Comput Math Organ Theory
https://doi.org/10.1007/s10588-018-9266-8

S.I. : CMKBO

The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis

Saqib Alam[1] · **Nianmin Yao**[1]

Abstract Big data and its related technologies have become active areas of research recently. There is a huge amount of data generated every minute and second that includes unstructured data which is the topic of interest for researchers now a days. A lot of research work is currently going on in the areas of text analytics and text preprocessing. In this paper, we have studied the impact of different preprocessing steps on the accuracy of three machine learning algorithms for sentiment analysis. We applied different text preprocessing techniques and studied their impact on accuracy for sentiment classification using three well-known machine learning classifiers including Naïve Bayes (NB), maximum entropy (MaxE), and support vector machines (SVM). We calculated accuracy of the three machine learning algorithms before and after applying the preprocessing steps. Results proved that the accuracy of NB algorithm was significantly improved after applying the preprocessing steps. Slight improvement in accuracy of SVM algorithm was seen after applying the preprocessing steps. Interestingly, in case of MaxE algorithm, no improvement in accuracy was seen. Our work is a comparative study, and our results proved that in case of NB algorithm, actuary was again significantly high than any other machine learning algorithm after applying the preprocessing steps; followed by MaxE and SVM algorithms. This research work proves that text preprocessing impacts the accuracy of machine learning algorithms. It further concludes that in case of NB algorithm, accuracy has significantly improved after applying text preprocessing steps.

✉ Saqib Alam
alamsaqib@mail.dlut.edu.cn

Nianmin Yao
lucos@dlut.edu.cn

[1] Department of Electronic Information and Electrical Engineering, Dalian University of Technology, Black Building, Linggong Road No. 2, Ganjingzi District, Dalian 116024, People's Republic of China

Published online: 16 March 2018

Keywords Preprocessing · Machine learning · Sentiment analysis · Word2Vec

1 Introduction

Twitter is a microblogging and real-time communication service used by millions of people and organizations to share and discover information. Users can post their messages called 'tweets' which are up to 140 characters. These messages or tweets are visible publically by default to all those who are following the tweeter. Users usually post their tweets to express their opinions or feelings about products, events or public figures which can be positive, negative or neutral. For sentiment analysis applications, twitter is the essential source of information and for analysis of this data numerous machine learning techniques were used recently (Asghar et al. 2017). In sentiment analysis, the practice of emoticons acquire a remarkable popularity particularly in the midst of youngsters to express their approach of sadness, happiness or anger towards an event, product or personality. Detecting and analyzing the applications of emoticons earned the concentration of researchers in different areas of computer science, medical and behavioral study (Asghar et al. 2017). Twitter broadcasts users' short messages regarding different aspects of life in which some of the tweets are useful for discovering the information and opinion mining. Organizations use these feedbacks for prediction and gathering useful information (Pak and Paroubek 2010).

Go et al. (2009) used emoticons in test dataset of tweets. They compared the accuracy of NB, MaxE and SVM algorithms and proved that emoticons can increase the accuracy. We took their work as a baseline, applied the preprocessing steps on the same dataset and proved that the accuracy of the same algorithms was significantly improved. Our preprocessing steps included removal of emoticons, elimination of stopwords, stemming, and the word2vec (Mikolov et al. 2013) technique that we applied on the same dataset and proved that the accuracy of NB, MaxE and SVM was improved.

The Twitter's application programming interface (API) in KNIME was used to extract large scale of tweets, which usually contains noisy data. We applied different preprocessing steps to remove the noisy data in order to improve the accuracy of the three machine learning algorithms. Go et al. (2009) used emoticons as positive and negative sentiments, while we are considering emoticons as noisy labels. We argue that some people use emoticons as sarcasm for example *"I missed his birthday :D"*. In this scenario, we cannot use emoticon as a positive or negative sentiment (Riloff et al. 2013). Hence we removed emoticons as a preprocessing step in order to improve the accuracy. Similarly we removed stopwords and applied stemming to furhter reduce the noise. For example, in the tweets, *"@stellargirl I loooooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right"*. Removing extra *o's, v's* and *e's* from *loooooooovvvvvveee* makes a proper word *love* which can be easily used in the sentiment. Finally we used the Word2vec algorithm (Mikolov et al. 2013) in order to make our results even more efficient.

