An Effective Framework for Tweet Level Sentiment Classification using Recursive Text Pre-Processing Approach

Muhammad Bux Alvi*†1, Naeem A. Mahoto†2, Mukhtiar A. Unar†3 and M. Akram Shaikh‡4
*Computer Systems Engineering, IUB, Bahawalpur, Pakistan
†MUET, Jamshoro, Pakistan
‡PASTIC National Center, Pakistan Science Foundation, Islamabad, Pakistan

***Abstract*—With around 330 million people around the globe tweet 6000 times per second to express their feelings about a product, policy, service, or an event. Twitter message majorly consists of thoughts. Thoughts are mostly expressed as a text and it is an open challenge to extract some insight from free text. The scope of this work is to build an effective tweet level sentiment classification framework that may use these thoughts to know collective sentiment of the folk on a particular subject. Furthermore, this work also analyses the impact of proposed tweet level recursive text pre-processing approach on overall classification results. This work achieved up to 4 points accuracy improvement over baseline approach besides mitigating feature vector space.**

***Keywords*—Machine learning; recursive text pre-processing; sentiment analysis; sentiment classification framework; Twitter**

I. Introduction

The proliferation of Internet based micro-blogging social networks have opened new avenues to masses to express their response and reaction on variety of topics in real time. People discuss current affairs, complain about a policy, raise voice on a social issue or give feedback about any product or service. This scenario is instigating unremitting pouring of data from the users. It is estimated that till 2020 there will be about 44 ZB of digital data[1]. Another assessment reports that 80% of available data is unstructured today [1]. With around 330 million active user[2] and 6000 tweets per second, twitter has emerged as a popular medium among people to discuss currently trending topical issues to exhibit their tendencies [2], [3]. However, it is a tedious task to discover and summarize collective popular sentiment from this scaling twitter data. Manual monitoring and analysis of such a huge volume of data may be a highly impractical solution. Therefore, a computational method is the only rescue to this issue and opportunity i-e computer mediated sentiment classification [4] for user generated twitter text data.

To extract meaningful features from the acquired dataset(s), text data needs to be pre-possessed properly because knowledge present in text data is not directly accessible. Text data requires two preliminary steps before its application to a machine learning algorithm: 1) removing trivial and non-discriminating data and 2) Text transformation. Text data especially twitter text is notoriously prone to noise and data sparsity. Text data which already has its own inherent challenges to process and analyze, utilization of informal social media language has added more severity to it. For example informal short form (Internet slang), word-shortening, neologism, spelling variations and elongation [5].

The contribution of this work includes an effective tweet level sentiment classification (TLSC) framework that provides comprehensive steps for twitter sentiment classification and allows to discover sentiment orientation embedded in the tweets. Additionally, this work proposes a 19-step recursive text pre-processing approach, initial version proposed in [6], that results in 1) better data cleaning, and 2) reduction in feature vector space. The recursive pre-processing approach separates out redundant and irrelevant tweets and removes noisy data from the tweets to acquire a cleaner dataset. Cleaned dataset is then prepared for learning model to produce an analytic engine to perform tweet level sentiment classification. We have used Multinomial Naive Bayes, LinearSVC and logistic regression machine learning algorithms with six feature extraction techniques to experiment with baseline pre-processing methods and recursive pre-processing approach. This work consisted of 108 experiments for each machine learning algorithm comparing baseline and recursive approaches with hold-out and k-fold cross validation evaluation indexes. We found Multinomial Naive Bayes and LinearSVC algorithms consistently performing well with ngrams and TFIDF + ngrams feature extraction technique using recursive pre-processing approach.

The extracted results can help an non-government organization (NGO) to begin an awareness campaign or the government in policy making to cope with challenges or opportunities. Tweet level sentiment classification framework is presented in Fig. 1.

This work is organized as: 1) Introduction, 2) Related Work, 3) Methodology, 4) Performance Evaluation Indexes, 5) Results and Discussion, 6) Conclusion, and 7) Future Direction.

II. Related Work

Twitter has been largely used to know about people's choice and interest in politics, sports, social issues or global problems [7], [8]. Research on twitter data is recent. However, sentiment analysis, a broader area of study, is around for two decade which is an application of natural language

