentiment detection for $N = 2$ to $N = 6$.

TABLE 3. Accuracy for change of N.

Number of N	Accuracy
N=2	53.2%
N=3	69.4%
N=4	72.3%
N=5	74.5%
N=6	73.7%

We have used PDF to make the comparison and showed that $N = 3$ may be better than $N = 2$. However, it may not mean that higher N is better. The increased complexity is one reason. In Table 3 we show the results of applying $N = 2 - 6$ on the dataset to compare the accuracy.

According to the test data, we can conclude that $N = 5$ will bring highest accuracy and $N = 3$ will give us the best marginal benefit. The other reason causing the accuracy drop after $N = 4$ is that the algorithm may take sentiment words that relate to other topics into consideration and generate another type of noise. $N = 5$ may provide the best accuracy, but the marginal benefit has dropped heavily. The length of the N-gram we feed into the NLP system will affect the computation complexity. Longer words will require more time to process each word of the N-gram, the computation time will increase accordingly and the issue may be more severe when applying SVM and RNN, which will convert N-gram into vectors to train the model. As a result, we can choose the number of N which suits our purpose better. The final decision may be restricted given time and resource

and we choose $N = 5$ in this paper because we value the accuracy most considering the importance of user opinion in our scenario. Now that the optimal N is obtained, we will evaluate the performance of our corpus by comparing the results with two other popular sentiment analysis corpora:

- **AFINN**: this is a publicly available lexicon motivated by some classic lexicons such as ANEW and GENERAL Inquirer. It uses Amazon Mechanical Turk to label several words lists, such as the original balanced affective word list and internet slang from the Urban Dictionary and obscene words. AFINN classifies messages in a range of $[-5, 5]$, with -5 and 5 being the most negative and most positive score, respectively.
- **SentiWordNet**: this is a tool that is widely used in opinion mining and is based on an English lexical dictionary called WordNet. This dictionary groups adjectives, nouns, verbs and other grammatical classes into synonym sets called synsets. SentiWordNet associates three scores with synset from the WordNet dictionary to indicate the sentiment of text: positive, negative, and neutral.

Having collected another 1700 Tweets with entity words "phone signal" in New York as a new test dataset, we apply three sentiment analysis methods with the $N = 5$ target specific window model on the data and analyze the result with three corpora (we consider detecting negative sentiment apart from non-negative sentiment polarity because we focus on evaluating the choice of sentiment word and detection of signal blackspot). The detection of tweets expressing negative sentiment is vital for our scenario, so that we will consider predicted negative sentiment tweets as positive condition when calculating F1 score. The actual sentiment status for the test tweets are done by annotator. The results show that our proposed corpus has the best precision of 0.752, recall of 0.693 and F1 score of 0.721, confirming its advantages. SentiWordNet behaved worst with an F1 score of 0.674. Table 4 shows the results.

We have present Fig. 3 to show how three corpora give sentiment scores on the same twitter data which is labeled with sentiment polarity by annotators in our scenario, and check the variation so that we can cross-validate all three corpora and analyze the compare each corpus for classifying sentiment. After detailed investigation of the 'behavior' of the three corpora, we summarize the following three reasons why our corpus achieved the best performance compared to AFINN and SWN: 1) internet languages have been ignored by both corpora, and emoticons are not included. For our scenario of Twitter, internet languages and emoticons are frequently used to show sentiment, which causes poor results of two corpora in classifying our test data. Although conventional sentiment words are still commonly used in Twitter, many short tweets written with internet languages will be mis-classified by AFINN and SWN. 2) Mobile related terms are not included, such as "two bar" and "4G". Such words are specific terms to describe the condition of phone signals.

TABLE 4. Accuracy for change of N.

Corpus	F1 Score
SSW	0.721
AFINN	0.703
SWN	0.674

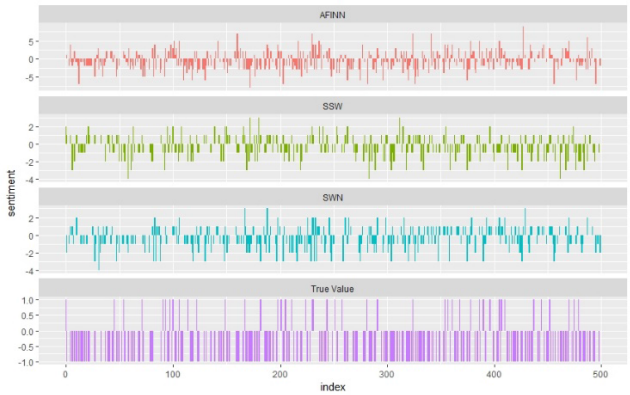


FIGURE 3. Sentiment score comparison for 3 corpora on same Twitter data.

"two bar" are commonly used to complain the poor phone signals, while "4G" are normally expressing a good phone reception. Both corpora fail to detect these sentiments. 3) the meaning of formal words has been changed in Twitter. For example, "dear" is considered as positive words in AFINN. In twitter, however, it is commonly used to show surprise or even angry, which is expressing negative sentiment. As a result, both corpora may generate the same contract due to these three reasons.