1.1 Sentiments

Sentiment analysis is a computational process which identifies and categorizes an opinion in a piece of text that expresses the positive, negative, or neutral attitude of a writer towards a particular product, event or personality. For example, consider Table 1 that includes tweets which are classified as positive, negative, or neutral. Sometimes sentiments are doubtful; that either a tweet is positive or negative? Such sentiments are called neutral sentiments. To avoid ambiguity in our research work, we removed the neutral sentiments. We used the same method to eradicate neutral sentiments, which was practiced by (Go et al. 2009) in their research work, that is, a tweet should be considered as neutral if it appears as a headline on the front page of a newspaper or a statement in Wikipedia. For instance the following tweet is considered to be a neutral as it was appeared as a news headlines about former President of United States Barak Obama: *Tonight, President Obama reflects on eight years of progress. Watch the #FarewellAddress at 9 pm ET:_ofa.bo/2iYYkWQ #ObamaFarewell.* This tweet is about the former President Barak Obama which gives information about his speech; however, there is no positive or negative statement in this tweet. Hence we are not including neutral sentiments as (Go et al. 2009) did in their work in order to see the real impact of the preprocessing steps in the accuracy of MLAs.

1.2 The Tweets

Like other microblogging services, twitter can be used for uploading videos, images, links and tweets. The characteristics of tweets make it different from other social media services and attracting a number of researcher towards it. The limit for tweets are 140 characters, which automatically counts all shared links as 23 characters. The average length of tweets are 14 words or 78 characters (Go et al. 2009). The availability of the Twitter data for research purposes makes it easy to collect millions of tweets through Twitter API. This characteristics of Twitter enhances the capability of researchers to gather millions of data to train and test their models. Users from different countries and languages use different mediums for tweeting and the frequency of spelling mistakes is high as compared to other domains. Twitter users tweet about different topics, events, personalities which makes it different and attractive than other microblogging services to the researchers. Previous researches were done mostly on one specific topic such as movies and medicines reviews; however, Twitter has changed this trend.

Table 1 Sentiment analysis of Tweets

Query	Tweet	Sentiment
Jquery	Jquery is my new best friend	Positive
San Francisco	San Francisco today. Any suggestions?	Neutral
AiG	ShaunWoohaten on AiG	Negative

2 Methodology

Go et al. (2009) used different MLAs and feature extraction techniques. These algorithms are NB, MaxE and SVM. The feature extractors they used were Unigrams, Bigrams, Unigrams and Bigrams, and Unigram with part of speech (POS) tags. In this research, we are using the same MLAs however we are applying the following preprocessing steps to see their impact in the improvement of accuracy.

2.1 Removal of emoticons

Users usually use emoticons to express their feelings; such as smiley, sad, angry, and happy. However, in some cases, users use emoticons as sarcasm which is a complicated linguistic approach used commonly on social media (Khan et al. 2017). For example consider the tweet "*Thoroughly enjoyed shoveling the driveway today! :)*" It is hard to identify a user's approach towards a product, personality or event to be positive or negative using emoticons in the post. In this research, we are removing emoticons from our training dataset. In our future work, we will focus on emoticons that represent sarcastic behaviors.

2.2 Removal of stopwords

In English, stopwords are: the, is, at, which, on and so on. They have lexical content and the presence of these words can fail the required results (Baradad and Mugabushaka 2015). We have filtered out stopwords from our dataset as they are conventionally high in frequency and are not giving any useful information. In fact, they may puzzle a machine learning algorithm.

2.3 Stemming

Stemming is a technique used in information retrieval to combat the lexicon mismatch problem, in which a query's words do not match with the words of a document. Stemmers equate certain variant forms of the same word like (so, sooo). In English and many other western European languages, stemming is primarily a process of suffix abstraction (Lovins 1968; Baradad and Mugabushaka 2015). In microblogging, spelling mistakes and mismatches are common, such as *@stellargirl I loooooooovvvvvveee my Kindle2.* In training dataset there is no such word *"love"* with so many *"os, vs and es"*. We used the stemming technique as a preprocessing step to improve the accuracy of MLAs.

2.4 Word vectorization

Recently the application of word2vec charmed a great number of researchers specially machine learning community (Rong 2014). Mikolovetl et al. (2013)

proposed continuous bag-of-words mode (CBOW) and continuous skip-gram model.

2.4.1 CBOW model

A language model that can only base its presages on past words, as it is assessed predicated on its competency to prognosticate each next word in the corpus, a model that only aims to engender precise word embedding is not subject to such restriction. Mikolov et al. (2013) consequently used both the *n* words afore and after the target word *wt*to prognosticate it as shown in Fig. 1. This is kenned as a continuous bag of words (CBOW), owing to the fact that it utilizes perpetual representations whose order is of no importance.

2.4.2 Skip-gram model

The skip-gram turns the language model objective on its head: rather than utilizing the circumventing words to prognosticate the centre word as with CBOW, skip-gram utilizes the centre word to soothsay the circumventing words as can be visually perceived in Fig. 2.