[1]https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

[2]https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

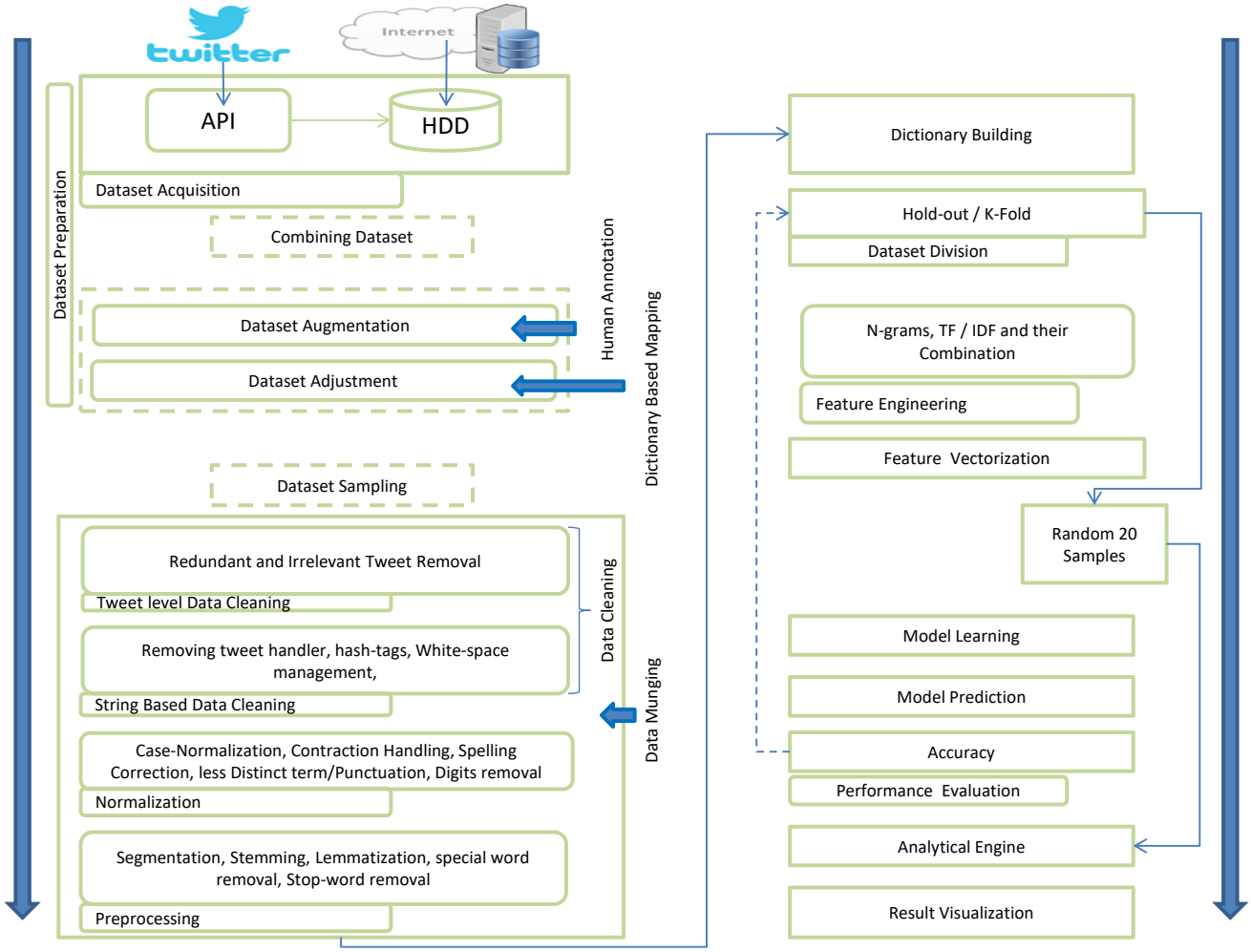

Fig. 1. Tweet Level Sentiment Classification Framework

processing. Most of the work in sentiment analysis, specially in twitter sentiment analysis, revolves around feature extraction. A few researchers have worked on developing a comprehensive framework for twitter sentiment analysis and data pre-processing techniques.

A document level unified framework for tweets classification has been proposed in [9]. The proposed method utilizes four classifiers to handle slang terms, emotions, term orientation and domain specific classifier. They claim to have achieved better results in comparison to other similar work.

In [10], authors reported about the significance of pre-processing and selection of correct pre-processing techniques in sentiment analysis. They have experimented on two datasets with 16 pre-processing techniques using four machine learning algorithms. They found lemmatization, removing digits and contraction handling beneficial and other pre-processing techniques trivial. Their further experiments encompass various combinations of basic pre-processing techniques.

In [11], authors demonstrated to elevate the importance of applying text pre-processing techniques before applying a learning algorithm for twitter sentiment analysis. They have used 05 twitter datasets, 06 pre-processing techniques, two feature models and 04 classifiers including Naive Bayes, support vector machine, Logistic Regression and Random Forest. They have reported that classification efficiency increased by handling contractions and negation but no changes were observed with other steps.

Khan et al., in [12] have addressed the problems of feature vector space i-e data sparsity in tweets. They have concentrated on data pre-processing steps to mitigate data sparsity and to achieve better accuracy.

Kim J. et al. in [13] suggested collaborative filtering method to cope with challenge induced due to sparse data when predicting sentiments in twitter data. They tested their collaborative filtering model on two different datasets and reported it to be quite effective.

Prieto et al. in their work have collected location based tweets about public concern and disease information in Portugal and Spain with supervised signals. They have used regular expressions for feature selection and machine learning for

classification and have achieved F-measure values of 0.8 and 0.9 which are quite promising compared to baseline methods. They have disregarded slang in their work [14].

Many other studies have suggested frameworks for twitter sentiment analysis and assessed the impact of text pre-processing on overall accuracy increase. However, this study offers more practical and comprehensive approach for building twitter sentiment classification system. Additionally we have proposed an ordered recursive pre-processing approach that can handle twitter data well.

III. Methodology: Recursive Pre-processing Approach

The experimental methodology in this paper is organized as: 1) Dataset Preparation, 2) Data Munging, 3) Feature Engineering, 4) Feature Vectorization, and 5) Modeling.

A. Dataset Preparation

1) Data Acquisition: The twitter dataset can be acquired programmatically using twitter STEAMING API or REST API. Alternatively, a twitter dataset may be obtained from an online repository. Two datasets have been acquired externally from an online repository [15]. Global warming dataset describes people's belief whether there is global warming or it is just a myth and over exaggerated matter. The other dataset is about people's acceptability towards self drive cars. Table I represents some statistics about these two datasets. The obtained datasets have majorly two parts i.e. tweets and meta-data.

TABLE I. Dataset Statistics

Dataset	Total Tweets	Positive	Negative	Neutral	Missing Value	Duplicate
Global Warming	6090	3111	1114	1865	0	542
Self Drive Cars	7156	1904	795	4248	209	10

Once the dataset is acquired, the obtained dataset may be augmented with meta-data through human annotation if needed. In this case, the dataset is already annotated but the meta-data in the given dataset is inconsistent as shown in Table II. It consists of all variants for $\{$*Yes, Y, N, yes, Na*$\}$. Therefore, it needs dataset adjustment. Dataset adjustment process includes: A) Managing inconsistent categorical meta-data. B) Handling missing values.

TABLE II. Raw Tweet Data

No	Tweet	Sentiment
1	Ocean Saltiness Shows Global Warming Is Intensifying Our Water Cycle http://bit.ly/bJsszY	Yes
2	RT @sejorg: RT @JaymiHeimbuch: Ocean Saltiness Shows Global Warming Is Intensifying Our Water Cycle	Y
3	Top Climate Scientist Under Fire for 'Exaggerating' Global Warming http://bit.ly/9Pq0gQ	N
4	For #EarthDay Global warming could affect patient symptoms	yes
5	Great article.	Na
6	W8 here is idea. it is natural Climate change not human induced global warming.	n

2) Target Data Adjustment: Dictionary Based Series Mapping: To align such inconsistent and object type data for computational purpose, dictionary based series mapping method has been used to remove inconsistency from the response vector data. An additional dictionary source is developed to handle these inconsistencies. Positive and negative labels are mapped to 1 and 0, respectively as shown in Table III.