After cross comparing the results of the three corpora, we show in Fig.4 the cumulated data error for the corpora to compare error rates. Clearly, the proposed SSW shows better performance than SWN and AFINN with slower cumulated error plot and lower total error. SWN has the worst performance, especially in detecting negative sentiment, while SSW is weak on detecting neutral sentiment.

IV. ANALYSIS WITH MACHINE LEARNING

As discussed in literature reviews, machine learning has been adopted in NLP for sentiment analysis. In this section, we will first apply three commonly used machine learning algorithms in our scenario. And then we will discuss how to combine our annotated SSW with them and evaluate the performances.

A. Naïve BAYES

So far, we have shown that a corpus-based dictionary is one important approach to sentiment analysis. It is normally based on a predefined dictionary with positive and negative words, and then matches sentiment words against those in documents to measure sentiment. With annotations for new sentiment words, the corpus may even cope with specialized vocabularies [25]. However, these specialized human-annotating

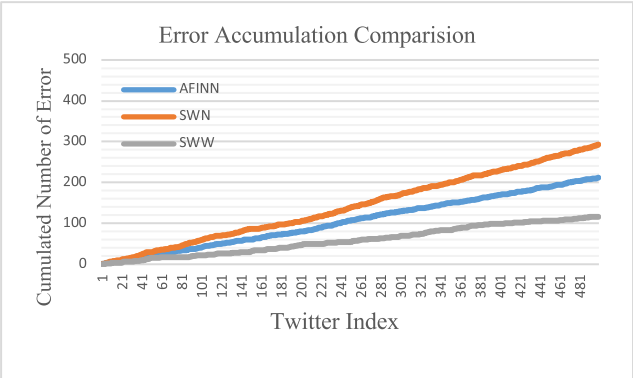


FIGURE 4. **Twitter accumulation comparison.**

corpora are likely to be labor intensive; and moreover, these corpora may not automatically adopt online slang.

The aim of this section is to provide an overview of machine learning approaches for sentiment analysis based on a Naïve Bayes classifier in order to help to solve these issues and develop a sentiment classifier for analyzing Twitter users' attitudes on mobile network QoE.

Naive Bayes is a probabilistic classifier, meaning that for a document d, out of all classes $c \in C$ the classifier returns the class c' which has the maximum posterior probability given the document [26].

$$c' = argmaxP\,(c \mid d) \tag{7}$$

This idea of Bayesian inference has been known since the work of Bayes (1763), Bayesian inference was first applied to text classification by Mosteller and Wallace (1964). Equation (8) represents Bayes' rule, which defines conditional probability $P\,(x|y)$ as follows:

$$P\,(x \mid y) = \frac{P\,(y \mid x)\,P(x)}{P(y)} \tag{8}$$

Combine Equation (7) and (8) by simplifying the results, we will be able to obtain Equation (9). Suppose that the document remains the same, $P(y)$ will be the same for all classes and can be ignored. The set $f1, f2 \ldots$ represents the features of the document.

$$c' = argmaxP\,(f_1, f_2, \ldots, f_n \mid c)\,P\,(c) \tag{9}$$

However, this equation is still hard to solve and we need further assumptions to simplify it. The first one is bag-of-words (BoW), which records only word frequencies within the document (i.e., word position is ignored). Thus, the feature set f_1, $f_2 \ldots f_i$, only represents word identity and not position.

The second assumption is the conditional independence assumption that the probabilities P $(f_i|c)$ are independent given the class c and hence Equation (9) can be finalized as follows:

$$c' = argmaxP\,(f_1 \mid c)\,P\,(f_2 \mid c) \ldots P\,(f_n \mid c)\,P(c) \tag{10}$$

We can now map the Naive Bayes classifier onto our scenario and suppose that there are three classes: Positive, Negative and Neutral. For the training data we will be able to calculate the probability of each word showing up in different classes. We can also calculate $P(c)$ for positive, negative and neutral sentiment by counting the respect tweets. However, a normal NB classifier may face issues when dealing with our scenario. During the corpus annotation, we found out that users tend to use intensive sentiment words when expressing positive or negative attitudes. When expressing neutral attitudes, however, it is hard to recognize special sentiment words (there are 366 frequently used negative words, but only 25 neutral words). Under such circumstance, if we still calculate the probability for all the words in the 'bag' and make a decision according to training data probabilities, it may lead to bias when recognizing neutral sentiment. Also, due to the limited size of the training dataset, neutral words such as 'internet' may be considered to contribute more negative sentiment, just because it shows up more times in negative tweets.

Our approach is to combine our annotated corpus with the classifier and only calculate the probability for the smaller bag of words that contains sentiment words. In specific, we will establish normal BoW first, which contains all vocabularies appearing in Tweets data. And then we will start to filter the BoW based on SSW, only entity, positive, negative and neutral words shown in SSW will be left in BoW. Entity word is used to apply TDSA model, and sentiment words within N-gram model will be left. Moreover, only sentiment words will be considered when calculating P(Word | Sentiment), or P(W | Sentiment). Because sentiment words contain high intensity for expressing attitudes, their contribution to positive or negative sentiment will be higher than neutral words (in other words, higher probability). As a result, if a tweet contains no sentiment word, P(Pos)=P(Neg)=P(Neu) and we may conclude it is a neutral tweet. By filtering Tweets with corpus words, we will also avoid the bias caused by high frequency irrelevant words (e.g. 'internet') due to limited or unbalanced training data size.