2.4.3 Bigram

To ameliorate the percentage of precision obtained a bigram feature was engendered. Bigrams are two consecutive words extracted from a sentence. If a

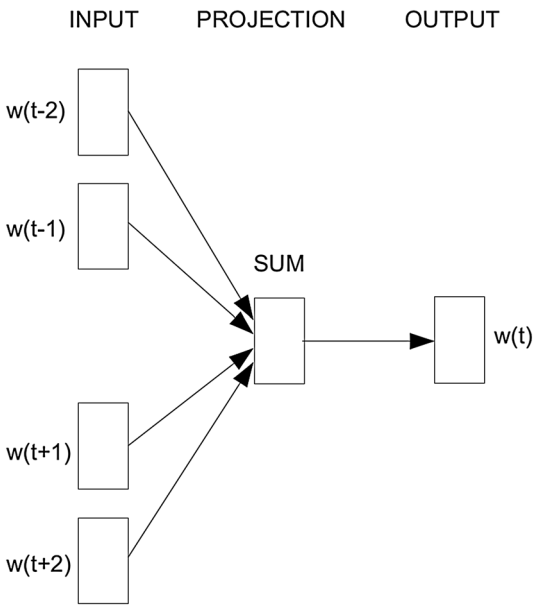


Fig. 1 New model architectures. The CBOW architecture predicts the current word based on the context

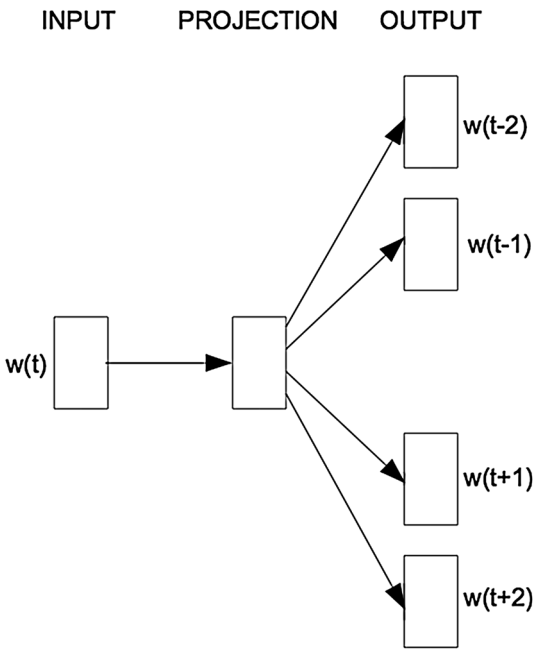


Fig. 2 The skip-gram predicts surrounding words given the current word

sentence contains n distinct terms, there would be n-2 bigrams composed from the sentence (Das and Balabantaray 2014).

3 Machine learning techniques

Machine learning gained more popularity in recent years with the rapid growth of Big Data. In our research work we are using different classifiers that is NB, MaxE and SVM.

3.1 Naïve Bayes

Due to the proliferated availability of texts in digital form and the incrementing need to access them in flexible ways, text relegation becomes an elementary and crucial task. In the past several years, many methods predicated on machine learning and statistical theory have been applied to text relegation. As the Naïve Bayesian classifier is very simple and efficient and highly sensitive to feature selection, so the research of feature selection, specially for it is paramount (Chen et al. 2009).

$$C_{NB} = \arg Max_{c_j}\{P(c_j|t_i)\} \tag{1}$$

Here $P(c_j|t_i)$ is a conditional probability and can be derived from Naïve Bayes assumption as

$$P(c_j|t_i) = \frac{\left(P(c_j)\sum_{i=1}^{m} P(f_k|c_j)^{n_i(t_i)}\right)}{P(t_i)} \tag{2}$$

In this equation, f_k represents a feature and $n_i(t_i)$ represents the count of feature *fi* found in tweet t_i. There are a total of m features. Parameters $P(c_j)$ and $P(f|c_j)$ are obtained through maximum likelihood estimates, and add-1 smoothing is utilized for unseen features.