TABLE III. Clean Tweet Data

No	Clean Tweet	Sentiment
1	ocean saltiness shows global warming is intensifying our water cycle	1
2	top climate scientist under fire for 'exaggerating' global warming	0
3	earthday global warming could affect patient symptoms	1
4	wait here is idea: it is natural climate change human induced global warming	0
5	wait here is idea it is natural Climate change not human induced global warming	0

3) Target Missing Values Management/Handling: One way to handle tweets that do not have any supervising signal is to disregard them. This may be a feasible solution if the number of missing value tweets is low. Conversely, they may be annotated with an appropriate label.

Let Twitter Datasets (TDS) be the aquired dataset. The redundant, irrelevant and missing value tweets are removed initially to obtain Extracted Twitter Dataset (ETDS). Data cleaning methods are then applied on ETDS to determine Clean Twitter Dataset (CTDS) that is used as an input to learning algorithms after feature engineering and proper transformation.

a) Definition 1: Twitter Dataset TDS. Let ETDS be the extracted twitter dataset, then:

$$ETDS \in TDS \mid ETDS = \{tw_1, tw_2, tw_3, \ldots, tw_n\} \quad (1)$$

Where tw_n represents individual tweet and is represented as;

$$tw = \{tk_1, tk_2, tk_3, \ldots, tk_n\} \quad (2)$$

where tk_n is the individual token in the tweet

b) Definition 2: Clean Twitter Dataset CTDS may be defined as;

$$CTDS \subseteq ETDS \mid CTSD = \{feat_1, feat_2, feat_3, \ldots, feat_n\} \quad (3)$$

where $feat_n \in tk_n$ and represent selected feature(s) from ETDS

B. Data Munging

In this work, we have proposed recursive pre-processing twitter text pre-processing approach in a compact and structured form under the umbrella of Data munging as shown in Fig. 2. Data munging is an essential step to prepare noisy twitter data for text analyses because about 80% of the time and effort for text analyses is consumed for data munging[3]. Experimental work has shown that the proposed recursive

[3]https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says

approach extracts cleaner dataset efficiently. Data Munging includes three major step. Each step involves multiple sub-steps. These three steps are:

1) Data Cleaning
2) Data Normalization
3) Data Pre-processing

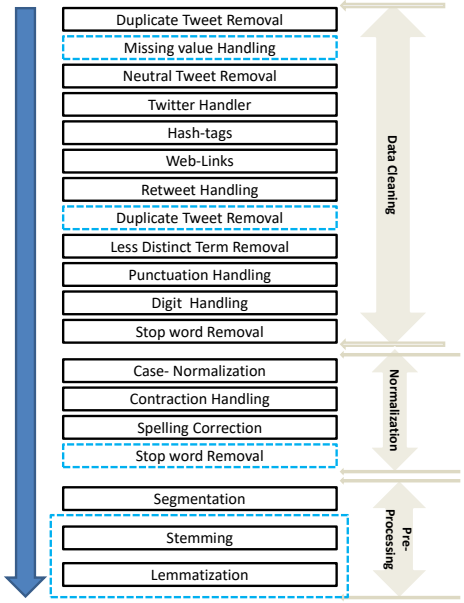

Fig. 2. Proposed Recursive Pre-Processing Approach

1) Data Cleaning: Cleaning of text data is a tedious but necessary step that requires a lot of care. In case of tweet level SA, which involves unprecedented word improvisations needs more attentions. Therefore, twitter data cleaning has become more challenging than traditional text pre-processing. Data cleaning is categorized as:

1) Tweet Level Data Cleaning
2) String Level Data Cleaning

a) Tweet level Data Cleaning: It includes 1) removal of redundant tweets, 2) removal of irrelevant tweets, and 3) removal of unintended tweets. As shown in Fig. 2, tweet level data cleaning is performed initially to get rid of redundant, irrelevant and unintended tweets to obtain CTDS at tweet level.

b) String level Data Cleaning: Twitter-handlers, hash tags, web-links and retweets are removed using regular expressions. Further processing include; word shortening (“w8”, “f9”, “gr8”), elongated terms (“Coooool”), unusual acronyms (“ASAP”), neologism (“webinar”), etc. All of these challenges are handled by creating a dictionary in this work. Additionally punctuation, digits and dataset specific less distinct terms are evicted.

2) Data Normalization: Text Normalization is a multi-step procedure to standardize the tweets.

a) Case-Normalization: It is a non-reversible practice to avoid multiple copies of semantically similar terms. However, this step may be taken carefully with some datasets. For example, case normalization of the term “United Nations” may negatively affect the performance of the learning model.

b) Contraction Handling and Spelling Correction:: Twitter short messages are written in an improvised language developed due to emergence of micro-blogging. Character bound tweets brings a lot of new challenges such as contractions which are informal shortened form of words as shown in Table IV. Contractions are avoided in formal writings but they are extensively used in informal way of expression.

TABLE IV. CONTRACTIONS

No	Normal Contraction	Actual	Negated Contraction	Actual
1	he’s	he is	can’t	can not
2	She’d	She would	was’nt	was not
3	you’ll’ve	you will have	haven’t	have not
4	y’all	you all	ynt	why not
5	y/n	yes or no	idonno — idunno	i do not know

3) Data Pre-processing: Some of data processing operations may have minimum incremental impact on the overall classification accuracy but these steps surely reduce feature vector space which is beneficial in improving estimation and execution time.

a) Word Segmentation: Given a tweet, splitting it into a list of words is referred as word segmentation or tokenization. We have used NLTK (version 3.2.5) tokenizer to segment the tweet into tokens.

b) Stemming / Lemmatization: It is a mapping task that maps different forms of verbs and nouns into a single semantically similar word. Stemming works on the principle of chopping off trailing character(s) from given word to reach base-form. Depending on the usage of stemmer genre, the converted base-form may be incorrect linguistically but works effectively for sentiment classification. Porter Stemmer algorithm [16], [17] has been used in this work for stemming. Optionally lemmatization may be used for this purpose with increased time complexity.

c) Language stop words: These terms rarely possess any sentiment significance, therefore they are discarded. We have used natural language toolkit (NLTK) library for this purpose [18], [19] has deeply observed the impact of stop word on twitter sentiment classification.