The following table contains 10 example negative sentiment words combined with their respect conditional probabilities.

The first column is the set of 10 negative example sentiment words. The second column is the probability this sentiment word appears given the current tweet is a negative tweet. The third column is the probability of this sentiment word appears given the current tweet is positive tweet.

It is obvious that $P(W|N)$ is higher than $P(W|P)$ for negative words, which means this word will more likely appear on negative Tweets. $P(W|N)$ and $P(W|P)$ can also be considered as the sentiment score for each sentiment word after normalization. Unlike the sentiment score provided by corpus-based methods, the score obtained from the NB classifier will automatically reflect the characteristic writing habits in the dataset. However, this may also become a drawback if the dataset is not big enough. From Table 5, $P(W|N)$ and $P(W|P)$

TABLE 5. The conditional probability of 10 example negative sentiment words.

SENTIMENT WORD	P(W|N)	P(W|P)
bad	0.001793372	0.00027
poor	0.00128655	0.000162
lose	0.001013645	0.000485
slow	0.000740741	0.000108
losing	0.000662768	0.000108
issue	0.000467836	0.000162
lack	0.00042885	0.000162
no bar	0.00042885	0
shit	0.000389864	0.0000162
low	0.000389864	0.000323

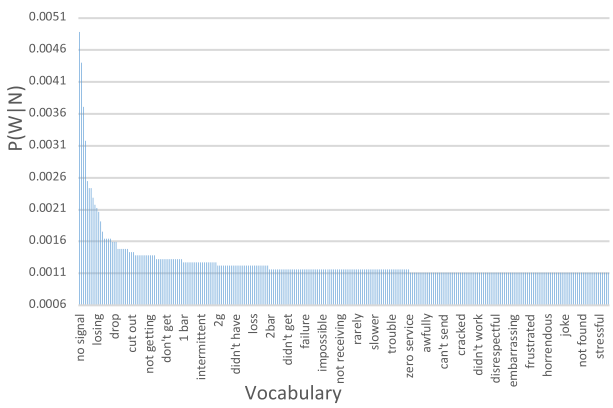


FIGURE 5. Negative sentiment words of bag with $P(W|N)$.

has very small difference compared to other typical negative words, although 'low' is an obvious negative sentiment word in our scenario. The reason may be special cases in the data or 'noise' generated from other topic words, which will become one major issue to hinder the performance of NB. As a result, we may need a larger dataset and more annotation work to train the NB classifier.

The second issue for NB methods in our scenario is tweet noise. Since NB relies on word statistics within the training data, words without sentiment but with high frequency may lead to high $P(W|P)$ or $P(W|N)$ due to data unbalance and user preference, such as 'Wi-Fi', 'phone' and so on. As a result, a NB classifier will mistakenly consider these 'neutral' words as high score sentiment words expressing negative or positive and the performance of this classifier will be impacted. As discussed above, our approach of combining our corpus SSW with a NB classifier to filter out noise words may help to mitigate this issue. Figure.5 shows top sentiment words or phrases showing negative sentiments generated by NB after filtering nose words form tweets with SSW. We can see that high frequency irrelevant words such as 'internet' or 'wi-fi' are eliminated and unique sentiment phrases such as "2G", "1 bar" has been adopted, which highly improves the performance of NB in our scenario.

B. SUPPORT VECTOR MACHINE

In last section, we have discussed a probability sentiment classifier - Naïve Bayes. In this section, we will discuss another type of machine learning classifier - Support Vector Machine (SVM). Conventional word representation models mark words as indices in a vocabulary and assess their property based on statistics, but this may fail to capture the rich relational structure of the lexicon. Compared to BoW model for NB, SVM has used a word vector model. By introducing a word vector model, we will be able to obtain more detailed information about word contexts and perform operations on words within vector space, which is not possible for BoW models. In order to map words from vocabulary to vectors, we need to represent words with real numbers in multiple dimensions. The conventional way is to decompose the word according to multiple features, so that each single word can be mapped into a geometric space with multiple dimensions [27]. Within this space, highly correlated words (good and great) tend to have similar vectors and prefer to form the same cluster, while distant words (bad and poor) will join other clusters. Therefore, by controlling the method for picking features we can obtain the required clusters and further supervised classification methods are applicable, such as SVM.

Another important application for representing words as vectors is that we can operate numerically. Therefore, the crucial factor for creating word vectors successfully is how to define features and assign respect scores. One common method is word2vec, which is 'a typical shallow, two-layer neural network' that is trained to reconstruct the linguistic contexts of words [28]. It requires a large corpus of text as the input and defines features according to the context of word, so that words sharing a common context in the corpus will be closely located. However, this model has been proven to be more effective for clustering nouns rather than adjectives because word2vec is poor in detecting comparatives and the superlatives [29].

Therefore, the core of word embedding is to find useful feature of Tweets and map them into vector system. Similar to NB, the problem of noise word will also limit the performance of the classifier in our scenario, and a new feature system specified in assessing sentiment instead of simply obtaining feature through statistics is required in SA. Nevertheless, the combination of SSW with SVM may help to solve this issue. Conventional one-hot word embedding will also adopt BoW and consider each word in vocabulary as one dimension. The value for each word is binary, 1 means this word appears in this Tweet, 0 means this word does not. Therefore, the number of dimensions will be equal to number of words in vocabulary, which may bring a dimension disaster if Tweet set is large. Also the feature extracted in this embedding method may not suit for sentiment analysis.