3.2 Maximum entropy

Maximum entropy is a probability distribution estimation technique widely used for a variety of natural language tasks, such as language modeling, part-of-speech tagging, and text segmentation. The conception behind MaxE models is that one should prefer the most uniform models that satisfy a given constraint (Nigam et al. 1999). MaxE models are feature-predicated models. In a two class scenario, it is equipollent to utilizing logistic regression to find a distribution over the classes. MaxE makes no independence postulations for its features, unlike Verdant Bayes. This designates we can integrate features like bigrams and phrases to MaxE without worrying about features overlapping. The model is represented by the following:

$$P_{MaxE}(c|d) = \frac{1}{Z_{(d)}}\exp(\Sigma_i \lambda_i f_i(d,c)) \tag{3}$$

where $f_i(d,c)$ is a feature function, λ_i is the weight parameter of the feature function and $Z_{(d)}$ is a normalization factor given by

$$Z_{(d)} = \Sigma_c \exp[\Sigma_i \lambda_i f_i(d,c)] \tag{4}$$

$$P_{MaxE}(c|d,\lambda) = \frac{\exp[\Sigma_i \lambda_i f_i(d,c)]}{\Sigma_{\acute{c}} \exp[\Sigma_i \lambda_i f_i(d,\acute{c})]} \tag{5}$$

In this formula, c is the class, d is the tweet. The weight vectors decide the consequentiality of a feature in classification. A higher weight betokens that the feature is a vigorous pointer for the class. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability (Go et al. 2009).

3.3 Support vector machines

Support vector machines have strong theoretical foundations and excellent empirical successes. They have been applied to tasks such as handwritten digit recognition, object recognition, and text classification (Tong et al. 2001).

Figure 3 can visually perceive this illustrated for the example of points plotted in 2D-space. The set of points are labeled with two categories (illustrated here with ebony and white points) and SVM culls the hyperplane that maximizes the margin between the two classes. This hyper plane is given by

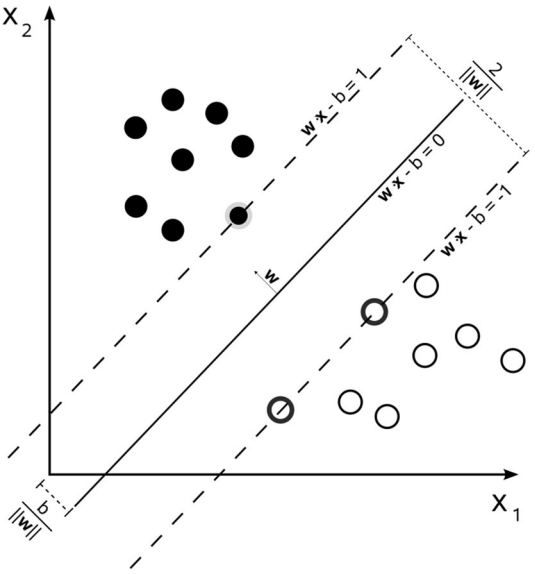


Fig. 3 Support vector machines

$$(\vec{x}.\vec{y}) + b = \Sigma_i y_i a_i (\vec{x}.\vec{y}) + b = 0, \tag{6}$$

where $\vec{x} = (x_{i1}, x_{i2}, \ldots, x_{in})$ is a n-dimensional input vector, y_i is its output value, $\vec{w} = (w_1, w_2, .., w_n)$ is the weight vector (the normal vector) defining the hyper plane and the a_i terms are the Lagrangian multipliers. Once the hyper plane is constructed (the vector $\vec{w}$ is defined) with a training set, the class of any other input vector x_i can be determined:

If $\vec{w} \cdot \vec{x} + b \geq 0$ then it belongs to the positive class (the class we are interested in), otherwise it belongs to the negative class (all of the other classes).

4 Assessment

This section discusses the dataset, experiment design and evaluation process of our research.

4.1 Dataset

Twittratr is a sentiment indicator web platform for tweets, they used a list of negative and positive sentiments. In our research work we used the list of keywords of Twittrat. This list contained 174 positive and 185 negative words (Mikolov et al. 2013). For each tweet, we count the number of negative keywords and positive keywords that appear. This classifier returns the polarity with the higher count. If there is a tie, then positive polarity (the majority class) is returned.

4.2 Experiment design

We collected the twitter data using KNIME tool (Minanovic et al. 2014). Using the Twitter's API. We retrieved specifically English language tweets using particular keywords. In KNIME we only enabled English language parameter as our classification methods were for English language only and likewise we trained our model on English only.

In social media emoticons are playing a key role, commonly users are using emoticons for expressing their expressions. For example, :-) and :D both express positive emotion. The list of a few emotions can be seen in Table 2.

In our research we considered emotions as noisy labels, we filtered them out from our collected data. As we mentioned that usually users expressing their expressions by using emoticons, using emojis as sarcasm is common exercise. Thus avoiding negative impact of emoticons on our result's accuracy, we removed them from our training dataset.

To train our data for training purpose, we used the following filters:

Emoticons listed in Table 3 are stripped off. This is important for training purposes. If the emoticons are not stripped off, then the MaxE and SVM classifiers tend to put a large amount of weight on the emoticons, which hurts accuracy.