Fig. 3 represents six graphs in pair i-e (1a, 1b), (2a, 2b) and (3a, 3b). 1a, 2a, and 3a show dataset statistics before data munging while 1b, 2b, and 3b display statistics after applying recursive data pre-procesing approach. In Fig. 3, 1(a) shows that few tweets in TDS have more than 140 characters that show lacking in data acquisition process. We infer that some unnecessary and irrelevant terms or characters have been padded into some tweets. 2(a) displays sentiment-wise distribution and 3(a) represents group wise distribution of tweets based on their frequency. 1(b), 2(b), and 3(b) show CTDS after applying recursive text pre-processing method i-e data munging. Extra characters have been deleted and there is no tweet having more than 140 characters as shown in 1(b), redundant tweets have been evicted as given in 2(b), and

3(b) shows group-wise tweet distribution and filtration. Fig. 3 graphically displays impact of recursive pre-processing on global warming dataset. Similar pre-processing is also applied on other dataset as well.

The resultant cleaned dataset needs to be split into training dataset and testing dataset for model learning and model evaluation purpose. There are two popular approaches to perform this division 1) Hold out Method, 2) K-fold method. These two strategies aim to determine the best model and the best parameters for the model and to estimate its suitability on out-of-sample data.

C. Feature Engineering

Features are the distinct measurable attributes in each input data sample. Feature preparation or engineering is combo process of feature extraction and feature selection. Feature extraction involves determination of all those input values that may describe the given object i-e label. While feature selection results in the minimum feature set that may best describe the same object. Each term in the twitter dataset can be candidate for being a feature. We have used ngrams and weighted versions of ngrams to test their suitability for tweet level sentiment classification with machine learning method as detailed in Table V and Table VI.

a) Unigrams: A single distict term in the dataset is referred as unigram. However, all unigrams cannot be selected as the features. With $'n'$ actual number of unigram and $'m'$ selected unigrams, the following always stands true for unigrams feature selection method;

$$m_{Sel\_feat} \subseteq n_{Act\_feat} \mid m_{Sel\_feat} \leq n_{Act\_feat} \tag{4}$$

This is a common but most popular approach. The downside of this method is that it looses the order of the term and just count them but in practice it produces good results.

b) N-grams: An n-gram is a sequence of n-neighboring tokens. Bi-grams having two and tri-grams with three adjacent tokens. N-grams approach covers the disadvantages of of unigram approach i.e. order is preserved, at least, at n-terms phrase level. This advantage, referred as capturing of partial contextual meaning, costs some complexity. For example, in case there are just 10000 tokens in the feature vector and bigrams approach is applied then we may end up with a huge number of tokens (all unigrams + bigrams). With trigram, the number of tokens may increase at least two-fold. The equation 5 calculates the number of ngrams produced given the selected features for $ngrams_{bigrams}$ and $ngrams_{trigrams}$, respectively.

$$ngrams = \begin{cases} ngrams_{bigrams} = (2 * m_{Sel\_feat}) - 1 & ; m_{Sel\_feat} > 1 \\ ngrams_{trigrams} = (2 * m_{Sel\_feat}) + i & ; n > 2 \\ & i = \{0, 1, 2, \ldots, n\} \end{cases} \tag{5}$$

c) TF/IDF: It is a weighted method that measures the significance of a feature in the document and in the dataset. N-grams approach is prone to overfit due to its capacity to increase the number of features exponentially. Usage of TFIDF handles high and low frequency ngrams implicitly. High frequency n-grams do not help to discriminate tweets while low frequency n-grams are likely to overfit. Medium frequecy n-grams are more likely to help in classification. The problem of sparse terms can be controlled by using n-grams approach with TF/IDF. It is mathematically denoted as:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \tag{6}$$

Actually, equation 6 combines two techniques: 1) Term Frequency (tf), and 2) Inverse Document Frequency (idf).

d) Term Frequency (tf): It is simply the count of the number of occurrences of a particular term in a document. Document here refers to a single tweet i-e $(f_{t,d})$. This gives higher weight to terms that are frequent in a tweet. The equation 7 represents term frequency in the normalized form.

$$tf = \frac{f_{t,d}}{\sum\limits_{t' \in d} f_{t',d}} \tag{7}$$

e) Inverse Document Frequency (idf): Document frequency (df) is computed at dataset level. Document frequency is the ratio of total number of tweets where term "t" appear to the total number of tweets in the given dataset and is represented as:

$$df = \frac{\mid d \in D : t \in d \mid}{\mid D \mid} \tag{8}$$

Accordingly, the idf is the inverse of df and may be denoted as;

$$idf = \frac{\mid D \mid}{\mid d \in D : t \in d \mid} \tag{9}$$

And its normalized equation is given by;

$$idf = log(\frac{\mid D \mid}{\mid d \in D : t \in d \mid}) \tag{10}$$

idf is biased toward unusual and more distinct terms in the dataset. Overall, a term achieves high weight using $tfidf$ when its tf is high and its df is low. This method extracts more discriminating features in the tweet that are not so frequent in the whole dataset.

D. Feature Vectorization

Tweets are unstructured data in nature. Unstructured features cannot be used as direct input to a machine learning algorithm for building a model. Feature vectorization is an important task that converts the extracted text features into numeric feature matrix to be used for model estimation and prediction. Feature vectorization replaces each piece of text i.e. tweet with a huge number vector. Each number dimension of that vector corresponds to a certain token in the dataset.