For our approach, we have replaced the binary value system adopted in one-hot with our established sentiment score system. Binary value system can only represent whether the word appears in the Tweets, but how it affects the Tweets in

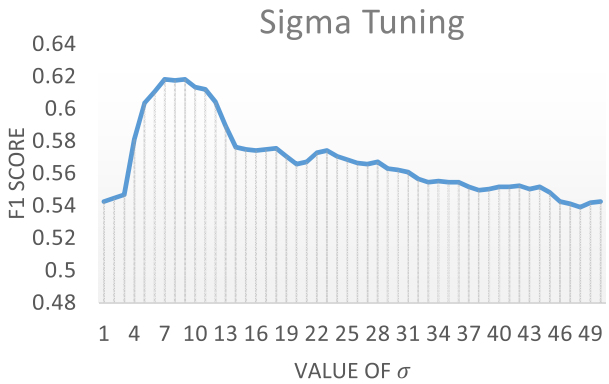


FIGURE 6. Value of σ tuning.

sentiment way and its intensity is ignored, which are actually crucial features we require in our scenario. Firstly, we narrow down the vector dimension of each Tweet from number of vocabulary to number of sentiment types, which are negative, positive and neutral. Since only sentiment words will be able to contribute sentiment score in each Tweet, we will then use SWW to filter BoW and reduce the size of vocabulary. As a result, the value in three dimensions for each Tweet will be only represented by its sentiment word within. Finally, the corresponding sentiment score calculated in SSW will be fed into three dimensions. Our established sentiment score system in section 1 has been able to help reduce the demission from several thousands to 3 compared to conventional one-hot encoding, and from 32 to 3 compared to the word2vec model. With the help of SSW, we may eliminate irrelevant dimensions and maintain relevant sentiment dimensions only.

Unlike NB, SVM trains the model after the word vector is fed into the system instead of training probability based on the bag of words. Therefore, we have modified the model SVM in two aspects to be compatible with our scenario. First, the choice of Kernel Function. The kernel function is applied to each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable. For our scenario, we have reduced the word vectors dimension to a very low level of 3 and thus the Gaussian Kernel Function has been chosen, which performs well with low vector dimension. The formula is shown below:

$$K(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2}\right) \tag{11}$$

The key factor of this kernel function is value of σ, which determines the reach of each element after mapping to higher dimension. If σ is too high, element will have small reach and those who are far from boundaries will fail to affect the boundary. Conversely, if σ is too low, element will have large reach and it may bring noise and hard to decide the boundary for large amount of elements. We must choose the parameter σ to decide the mapping area of each element for our SVM classifier. Fig. 6 shows the performance of the classifier when tuning σ, and the optimal value has been set to 7.

Furthermore, we have also modified the error function according to our scenario. Error function is used to relax the boundary between two clusters in case of unbalanced data set. Imbalance in the training set will severely affect the performance of the SVM classifier because the boundary line may be pushed to the cluster with fewer elements. Regularization parameter C and slack variable e are then introduced into error function. e represents the distance between one biased element and its correct position, which can be considered as error in vector space. C is the punishment factor for biased elements. The larger C is, the less error we want for the elements within this cluster because of the large punishment. Therefore, controlling C will also control the boundary between imbalanced clusters. One common suggestion is the ratio of C equals to the ratio of elements number for different set [50]. For example in our scenario, number of negative Tweets: positive Tweets = 5.47:1 in our training data set. As a result, the error function deciding boundary between negative and positive is modified as follows with C1:C2 = 5.47:1, where e_i represents error of positive element and e_j represents error of negative element.

$$C = C_1 \sum_{i=1}^{p} e_i + C_2 \sum_{j=p+1}^{n} e_j \tag{12}$$

C. RECURRENT NEURAL NETWORK

A Recurrent Neural Network (RNN) is a type of artificial neural network where neural nodes establish directed cycles with specific sequence [33]. The special characteristic of RNN relies on its structure, which is able to memorize temporal dynamic behavior and still retain information after several transactions. The ability of tracing back has made RNN a potential algorithm to do image processing, long sentence language processing, and so on. We have adopted Long short-term memory (LSTM) which is a modified version of the RNN, to do sentiment analysis for our scenario. Compared to a conventional RNN, LSTM introduces a hidden layer that can remember information for long periods of time [47]. The additional memory state is the line from C_{t-1} to C_t in Fig. 7. The, f_t gate is used to forget useless information and to decide which information should be stored in the cell, and $\bar{c}$ is the part used to decide how much input information should be stored and moved to the next cell. Finally, O_t is the output gate to the next cell. Using continuous RNN cells, we will be able to train and obtain useful information for sentiment analysis from longer sentences such as recent tweets since the relaxation of the word limit.