Any tweet containing both positive and negative emotions are abstracted. This may transpire if a tweet contains two subjects. Here is an example of a tweet with this property: *Target orientation :(But it is my day of inchoation today :).* These tweets are abstracted because we do not optate positive features marked as a component of a negative tweet, or negative features marked as a component of a positive tweet.

Retweets are abstracted. Retweeting is the process of replicating another user's tweet and posting to another account. This conventionally transpires if a utilizer relishes another user's tweet. Retweets are commonly abbreviated with "RT." For example, consider the following tweet: *Awe-inspiring! RT @rupertgrintnet Harry Potter Marks Place in Film History* http://bit.ly/Eusxi:). In this case, the utilizer is rebroadcasting tweet and integrating the comment Awe-inspiring!. Any tweet with RT is abstracted from the training data to evade giving a particular tweet extra weight in the training data.

Reiterated tweets are abstracted. Infrequently, the Twitter API returns duplicate tweets. The scraper compares a tweet to the last 100 tweets. If it matches any, then it discards the tweet. Homogeneous to retweets, duplicates are abstracted to evade putting extra weight on any particular tweet.

Table 2 List of emoticons

Emoticons mapped to :)	Emoticons mapped to :(
:)	:(
:-)	:-(
:)	: (
:D	
=)	

Table 3 Categories for test data

Category	Total	Percent
Company	119	33.15
Event	8	2.23
Location	18	5.01
Mics	67	18.66
Movie	19	5.29
Person	65	18.11
Product	63	17.55
Grand total	359	

To ameliorate the quality of textual data, it is compulsory to filter out noise textual data from our tweets, the high degree of spelling errors, irregularities and idiosyncrasies in the utilization of punctuation, white space and capitalization effect the results (Clark et al. 2003).

In our research work, we collected 177 negative tweets and 182 positive manually using KNIME Twitter API. For data collection, we used the following process:

We probed the Twitter API with categorical queries. These queries are arbitrarily called from different domains. For example, these queries consist of consumer products *(Weka, Nike, g2),* companies *(Nike, boozallen),* and people *(Obama, Danny Gokey).* The query terms we used are listed in Table 3. The different categories of these queries are listed in Table 6.

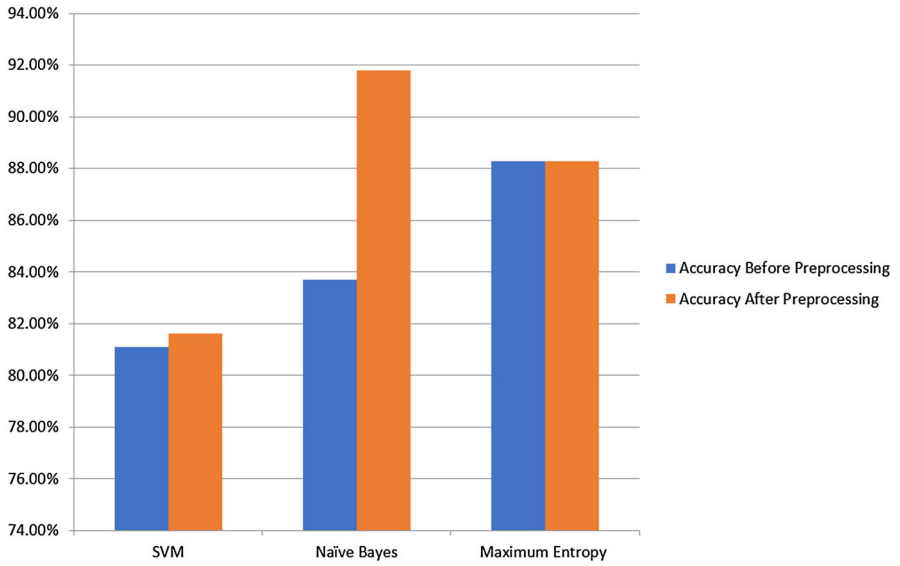
We visually examine the result set for a query. If we optically discern a result that contains a sentiment, we mark it as positive or negative. Thus, this test set is culled independently of the presence of emoticons.

4.3 Results and discussion

We removed emoticons and applied bigrams on the dataset to see the real impact of text preprocessing steps. We used removal of stop words, stemming, and work vectorization as our text preprocessing techniques. The experiment was run on a dataset of 359 documents. Consider Table 4 that shows accuracy of different machines learning algorithms before and after applying the preprocessing steps. In the first part of our experiment, we did not apply the preprocessing steps and calculated accuracy of SVM, NB and MaxE algorithms which was 81.09, 83.69, and

Table 4 Classifiers' accuracy before and after preprocessing

Algorithms	Accuracy before preprocessing (%)	Accuracy after preprocessing (%)	Net improvement (%)
SVM	81.09	81.63	0.54
NB	83.69	91.81	8.12
MaxE	88.27	88.27	0