E. Modeling

Modeling refers to model learning and model evaluation process. Supervised algorithms take a training subset and learn mapping of given feature to respective target values. In other words, the supervised learning algorithms learn by estimating their internal parameters from given examples. These parameters may then be used with out-of-sample data instances to predict the targets as shown in equation 11:

$$TSC : T_w \rightarrow C_{pos|neg} \tag{11}$$

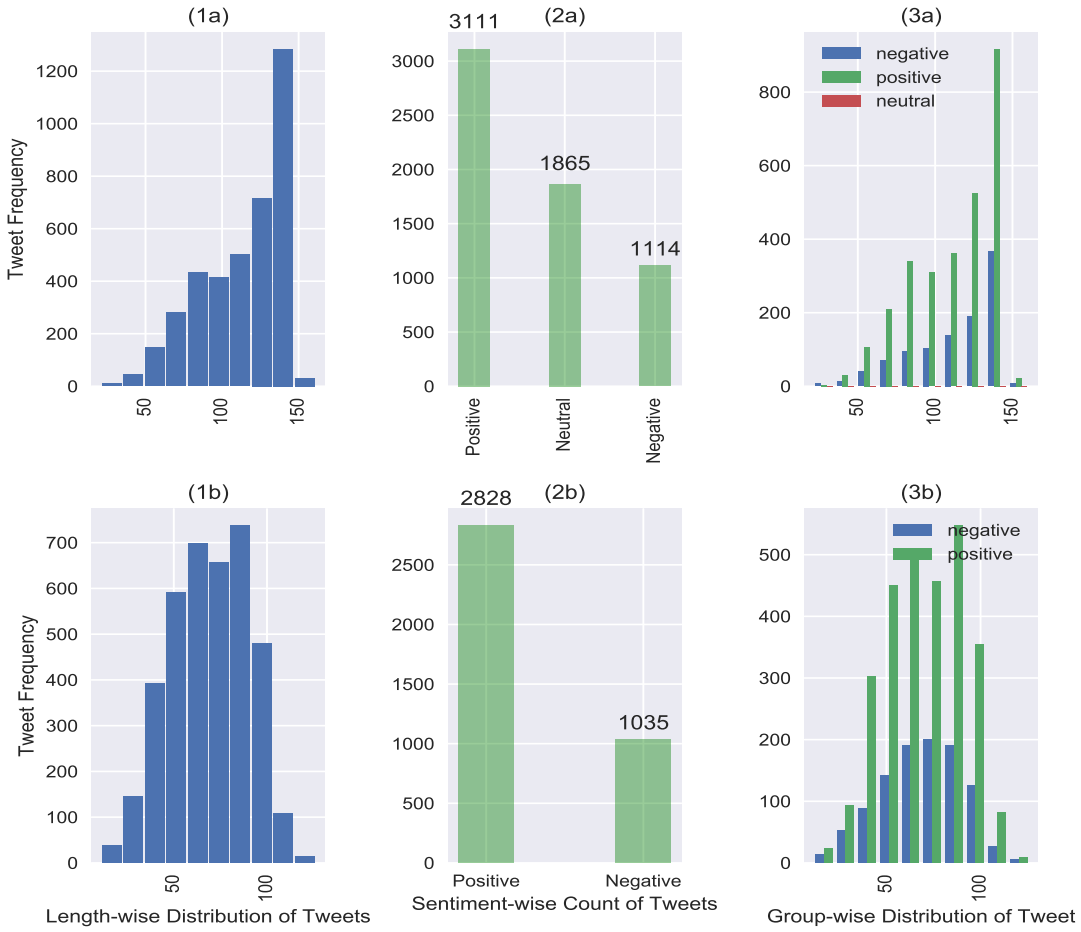

Fig. 3. Graphical representation of Impact of Recursive Text Pre-processing on Tweets - Global Warming Dataset

where TSC represents Twitter Sentiment Classifier, t_w is the input tweet to be assigned either class while C may be any of the two possible categories i-e positive or negative. Moreover, the input feature set must fulfill four prerequisites:

1) input features and the corresponding labels be stored separately,
2) both objects be numeric,
3) both be numpy array, and
4) their dimension must comply to each other.

1) Learning Algorithm: We have used Multinomial Naive Bayes, LinearSVC and logistic regression, most popular machine leaning algorithms for text analyses, in this work [10], [11], [20], [21]. Multinomial Naïve Bayes is one of the most widely used probabilistic models. It takes into account the frequency of features in each twitter text communication (ttc_n) and represents it as Vector Space Model. This technique outperforms the Bernoulli probabilistic model and all of its variations if the vector space is high. LinearSVC determines optimal decision boundary that is the hyperline having highest margin from the given sample in the extracted twitter data dataset (ETDS). All the tw_n with given margin form the hyperline are the support vectors that specify the correct location of the hyperline. If the tw_n are linearly inseparable then a hyperline is determined such that there is minimum loss in accuracy. This technique is robust to high dimension datasets. Logistic Regression is another widely used dichotomous Machine Learning algorithms to describes and estimates the dependent variable using input feature vector. Logistic regression algorithm utilizes sigmoid function and learning is performed through maximum likelihood.

2) Model Training: Training dataset (T_rDS) is used for model learning to develop a classifier. Model learning process refers to building up of patterns based upon extracted feature set and updating of learning algorithm's internal parameters. Given a supervised machine learning algorithm (S), trained on T_rDS, we build a sentiment classifier (F) such that

$$S(T_rDS) = F \tag{12}$$

3) Model Prediction: Model prediction refers to the process of predicting class labels for out-of-sample data of test data. This process is usually used as preliminary stage for model evaluation in which test data is normalized and features are extracted to be fed into trained classifier. The trained classifier receives out-of-sample tweet denoted as tw_{nd} and predicts its class c_{nd} based on previously learned patterns. While making prediction, the trained model will ignore all those tokens of the new tweet(s) that were not learned during model building process. This is the very reason that to have

more data during learning process is always recommended. Model prediction process may be represented as:

$$F(tw_{nd}) = c_{nd} \tag{13}$$

IV. Performance Evaluation Indexes

We already have expert annotated true class labels for the test dataset T_eDS. Now with predicted class labels, we can compare true class labels and corresponding predicted class labels to evaluate the efficiency of the developed sentiment classification model using different model evaluation metrics such as Accuracy. There is no hard and fast rule for selection of an evaluation criteria. Actually it depends on the requirement of problem and the dataset.

A. Hold out Method

This strategy is computationally inexpensive and needs to run once only. Now given the clean twitter dataset, it is split into training dataset (T_eDS) and testing dataset (T_rDS) in a suitable proportions as shown in Table V and Table VI.

$$CTDS = \begin{cases} T_rDS = \{(tw_1, c_1), (tw_2, c_2), \ldots, (tw_n, c_m)\} \\ T_eDS = \{tw_1, tw_2, \ldots, tw_n\} \end{cases} \tag{14}$$

where tw_n and c_m denote individual tweet and corresponding label. The dataset subsets consisting of (80-20)% to (60-40)% train-test ratio are tested for suitability and their results are given in Table V and VI. This is more comprehensive approach but prone to test data leakage that may cause decrease in final classification model accuracy.