The reason we wish to apply RNNs to do sentiment analysis in our scenario is in twofold: 1) RNN is a popular and conventional method in NLP for text classification. The advantage of RNN is its 'memory' feature, which is highly effective for extracting information in the context of long sentences [30]. For example, the word 'French' in the beginning of the paragraph will contribute information for all sentence classifications in the whole paragraph. However, RNNs may require information from the whole sentence instead of just a window of N-grams like TDSA, because it depends highly on

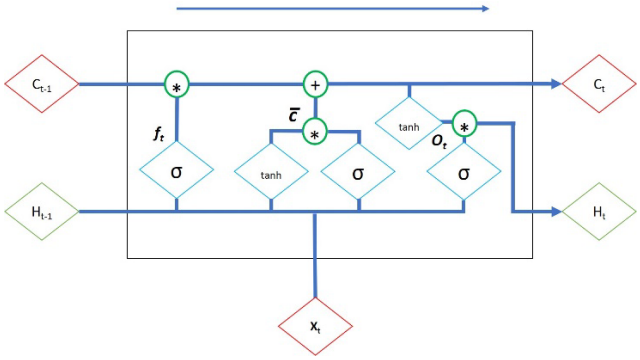


FIGURE 7. LSTM structure.

TABLE 6. performance for various methods.

Methods	F1 Score
Corpus Based	0.736
NB with Corpus	0.720
SVM with Corpus	0.618
RNN without Corpus	0.423
NB without Corpus	0.395
SVM without Corpus	0.287
RNN with Corpus	*

*The output of RNN classifier with Corpus is negative for all tweets

the context of the text. Therefore, we feed the whole sentence to the RNN classifier to train so that we can compare these two models; 2) Twitter has extended their word limitation on each tweet from 140 to 280 since 2017, which has enabled each tweet expressing users' opinions to become a short review. Therefore, we want to assess the performance of RNN on tweets given its success on conventional text classification tasks.

The configuration of a RNN is similar to SVM and the crucial step is establishing word vectors. Again, we applied one-hot and SSW filtered embedding as we did with the SVM. For the sentiment detection part, however, we feed the whole sentence instead of N-gram window used in TDSA model. For the training phrase, we have chosen the LSTM version and two embedding models are fed separately to train the classifiers. The LSTM is the 2017 Python version based on Keras (an open source neural network library running on top of TensorFlow, Microsoft Cognitive Toolkit, or Theano [31]).

D. MACHINE LEARNING METHODS COMPARISON ANALYSIS

The performance comparison of the various methods is shown in Table 6. First, we observe that using NB, SVM and RNN on the training data for sentiment analysis achieves very poor results. The F1 score for NB is only 0.395 and SVM is even worse at 0.287, while RNN achieves the best result of 0.423. This performance is too poor to be used for classifying sentiment. This result follows our prediction in previous discussion, which is due to the high frequency noise words and high dimension word vectors. After applying our corpus SSW to refine the training data both in filtering noise word and reducing word vector dimensions, both SVM and NB achieve dramatically performance improvement. For NB, the F1 score has reached 0.720. SVM has also obtained a moderate F1 score of 0.618. However, the RNN model with filtered SSW has achieved the worst result – it classifies all test tweets as negative and is a complete failure. Therefore, we conclude that combining our corpus SSW with a NB or SVM classifier will achieve a better result and NB with the SSW corpus may be the optimal result when considering both human labor and performance. RNN, on the contrary, is not suitable to combine with the SSW corpus.

TABLE 7. Word vectors for 10 negative words with positive and negative features.

Sentiment Word	POSITIVE	Negative
bad	0.25	0.614
poor	0.125	0.701
lose	0	0.759
slow	0	0.417
issue	0.228	0.125
fail	0	0.483
problem	0	0.55
weak	0	0.478
terrible	0.125	0.571
awful	0.188	0.579

In order to check why there is performance difference between NB and SVM, we have chosen 10 popular negative sentiment words. Table 7 shows ten negative word vectors established with modified sentiment scores. Table 8 shows the conditional probability for ten negative words established using Naïve Bayes. In general, they have achieved a similar result. Compared to NB, the sentiment score system is 'stricter' on detecting polarity and introducing the value of 0 (it is not likely in NB due to statistically calculation and non-zero requirement of Bag of Words). For example, 'weak' in NB has relatively the same score in both positive and negative sentiment, while it is 0.478 against 0 in Table 1. However, this score system may not reflect the real situation when creating training Twitter datasets. For example, 'issue' is considered to have higher positive score than negative when using word vectors. NB, however, considers 'weak' as definite negative sentiment words according to its probability from the training data. This may result in poor prediction and even opposite result compared to true value. As a result, SVM has a large number of false positives when classifying negative sentiment because the sentiment score may have failed to reflect the word usage preference of various Twitter users.

For RNN we have two conclusions. The first is that although RNN has proved to be successful in conventional text classification with long sentences grounded on common rules (e.g., grammar), it may not suitable for the analysis of tweets (with F1 score as 0.423). The first reason is due to the

TABLE 8. The conditional probability of 10 example negative sentiment words.