Graph 1 Comparison of accuracy before and after the preprocessing steps

Table 5 Classifiers' accuracy with Go et al. (2009) and our research work for biagram

Algorithms	Accuracy with Go et al. (2009) (%)	Accuracy with our research work (%)	Net improvement (%)
SVM	78.8	81.63	2.29
Naïve Bayes	81.6	91.81	10.21
Maximum entropy	79.1	88.27	9.17

88.27% respectively. In the second part of our experiment, we applied the preprocessing steps and then applied machine learning algorithms. As shown in Table 4, accuracy of SVM was improved to 81.63 from 81.09%. Similarly the accuracy of NB was improved to 91.81 from 83.69%. Interestingly there was no change in the accuracy of MaxE algorithm.

The following Graph 1 shows the comparison of the accuracy of the three MLAs before and after applying the preprocessing steps. It is obvious from the graph the accuracy of NB algorithm was significantly improved after applying the preprocessing steps. However, in case of MaxE algorithm, no improvement in accuracy was seen.

We also compared our results with the findings of Go et al. (2009). The following Table 5 shows the variance between Go et al. (2009) and our research work. We can see that preprocessing has increased the accuracy of MLAs. Accuracy of the three algorithms SVM, NB, and MaxE was improved to 2.29, 10.21, and 9.17% respectively (Table 6).

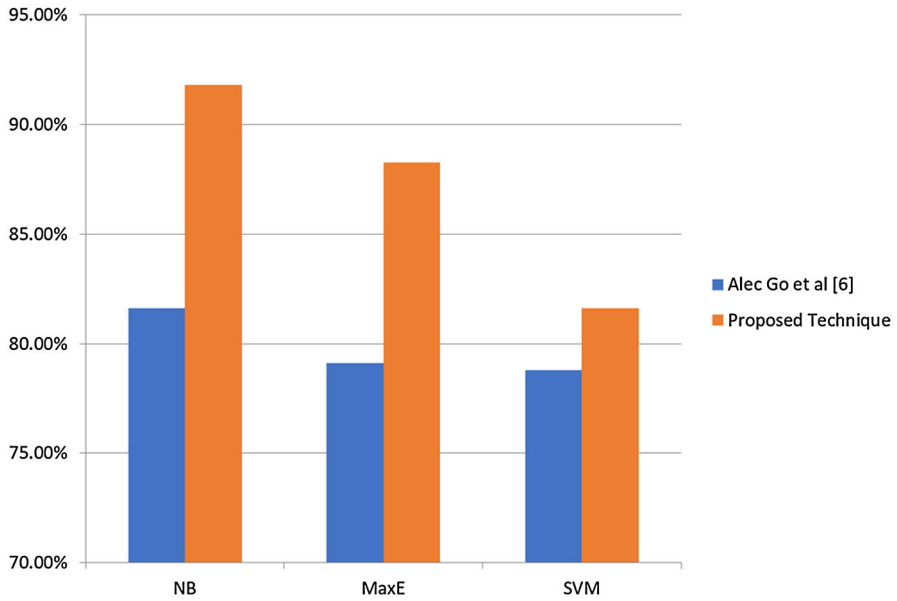
Table 6 List of queries used to create test set

Query	Negative	Positive	Total	Category
40d		2	2	Product
50d		5	5	Product
Aig	7		7	Company
At&t	13		13	Company
Bailout	1		1	Misc
Bing	1		1	Product
Bobby Flay		6	6	Person
Boozallen	1	2	3	Company
Car warranty call	2		2	Misc
Cheney	5		5	Person
Comcast	4		4	Company
Danny Gokey		4	4	Person
Dentist	9	3	12	Misc
East palo alto	1	2	3	Location
Espn	1		1	Product
Exam	5	2	7	Misc
Federer		1	1	Person
Fredwilson		2	2	Person
g2		7	7	Product
Gm	16		16	Company
Goodbysilverstein		6	6	Company
Google	1	4	5	Company
Googleio		4	4	Event
India election		1	1	Event
Indian election		1	1	Event
Insects	5	1	6	Misc
Iphone app	1	1	2	Product
Iran	4		4	Location
Itchy	5		5	Misc
Jquery	1	3	4	Product
Jquery book		2	2	Product
Kindle2	1	16	17	Product
Lakers		4	4	Product
Lambda calculus	2	1	3	Misc
Latex	5	3	8	Misc
Lebron	4	14	18	Person
Lyx		2	2	Misc
Malcolm Gladwell	3	7	10	Person
Mashable		2	2	Product
Mcdonalds	1	5	6	Company
Naive Bayes	1		1	Misc
Night at the museum	3	12	15	Movie