B. Cross Validation Method

In this division technique, feature set is divided into k equal parts. For the first iteration the model is fitted with $k-1$ parts of the given feature set and computes the fitted model prediction error on the k^{th} left out part. This process occurs k times and results are averaged to get the over all conclusion. This is computationally an expensive strategy but less biased method and not prone to data leakage. Furthermore, If k-fold split does not evenly separates the dataset, then one group will have remainder of the dataset.

a) Accuracy: Accuracy is a popular evaluation measure. Mathematical form for accuracy is given by

$$Accuracy = \frac{CorrectPredictionsMade}{TotalNo.ofPredictionsMade} * 100 \tag{15}$$

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} * 100 \tag{16}$$

where TP denotes true positive, TN represent true negative, FP is false positive and false negative is given by FN

V. Results and Discussion

Table V and Table VI represent detailed experimental results for the two datasets using Multinomial Naive Bayes, LinearSVC and logistic regression algorithms. Various dataset division strategies for hold out and cross validation methods have been used to evaluate the impact of tweet level sentiment classification framework and recursive pre-processing approach by comparing accuracy achieved through baseline and recursive pre-processing technique.

As shown in Fig. 4, using global warming dataset, this work with recursive pre-processing approach, achieved about 4-points accuracy rise in comparison to baseline using ngrams features. Multinomial Naive Bayes and LinearSVC consistently shown better results. With TFIDF + ngrams models, Multinomial Naive Bayes algorithm outperformed other algorithms with most dataset division strategies.

Fig. 5, demonstrates that this work attained above 4-point accuracy increase using ngrams approach with self drive car twitter dataset. Here, Multinomial naive bayes algorithm was consistent to produce better results. With TFIDF + ngrams models, LinearSVC proved better. Trivial degradation in accuracy was also observed occasionally with this dataset

VI. Conclusion

In this work, we have proposed an effective and comprehensive tweet level sentiment classification framework with recursive twitter data pre-processing approach. This framework encompasses all the necessary steps involved from twitter dataset acquisition to classification results generation as shown in Fig. 1. Moreover, a 19-step recursive twitter data pre-processing approach is presented that covers all necessary twitter data munging operations in an ordered form. Couple of steps, duplicate tweets and stop word removal, may be required to be retaken for handling regenerated text segments. Moreover, it is observed that a few data munging step may not have significant impact on classification efficiency but they mitigate the issue of feature vector space that makes the process computationally efficient. For example, common punctuation, neologism and digits. Further investigation of this work concludes that Multinomial Naive Bayes and LinearSVC algorithms showed consistently better performance with ngrams and TFIDF + ngrams feature extraction methods using proposed recursive pre-processing approach, achieving up to 4 point accuracy improvement in comparison to baseline models.

VII. Future Direction

Future experimental investigation with this framework and proposed recursive pre-processing approach may include application of advanced machine learning algorithms to check their suitability. Furthermore, text pre-preprocessing techniques may be tested separately as well as combined together to determine the best text pre-processing pipeline configuration for sentiment classification.

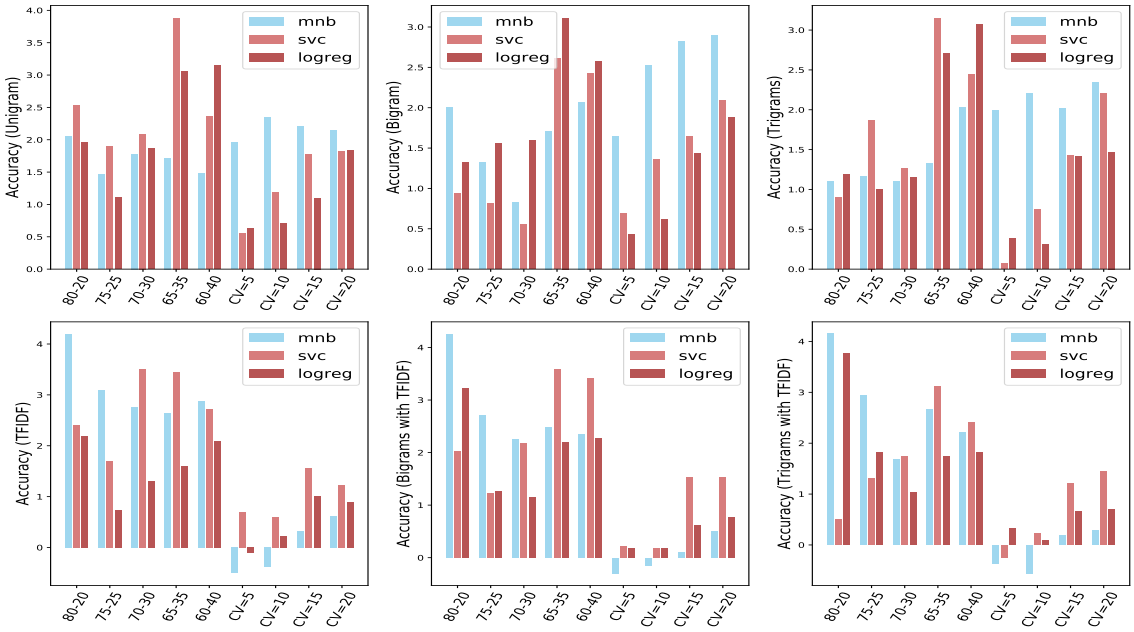

Fig. 4. Global Warming Twitter Dataset - Impact of Recursive Pre-processing approach using;
(**a**) unigrams(**b**) bigrams (**c**) trigrams (**d**) tfidf+unigrams (**e**) tfidf+bigrams (**f**) tfidf+trigrams