Sentiment Word	P(w|N)	P(w|P)
bad	0.001793372	0.00027
poor	0.00128655	0.000162
lose	0.001013645	0.000485
slow	0.000740741	0.000108
issue	0.000662768	0.000108
fail	0.000467836	0.000162
problem	0.00042885	0.000323
weak	0.00042885	0.000385
terrible	0.000389864	0.000162
awful	0.000389864	0.000323

special characteristics of Twitter that each tweet may be written or edited by different users, therefore the usage of words and structure of context may be totally different among training data. LSTM is specialized for detecting and summarizing long-term features of each text category. In our scenario, it is hard to find tweets with the same author because the topic of 'mobile signal' is not very popular. As a result, each dataset collected may have their own features or tweeting habits, which seriously limits the advantage of RNN. The second reason is the length of tweets. Although the limitation of words was increased to 280 in 2017, the most common length of each Tweet is only 33 characters. Moreover, the percentage of tweets reaching 140 characters has reduced from 9% to 1% [32]. The topic and expressed sentiment are actually quite focused and long tweets are either mixing topics or expressing a series of facts. Therefore, RNN considering the whole sentence may not be suitable for our scenario. As a result, RNN may not be suitable to be combined with a corpus. Our results show that filtering out words with SSW may result in a complete failure of RNN (all tweets are classified as negative sentiment). This is due to the characteristics of RNN, which is trained based on a large text context. After filtering with a corpus, only a few words are left in each tweet, which results in relatively fixed structures. The classifier will be over-fitted and classified all test tweets as negative due to the majority of negative sentiment tweets in the training dataset and therefore a large proportion of negative sentiment words.

V. CONCLUSION

A. SUMMARY AND FUTURE WORK

In this paper, we have presented the structure, protocol and annotating process of our proposed SSW corpus for detecting consumer sentiment to wireless services. After comparing with two other popular corpus libraries, our results show that SSW has an advantage in both accuracy and expertise in 'mobile signal blackspot' sentiment classification. Moreover, we have analyzed three popular machine learning methods when applied to our scenario and assessed the capability of combining the SSW corpus with ML methods. The results show that both NB and SVM have dramatically benefited from SSW filtering and NB may reach a F1 score as high as 0.720. However, RNN is not compatible with corpus filtering due to the relatively short messages on Twitter. Nevertheless, we suggest that RNN may still have potential for Twitter sentiment analysis and plan to conduct further work on this area using conditional reflection with bidirectional LSTM (BLSTM). Moreover, long text strings can be classified using conditional reflection driven RNN and this will be the focus of future work [57]. Finally, the main purpose of this paper is focusing on extracting users' opinion upon mobile network QoE from geo-tagged Tweets, so that we can use the synthesized information to detect mobile blackspots. Therefore, the spatial information is another important recourse we want to mine from Tweets. We will focus on the methods of geo information extraction in next paper and make a case study on blackspots detection.

B. TWITTER PRIVACY CONCERN

Social media data may also have different access restriction when concerning Privacy. The majority of Facebook data is protected as private and can only be accessed at an aggregate level. Twitter profiles and Tweets, on the other hand, are considered as public data by default unless user set their data as private [56]. As discussed above, we have used Twitter API to collect data, where only public Tweets are available. Moreover, we have signed and agreed developers' policies when applying Twitter API. Therefore, we will not republish the Tweets contents in our research. We will not refer or trace specific user only. We will not share data and developer's account with third party.

REFERENCES

[1] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[2] M. H. Alsharif and N. Nordin, "Evolution towards fifth generation (5G) wireless networks: Current trends and challenges in the deployment of millimetre wave, massive MIMO, and small cells," *Telecommun. Syst.*, vol. 64, no. 4, pp. 617–637, Apr. 2017.

[3] J. Huang, R. Kornfield, G. Szczypka, and S. L. Emery, "A cross-sectional examination of marketing of electronic cigarettes on Twitter," *Tobacco Control*, vol. 23, pp. 26–30, Jul. 2014.

[4] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. 12th Int. Conf. World Wide Web*, May 2003, pp. 519–528.

[5] M. A. Bekafigo and A. McBride, "Who tweets about politics?: Political participation of Twitter users during the 2011gubernatorial elections," *Social Sci. Comput. Rev.*, vol. 31, no. 5, pp. 625–643, 2013.

[6] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, Mar. 2011.

[7] R. Procter, J. Crump, S. Karstedt, A. Voss, and M. Cantijoch, "Reading the riots: What *were* the police doing on Twitter?" *Policing Soc.*, vol. 23, no. 4, pp. 413–436, 2013.

[8] A. J. Saffer, E. J. Sommerfeldt, and M. Taylor, "The effects of organizational Twitter interactivity on organization–public relationships," *Public Relations Rev.*, vol. 39, no. 3, pp. 213–215, Sep. 2013.

[9] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, "Estimating mobile traffic demand using Twitter," *IEEE Wireless Commun. Lett.*, vol. 5, no. 4, pp. 380–383, Aug. 2016.

[10] W. Guo and J. Zhang, "Uncovering wireless blackspots using Twitter data," *Electron. Lett.*, vol. 53, no. 12, pp. 814–816, Aug. 2017.

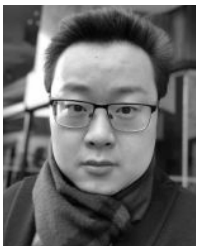
[11] K. Takeshita, M. Yokota, and K. Nishimatsu, "Early network failure detection system by analyzing Twitter data," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 279–286.