Table 6 continued

Query	Negative	Positive	Total	Category
Nike	4	11	15	Company
North Korea	6		6	Location
Notre dame school		2	2	Misc
Obama	1	9	10	Person
Pelosi	4		4	Person
Republican	1		1	Misc
Safeway	5	2	7	Company
San Francisco	3	1	4	Location
Scrapbooking		1	1	Misc.
Shoreline amphitheatre		1	1	Location
Sleep	3	1	4	Misc
Stanford		7	7	Misc
Star trek		4	4	Movie
Summize	2		2	Product
Surgery	1		1	Misc
Time warner	33		33	Company
Twitter		1	1	Company
Twitter api	6	2	8	Product
Viral marketing	1	2	3	Misc
Visa		1	1	Company
Visa card	1		1	Product
Warren buffet		5	5	Person
Waves & box		1	1	Product
Weka	1		1	Product
Wieden		1	1	Company
Wolfram alpha	1	2	3	Product
World cup		1	1	Event
World cup 2010		1	1	Event
Yahoo	1		1	Company
Yankees		1	1	Misc
Total	177	182	359	–

Graph 2 shows the comparison of Accuracy with different MLAs for Go et al. (2009) and our proposed technique. The graph depicts significant improvement in accuracy in case of NB algorithm; followed by MaxE and SVM algorithms. It is obvious from both Graphs 1 and 2 that accuracy was improved after applying the preprocessing steps. However, accuracy was significantly improved in case of the NB algorithm.



Graph 2 Comparison of accuracy with Go et al. (2009) and our research work

5 Related work

Asghar et al. (2017) in their work they discussed about the twitter challenges, such as abbreviations, slang language and insufficient words, emoticons and domain specific words. For solving these issues, they proposed a technique that can pointing out the emoticons and classify them by applying emoticons dictionary. It further classifying the slang in the tweets.

Yadav and Pandya (2017) in their research they discussed the importance of emoticons in the sentiment analysis. They used various ML techniques along with text pre-processing for movie or product review. They proposed a hybrid model which is the combination of lexicon and different ML techniques.

Mubarok et al. (2006) discussed in their research work that NB classifier achieve best results on the basis of numerous tests which were conducted on aspect based, aspect classification and sentiment classification, and he F1-Measures were 78.12, 88.13 and 75% simultaneously. They further stated that in their work that POS tagging and Chi Square can improve and speedup the calculation time of NB classifier.

Xie et al. (2017) In this article, they used Maximum entropy probabilistic semantic analysis (PLSA) model. In their proposed model they exercised the probabilistic semantic analysis (PLSA) for grabbing the seed emotions words from Wikipedia and training data. Similarly the test set was also practiced in a manner similar to the maximum entropy model for affective classification. At the same time, the training set and test set are divided by the K-fold method. Based on the maximum entropy classification of probabilistic and latent semantic analysis, words

are classified using important affective classification features, such as the meaning of words and parts of speech in context, that is, the degree of similarity between adverbs and typical affective words.

Manek et al. (2017) in their study, they proposed a statistical method that uses the weight of the Gini index to select features in sentiment analysis by using multiple datasets to improve the accuracy of prediction of sentiment analysis as well. They compared their proposed SVM model with other predictive classifiers for sentiment analysis of movie reviews.

Bhavitha et al. (2017) in their work they indicated that supervised learning methods such as, NB and SVM are the typical techniques for sentiment analysis. As compared to other classifiers, the accuracy of SVM is provides admirable. According to their study they concluded that for small data sets NB performance is good, while for large data sets SVM works good. Lexical based approaches and MaxE achieves good results as well, but due to manual effort and affected by inappropriate adjustment can influence the accuracy.

Go et al. (2009) in their work they used unigrams, bigrams, unigrams and bigrams, and parts of speech as features and they further used ML algorithms such as NB, SVM and MaxE. They used bigram because such tweets used contained negated phrases, such as "/not good" and "/not bad". They were facing the problem in the sparseness tweets such as: "*@stellargirl I loooooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right"*. Due to this matte the bigram was not producing good results by using MaxE. They applied unigram and bigram for different proposed algorithm. In our proposed preprocessing model we can avoid such issues and can get better results.