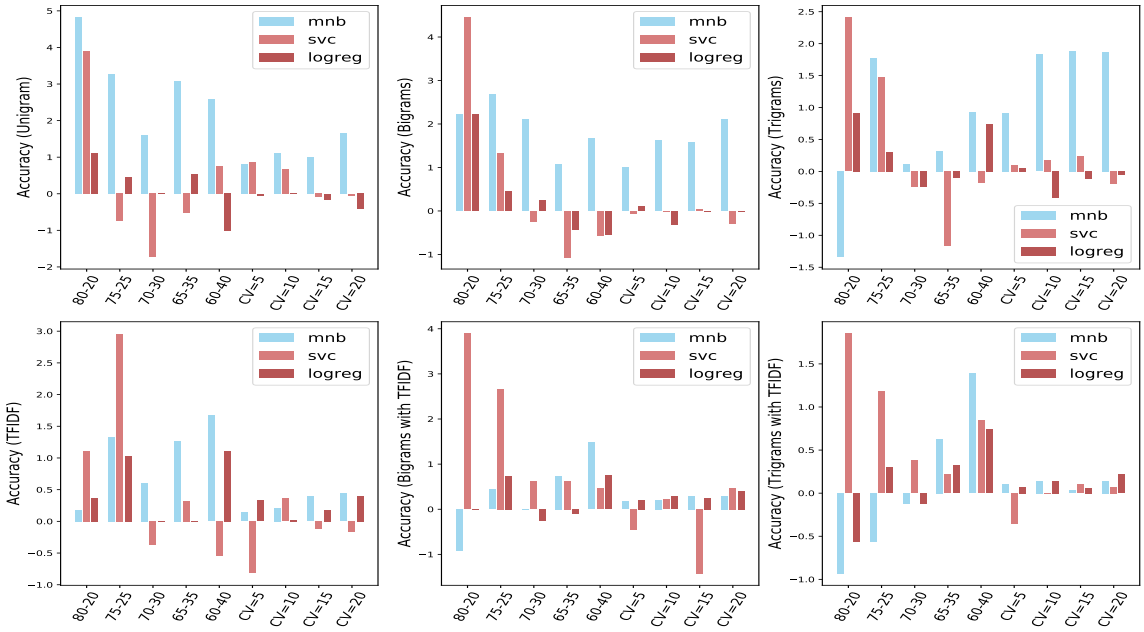

Fig. 5. Self Drive Car Twitter Dataset - Impact of Recursive Pre-processing approach using
(**a**) unigrams (**b**) bigrams (**c**) trigrams (**d**) tfidf+unigrams (**e**) tfidf+bigrams (**f**) tfidf+trigrams

TABLE V. **DATASET1-GLOBAL WARMING:** ACCURACY ANALYSIS OF TWEET LEVEL SENTIMENT CLASSIFICATION MODEL

		Hold-out										K-Fold Cross-Validation							
		BaseLine					Recursive Pre-processing					BaseLine				Recursive Pre-processing			
		(80-20)%	(75-25)%	(70-30)%	(65-35)%	(60-40)%	(80-20)%	(75-25)%	(70-30)%	(65-35)%	(60-40)%	CV=05	CV=10	CV=15	CV=20	CV=05	CV=10	CV=15	CV=20
MNB	Unigrams	81.50	81.88	82.13	81.44	81.95	83.55	83.34	83.91	83.16	83.43	77.58	77.80	78.01	78.37	79.54	80.15	80.21	80.52
	Bigram	82.01	82.50	83.00	82.33	81.95	84.02	83.82	83.83	84.04	84.02	77.34	76.84	77.10	77.43	78.98	79.37	79.92	80.33
	Trigram	82.92	83.12	82.65	81.96	81.75	84.02	84.29	83.75	83.29	83.78	74.55	74.67	74.54	74.60	76.54	76.88	76.56	76.94
	TF/IDF (Unigrams)	77.10	77.12	76.96	76.86	76.58	81.30	80.22	79.73	79.51	79.46	77.14	77.42	77.71	77.86	76.63	77.04	78.03	78.48
	TF/IDF(Bigrams)	76.45	77.22	77.39	76.49	76.64	80.71	79.94	79.65	78.97	78.99	76.08	76.52	77.24	77.21	75.78	76.37	77.34	77.72
	TF/IDF (Trigrams)	76.32	77.01	77.48	76.64	76.84	80.47	79.94	79.17	79.31	79.05	75.69	76.39	76.80	77.19	75.33	75.83	76.98	77.48
Linear SVC	Unigrams	80.07	80.12	79.29	78.27	79.88	82.60	82.02	81.38	82.15	82.24	78.17	78.53	78.63	79.09	78.72	79.71	80.40	80.91
	Bigram	80.59	80.74	80.75	79.60	79.81	81.53	81.55	81.30	82.21	82.24	79.21	79.10	79.10	79.38	79.90	80.46	80.74	81.47
	Trigram	81.11	80.43	80.75	79.74	80.27	82.01	82.30	82.01	82.89	82.72	79.62	79.83	79.57	79.72	79.69	80.58	81.00	81.92
	TF/IDF (Unigrams)	81.37	81.26	81.19	80.33	81.17	83.78	82.97	84.70	83.77	83.90	79.26	79.72	79.75	80.36	79.95	80.32	81.31	81.59
	TF/IDF (Bigrams)	81.75	82.40	82.05	81.07	81.56	83.78	83.63	84.22	84.65	84.97	79.47	79.98	79.65	80.21	79.69	80.16	81.19	81.74
	TF/IDF (Trigrams)	83.05	82.71	82.57	81.74	82.21	83.55	84.01	84.30	84.85	84.61	78.77	79.08	79.34	79.77	78.53	79.31	80.55	81.22
Log.Regression	Unigrams	81.11	81.67	80.93	79.97	80.40	83.07	82.78	82.80	83.02	83.55	79.10	79.49	79.93	79.74	79.73	80.20	81.03	81.57
	Bigram	81.75	81.88	81.53	80.93	80.98	83.07	83.44	83.12	84.04	83.55	79.80	80.11	80.11	80.47	80.23	80.73	81.55	82.35
	Trigram	81.88	81.78	82.05	80.85	80.59	83.07	82.78	83.20	83.56	83.66	79.57	80.11	79.90	80.62	79.95	80.42	81.31	82.09
	TF/IDF (Unigrams)	79.81	80.53	80.24	79.45	78.97	82.01	81.26	81.54	81.06	81.06	77.71	78.07	78.14	78.43	77.60	78.29	79.16	79.33
	TF/IDF (Bigrams)	78.78	79.81	80.15	79.08	78.91	82.01	81.07	81.30	81.27	81.18	76.41	76.96	77.27	77.39	76.59	77.15	77.88	78.17
	TF/IDF (Trigrams)	78.13	78.98	79.63	78.93	78.84	81.89	80.79	80.67	80.66	80.65	75.51	76.05	76.49	76.62	75.83	76.14	77.15	77.32