[12] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2004, pp. 168–177.
[13] K. Moilanen, S. Pulman, and Y. Zhang, "Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression," in *Proc. 19th Eur. Conf. Artif. Intell.*, Aug. 2010, pp. 36–43.
[14] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," Mar. 2011, *arXiv:1103.2903*. [Online]. Available: https://arxiv.org/abs/1103.2903
[15] M. Thelwall, "Heart and soul: Sentiment strength detection in the social Web with sentistrength," in *Cyberemotions: Collective Emotions in Cyberspace*. Berlin, Germany: Springer, 2013.
[16] M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5.
[17] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter sentiment analysis of movie reviews using machine learning techniques," *Int. J. Eng. Technol.*, vol. 7, no. 6, pp. 2038–2044, 2016.
[18] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2011, pp. 151–160.
[19] R. Rezapour, L. Wang, O. Abdar, and J. Diesner, "Identifying the overlap between election result and candidates' ranking based on hashtag-enhanced, lexicon-based sentiment analysis," in *Proc. IEEE 11th Int. Conf. Semantic Comput. (ICSC)*, Jan./Feb. 2017, pp. 93–96.
[20] E. Hovy and J. Lavid, "Towards a 'science' of corpus annotation: A new methodological challenge for corpus linguistics," *Int. J. Transl.*, vol. 22, no. 1, pp. 13–36, Jan./Jun. 2010.
[21] W. B. A. Karaa, "Named entity recognition using Web document corpus," Feb. 2011, *arXiv:1102.5728*. [Online]. Available: https://arxiv.org/abs/1102.5728
[22] J. S. Kessler, M. Eckert, L. Clark, and N. Nicolov, "The ICWSM 2010 JDPA sentiment corpus for the automotive domain," in *Proc. 4th Int. AAAI Conf. Weblogs Social Media Data Workshop Challenge (ICWSM-DWC)*, Washington, DC, USA, May 2010, pp. 1–8.
[23] M. Stevenson and Y. Wilks, "Word-sense disambiguation," in *The Oxford Handbook of Computational Linguistics*. New York, NY, USA: Oxford Univ. Press, 2003, pp. 249–265.
[24] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Sentiful: Generating a reliable lexicon for sentiment analysis," in *Proc. 3rd Int. Conf. Affect. Comput. Intell. Interact. Workshops*, Sep. 2009, pp. 1–6.
[25] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," in *Proc. ICWSM*, Mar. 2007, vol. 7, no. 21, pp. 219–222.
[26] D. Jurafsky and J. H. Martin, "Classification: Naive Bayes, logistic regression, sentiment," in *Speech and Language Processing*. London, U.K.: Pearson, 2015.
[27] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and P. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2011, pp. 142–150.
[28] Y. Goldberg and O. Levy, "word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," Feb. 2014, *arXiv:1402.3722*. [Online]. Available: https://arxiv.org/abs/1402.3722
[29] W. Ling, C. Dyer, A. W. Black, and I. Trancoso, "Two/too simple adaptations of word2vec for syntax problems," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, May/Jun. 2015, pp. 1299–1304.
[30] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," May 2016, *arXiv:1605.05101*. [Online]. Available: https://arxiv.org/abs/1605.05101
[31] F. Chollet, "Keras: The python deep learning library," *Astrophys. Source Code Library*, Jun. 2018. [Online]. Available: https://keras.io
[32] K. Gligorić, A. Anderson, and R. West, "How constraints affect content: The case of Twitter's switch from 140 to 280 characters," in *Proc. 12th Int. AAAI Conf. Web Social Media*, Jun. 2018, pp. 596–599.
[33] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 2. Jun. 2014, pp. 49–54.
[34] T. E. Bogale, X. Wang, and L. B. Le, "Machine intelligence techniques for next-generation context-aware wireless networks," Jan. 2018, *arXiv:1801.04223*. [Online]. Available: https://arxiv.org/abs/1801.04223
[35] P. Naik, S. G. Patil, and G. Naik, "Natural language interface for querying hardware and software configuration of a local area network," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 2, pp. 952–963, 2019.
[36] B. Qi and A. M. Costin, "Investigation of the influence of Twitter user habits on sentiment of their opinions towards transportation services," in *Proc. ASCE Int. Conf. Comput. Civil Eng.*, 2019, pp. 314–321.
[37] Y. Wang and J. E. Taylor, "DUET: Data-driven approach based on latent Dirichlet allocation topic modeling," *J. Comput. Civil Eng.*, vol. 33, no. 3, 2019, Art. no. 04019023.
[38] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. 7th Conf. Int. Lang. Resour. Eval. (LREC)*, Vol. 10, May 2010, pp. 2200–2204.
[39] N. M. Elfajr and R. Sarno, "Sentiment analysis using weighted emoticons and SentiWordNet for Indonesian language," in *Proc. Int. Seminar Appl. Technol. Inf. Commun.*, Sep. 2018, pp. 234–238.
[40] V. Ramanathan and T. Meyyappan, "Twitter text mining for sentiment analysis on people's Feedback about Oman tourism," in *Proc. 4th MEC Int. Conf. Big Data Smart City (ICBDSC)*, Jan. 2019, pp. 1–5.
[41] S. Kunal, A. Saha, A. Varma, and V. Tiwari, "Textual dissection of live Twitter reviews using naive Bayes," *Procedia Comput. Sci.*, vol. 132, pp. 307–313, Jan. 2018.
[42] G. A. Alkubaisi, S. S. Kamaruddin, and H. Husni, "Stock market classification model using sentiment analysis on Twitter based on hybrid naive Bayes cLASSIFIERS," *Comput. Inf. Sci.*, vol. 11, no. 1, pp. 52–64, 2018.
[43] A. Krishna, A. Aich, V. Akhilesh, and H. C. Hegde, "Analysis of customer opinion using machine learning and NLP techniques," *Int. J. Adv. Stud. Sci. Res.*, vol. 3, no. 9, pp. 128–132, 2018.
[44] A. Lopardo and M. Brambilla, "Analyzing and predicting the US midterm elections on Twitter with recurrent neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 5389–5391.
[45] F. Z. Xing, F. Pallucchini, and E. Cambria, "Cognitive-inspired domain adaptation of sentiment lexicons," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 554–564, May 2019.
[46] M. Fernández-Gavilanes, J. Juncal-Martínez, S. García-Méndez, E. Costa-Montenegro, and F. J. González-Castaño, "Differentiating users by language and location estimation in sentiment analisys of informal text during major public events," *Expert Syst. Appl.*, vol. 117, pp. 15–28, Mar. 2019.
[47] S. W. Pienaar and R. Malekian, "Human activity recognition using LSTM-RNN deep neural network architecture," May 2019, *arXiv:1905.00599*. [Online]. Available: https://arxiv.org/abs/1905.00599
[48] A. Jaech and M. Ostendorf, "Low-rank RNN adaptation for context-aware language modeling," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 497–510, Jul. 2018.
[49] S. Li and J. Xu, "A recurrent neural network language model based on word embedding," in *Proc. Asia–Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data*. Cham, Switzerland: Springer, Jul. 2018, pp. 368–377.
[50] V. Palade, *Class Imbalance Learning Methods for Support Vector Machines*. Hoboken, NJ, USA: Wiley, 2013.
[51] X. Chen, Y. Zhao, and Y. Li, "QoE-aware wireless video communications for emotion-aware intelligent systems: A multi-layered collaboration approach," *Inf. Fusion*, vol. 47, pp. 1–9, May 2019.
[52] D. Minovski, C. Åhlund, K. Mitra, and P. Johansson, "Analysis and estimation of video QoE in wireless cellular networks using machine learning," in *Proc. 11th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Jun. 2019, pp. 1–6.
[53] S. I. Popoola, A. A. Atayero, and N. Faruk, "Received signal strength and local terrain profile data for radio network planning and optimization at GSM frequency bands," *Data Brief*, vol. 16, pp. 972–981, Feb. 2018.
[54] M. Machado, E. Ruiz, and K. J. Abraham, "A new statistical approach for comparing algorithms for lexicon based sentiment analysis," Jun. 2019, *arXiv:1906.08717*. [Online]. Available: https://arxiv.org/abs/1906.08717
[55] D. Alessia, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, tools and applications for sentiment analysis implementation," *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 1–9, 2015.
[56] W. Ahmed, P. A. Bath, and G. Demartini, "Using Twitter as a data source: An overview of ethical, legal, and methodological challenges," in *Advances in Research Ethics and Integrity*. Bingley, U.K.: Emerald, Dec. 2017, pp. 79–107.
[57] Y. Jin, C. Luo, W. Guo, J. Xie, D. Wu, and R. Wang, "Text classification based on conditional reflection," *IEEE Access*, vol. 7, pp. 76712–76719, 2019.