6 Conclusion and future work

In this research, we studied the impact of different preprocessing steps on the accuracy of three machine learning algorithms for sentiment analysis. We designed an experiment on a dataset of 359 documents. We removed emoticons and applied the Biagram technique as preliminary steps in order to see the actual impact of the preprocessing steps. We applied preprocessing techniques including removal of stopwords, stemming, and word vector on the dataset. The three well-known machine learning classifiers including Naïve Bayes, maximum entropy, and support vector machines were applied on the dataset for sentiment classification. We calculated accuracy of the three machine learning algorithms before and after applying the preprocessing steps. Results proved that the accuracy of Naïve Bayes algorithm was significantly improved after applying the preprocessing steps. Slight improvement in accuracy of support vector machine algorithm was seen after applying the preprocessing steps. Interestingly, in case of maximum entropy algorithm, no improvement in accuracy was seen. We compared our results with Go et al. (2009). Results proved that in case of Naïve Bayes algorithm, accuracy was again significantly high than any other machine learning algorithm after applying the preprocessing steps; followed by maximum entropy and support vector machine algorithms. This research work proves that text preprocessing impacts the accuracy

of machine learning algorithms. It further concludes that in case of Naïve Bayes algorithm accuracy has significantly improved after applying text preprocessing steps. We believe that every preprocessing step has its individual impact on the accuracy of a machine learning algorithm. In our future work, we will add a few more preprocessing steps for example, removal of numbers, spell checking, removal of punctuation marks, lemmatization, single case transformation and so on to our experiment and discover how much is the individual impact of the each preprocessing step in terms of the accuracy of a machine learning algorithm.

References

Asghar MZ, Khan A, Ahmad S, Qasim M, Khan A (2017a) Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. PLoS ONE 12:1–23

Asghar MZ, Khan A, Bibi A, Kundi FM, Ahmad H (2017b) Sentence-level emotion detection framework using rule-based classification. Cogn Comput 9(6):868–894

Baradad VP, Mugabushaka A (2015) Corpus specific stop words to improve the textual analysis in scientometrics. In: International Conference on Science in Information, pp 999–1005

Bhavitha BK, Rodrigues AP, Chiplunkar NN (2017) Comparative study of machine learning techniques in sentimental analysis. In: Proceedings of International Conference Inventory Communication Computing Technology ICICCT 2017, No. Icicct, pp 216–221

Chen J, Huang H, Tian S, Qu Y (2009) Expert systems with applications feature selection for text classification with Naïve Bayes. Expert Syst Appl 36(3):5432–5435

Clark A (2003) Pre-processing very noisy text. In: Proceeding of Work Shallow Process Large Corpora, p 11

Das O, Balabantaray RC (2014) Sentiment analysis of movie reviews using POS tags and term frequencies. Int J Ldots 96(25):36–41

Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. Processing 150(12):1–6

Khan A, Asghar MZ, Ahmad H, Kundi FM, Ismail S (2017) A rule-based sentiment classification framework for health reviews on mobile social media. J Med Imaging Health Inf 7(6):1445–1453

Lovins JB (1968) Development of a stemming algorithm. Mech Transl Comput Linguist 11:22–31

Manek AS, Shenoy PD, Mohan MC, Venugopal KR (2017) Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. World Wide Web 20(2):135–154

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. Arxiv, pp 1–12

Minanovic A, Gabelica H, Krstic Z (2014) Big data and sentiment analysis using KNIME: online reviews vs. social media. In: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp 1464–1468

Mubarok MS, Adiwijaya, Aldhi MD (2017) Aspect-based sentiment analysis to review products using Naïve Bayes. In: AIP Conference Proceedings, vol. 020060, p 020060

Nigam K, Lafferty J, Mccallum A (1999) Using maximum entropy for text classification. In: IJCAI-99 workshop on machine learning for information filtering, pp 61–67

Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation, pp 1320–1326

Riloff E, Qadir A, Surve P, Silva LD, Gilbert N, Huang R (2013) Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, No. Emnlp

Rong X (2014) word2vec parameter learning explained continuous bag-of-word model, pp 1–21

Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66

Xie X, Ge S, Hu F, Xie M, Jiang N (2017) An improved algorithm for sentiment analysis based on maximum entropy. Soft Comput. https://doi.org/10.1007/s00500-017-2904-0

Yadav MP, Pandya D (2017) SentiReview: sentiment analysis based on text and emoticons. In: International Conference Innovation Mechanical Industry Application ICIMIA 2017 SentiReview, no. Icimia, pp 467–472

Saqib Alam Ph.D., Scholar at Dalian University of Technology got his Master degree from Abasyn University of Peshawar and Bachelor degree from Islamia Collage University Peshawar. His main research areas are, Text Analysis and Machine Learning.

Nianmin Yao Ph.D., born in 1974. He is now a Professor in Dalian University of Technology, Dalian, China. He has been a visiting scholar in University of Connecticut in 2010. He is a senior member of China Computer Federation. He Got the Bachelor, Master and doctor degree from Jilin University. His main research interests include network storage, system performance analysis, theory of computation, wireless sensor networks and network storage.