TABLE VI. **DATASET2-SELF-DRIVE CARS:** ACCURACY ANALYSIS OF TWEET LEVEL SENTIMENT CLASSIFICATION MODEL

MNB	Unigrams	73.14	75.11	76.04	75.13	75.00	77.96	78.37	77.65	78.20	77.59	74.99	75.09	74.87	75.02	75.79	76.20	75.86	76.67
	Bigram	73.14	73.92	75.06	75.76	75.55	75.37	76.59	77.16	76.82	77.22	73.69	73.69	73.28	73.35	74.68	75.31	74.86	75.45
	Trigram	72.59	73.77	75.43	75.55	75.18	74.25	75.55	75.55	75.87	76.11	69.32	68.77	67.57	68.02	70.23	70.60	69.45	69.89
	TF/IDF (Unigrams)	70.37	71.11	71.85	71.95	71.66	70.55	72.44	72.46	73.22	73.33	71.13	71.32	71.32	71.28	71.27	71.53	71.71	71.72
	TF/IDF (Bigrams)	70.37	70.96	71.72	71.64	71.20	69.44	71.40	71.72	72.38	72.68	70.58	70.65	70.65	70.61	70.75	70.86	70.94	70.90
	TF/IDF (Trigrams)	70.37	70.96	71.60	71.64	71.11	69.44	71.40	71.48	72.27	72.50	70.54	70.61	70.72	70.65	70.64	70.75	70.75	70.79
LinearSVC	Unigrams	73.14	77.18	77.28	76.08	75.37	77.03	76.44	75.55	75.55	76.11	73.87	73.54	73.84	73.99	74.72	74.20	73.75	73.94
	Bigram	73.88	76.29	77.16	77.88	77.59	78.33	77.62	76.91	76.82	77.03	74.95	75.36	74.87	75.25	74.90	75.35	74.90	74.97
	Trigram	73.70	74.96	76.29	77.46	76.66	76.11	76.44	76.04	76.29	76.48	73.32	74.28	73.73	74.69	73.42	74.46	73.97	74.49
	TF/IDF (Unigrams)	75.92	76.59	77.77	76.71	77.12	77.03	77.92	77.40	77.03	76.57	75.61	75.10	75.17	75.24	74.79	75.46	75.05	75.08
	TF/IDF (Bigrams)	74.81	75.25	76.79	77.88	77.96	78.70	77.62	77.40	78.51	78.42	74.36	74.76	74.65	75.09	73.90	74.98	75.23	75.56
	TF/IDF (Trigrams)	74.07	75.25	76.41	77.24	77.03	75.92	76.14	76.79	77.46	77.87	73.47	73.50	73.21	73.46	73.12	73.49	73.31	73.53
Log.Regression	Unigrams	74.81	76.29	77.03	77.24	77.96	75.92	76.74	77.03	77.77	76.94	74.36	74.95	74.54	75.21	74.31	74.97	74.38	74.79
	Bigram	73.33	74.96	76.41	77.77	76.66	75.55	75.40	76.66	77.35	76.11	73.76	74.73	74.58	74.87	73.86	74.42	74.57	74.86
	Trigram	73.33	74.81	75.55	75.76	75.64	74.25	75.11	75.30	75.66	76.38	73.10	74.13	73.95	74.24	73.16	73.71	73.83	74.19
	TF/IDF (Unigrams)	71.85	72.74	73.82	74.07	73.24	72.22	73.77	73.82	74.07	74.35	71.98	72.47	72.61	72.61	72.31	72.49	72.79	73.01
	TF/IDF (Bigrams)	70.92	71.70	72.59	72.80	72.12	70.92	72.44	72.34	72.69	72.87	70.80	70.95	70.98	70.91	71.08	71.23	71.23	71.31
	TF/IDF (Trigrams)	70.74	71.55	72.09	72.06	71.66	70.18	71.85	71.97	72.38	72.40	70.61	70.69	70.61	70.61	70.68	70.83	70.75	70.83

REFERENCES

[1] G. Miner, J. Elder IV, A. Fast, T. Hill, R. Nisbet, and D. Delen, *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.

[2] V. Kagan, A. Stevens, and V. Subrahmanian, "Using twitter sentiment to forecast the 2013 pakistani election and the 2014 indian election," *IEEE Intelligent Systems*, vol. 30, no. 1, pp. 2–5, 2015.

[3] P. Burnap, R. Gibson, L. Sloan, R. Southern, and M. Williams, "140 characters to victory?: Using twitter to predict the uk 2015 general election," *Electoral Studies*, vol. 41, pp. 230–233, 2016.

[4] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.

[5] S. Brody and N. Diakopoulos, "Cooooooooooooooollllllllll-lll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 562–570.

[6] M. B. Alvi, N. A. Mahoto, M. Alvi, M. A. Unar, and M. A. Shaikh, "Hybrid classification model for twitter data-a recursive preprocessing approach," in *2018 5th International Multi-Topic ICT Conference (IMTIC)*. IEEE, 2018, pp. 1–6.

[7] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[8] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, 2011, pp. 30–38.

[9] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, "T-saf: Twitter sentiment analysis framework using a hybrid classification scheme," *Expert Systems*, vol. 35, no. 1, p. e12233, 2018.

[10] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Systems with Applications*, vol. 110, pp. 298–310, 2018.

[11] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[12] F. H. Khan, S. Bashir, and U. Qamar, "Tom: Twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245–257, 2014.

[13] J. Kim, J. Yoo, H. Lim, H. Qiu, Z. Kozareva, and A. Galstyan, "Sentiment prediction using collaborative filtering," in *Seventh international AAAI conference on weblogs and social media*, 2013.

[14] V. M. Prieto, S. Matos, M. Alvarez, F. Cacheda, and J. L. Oliveira, "Twitter: a good place to detect health conditions," *PloS one*, vol. 9, no. 1, p. e86191, 2014.

[15] Datasets. [Online]. Available: https://www.figure-eight.com/data-for-everyone

[16] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[17] P. Willett, "The porter stemming algorithm: then and now," *Program*, vol. 40, no. 3, pp. 219–223, 2006.

[18] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[19] H. Saif, M. Fernández, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of twitter," 2014.

[20] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[21] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.