WEIJIE QI received the Ph.D. degree from the University of Sheffield. He is currently a Research Data Scientist with the University of Warwick and seconded to Ranplan Wireless, Cambridge, U.K., as part of an Innovate UK knowledge transfer partnership. He is currently researching natural language processing and optimal wireless deployment in urban and indoor areas.

ROB PROCTER is currently a Chair Professor in social informatics with the University of Warwick and a Turing Fellow with the Alan Turing Institute, U.K. His research is in social data science and natural language processing for human behavior modeling, with research reported in Guardian newspaper. He has been an Investigator with several grants, including H2020, FP7, EPSRC, and ESRC, in the areas of social media analytics and computational social science.

JIE ZHANG is currently a Chair Professor of wireless systems with the University of Sheffield and a CTO and a Founder of Ranplan Wireless. Since 2006, he has been receiving over 20 Grants from EPSRC, Innovate UK, and the EC FP6/FP7/H2020 as PI. His research interests include the intersection of RAN, network simulation, and optimization.



WEISI GUO (S'07–M'11–SM'17) received the M.Eng., M.A., and Ph.D. degrees from the University of Cambridge. He is currently an Associate Professor with the University of Warwick and a Turing Fellow with Alan Turing Institute, U.K. He is the Head of the Data-Embedded-Networks Laboratory and has been PI on over £2m of funding and an Investigator on over £6m. He specializes in network science, wireless networks, machine learning, data science, and molecular communications. He is an Editor of two IEEE journals, has published over 110 peer-reviewed papers, and was a recipient of the IET Innovation Award.