Tweet Normalization: A Knowledge Based Approach

Itisha Gupta[1], Nisheeth Joshi[2]

[1]*Research Scholar, Department of Computer Science, Banasthali University, Rajasthan, India*
itishagupta07@gmail.com
[2]*Associate Professor, Department of Computer Science, Banasthali University, Rajasthan, India*
nisheeth.joshi@rediffmail.com

Abstract: Twitter Sentiment Analysis has attracted a lot of attention recently due to its promising commercial advantages. Data pre-processing is a leading and fundamental step towards sentiment analysis since it may lead to increase the correctly classified instance. The occurrence of slangs, misspelled words, hashtags, URLs, emoticons etc. in tweets make Twitter text notoriously noisy and unstructured thus sentiment analysis of Twitter text is a challenging task. So focus of this paper is to highlight pre-processing significance that would normalize and clean tweets for sentiment classification fortification. In this research work, we explore various pre-processing methods and presented a framework of the pre-processing system with a detailed description of the process which comprises of 2 segments: denoising such as removal of StopWords, URLs, username, punctuation etc. and normalization such as conversion of Non-standard words to their canonical forms. Tweets are pre-processed by normalizing elongated words (loooove to looove), misspelled words (kkk to okay), informal acronyms (rofl to rolling on the floor laughing), negation handling, emoticon replacement, etc. An evaluation is also carried out for analyzing the performance of the pre-processing system by comparison of manually pre-processed tweets and automatic pre-processed tweets and report 87.6% accuracy of proposed pre-processing method on 1000 tweets of demonetization, 90.5% accuracy on 200 tweets of iphone7 and 88.08% combined accuracy on 1200 tweets (1000 of demonetization and 200 of iphone7).

Keywords: Non-standard; Normalization; Pre-processing; Sentiment analysis; Twitter;

I. INTRODUCTION

Social media ubiquitous nature is one of the big potential sources for extraction of crowd opinion. Among the various microblogging websites, there is an abrupt usage of Twitter by people for posting their opinionated view. People tweets on each and every topic such as sports, politics etc. Such opinionated views are very valuable for organizations so they are seeking the ways of mining Twitter data to know about customer's opinion on their product and services. One such technique of mining opinion is sentiment analysis which is a machine learning technique that is used to classify the people sentiments towards any topic or contextual polarity of any text (containing people opinions) by using natural language processing techniques, computational linguistics, and text analysis. Machine learning algorithm success depends on no. of factors and one of the foremost is the quality of corpus. The Twitter text is notoriously noisy and unstructured which is the biggest obstacle in the application of sentiment analysis algorithms. Tweets often contain Non-standard, Non-English words, domain-specific entities, slangs, misspell words etc. which in turn lead to declining in the accuracy of models. Uninformative data like URLs, scripts, HTML tags etc. do not make any sense in analyzing sentiment. Such data increase dimensionality which makes classification task more difficult because each word of a tweet is considered as a dimension in supervised machine learning approach. So polishing and normalization of the collected corpus is needed for getting good results. Data pre-processing is cleaning and normalizing unstructured noisy data and preparing it for classification. It may significantly impact generalization performance of algorithms. It leads to the selection of relevant feature thus reduces data dimensionality resulting in the reduction of I/O overhead. In this paper, we have proposed the pre-processing framework which articulates two major phases: basic cleaning operations and normalization of tweets like normalizing elongated words (huuuury to huuury), informal acronyms (lol to laughing out loud), phonetic substitution (2mrw to tomorrow), expansion of contraction words, emoticon replacement etc. Rest of paper is organized as follows: section 2 presents the literature review of earlier work on normalization. Section 3 describes experimental setup for the proposed pre-processing framework. Section 4 presents the proposed framework of pre-processing. Section 5 discussed evaluation result and the last section concludes this paper with future work.

II. LITERATURE REVIEW

Haddi et al. [1] demonstrated the role of data pre-processing (data transformation and filtering) in increasing accuracy of SVM model on online movie reviews. Their experimental results show the rise in SVM accuracy when appropriate feature selection and representation is done after pre-processing. They obtained the highest accuracy of 93.5 in TF-IDF matrix when data is pre-processed and chi-squared is applied for feature selection. Singh and Kumari [2] described tweet pre-processing especially the slang word. They proposed

978-1-5386-0514-1/17/$31.00 ©2017 IEEE

a technique for normalization of slang word in which firstly coexisting words are gathered for slang and then used n-gram language model for finding out binding and CRM for determining the importance of slang words and results show improvement in SVM accuracy. Kotsiantis et al. [3] addressed data pre-processing issue which can significantly affect machine learning algorithm performance. They described an algorithm for each step of data pre-processing.

Elgamal [4] used a NB classifier for sentiment analysis of tweets collected during Hajji Season. They calculated accuracy, precision and recall for 10, 100, 1000, 10000, 15000 word features and obtained 57-85% of accuracy. Angiani et al. [5] explained different pre-processing techniques and presented the comparison to find out the best combination of pre-processing modules which is more effective in improving accuracy. They found out highest accuracy with basic cleaning and stemming by experimentation and also the usage of dictionary did not help in performance improvement. Singh and Kaur [6] used WEKA tool for doing sentiment analysis of online movie review data set. Then classification is done by using NB classifier and got 94.968% accuracy for review and 82.69% for twitter dataset.

Ishtiaq [7] proposed rule based scoring unsupervised approach aiming on the ranking of sentiment influencers (noun, adjective etc.). They obtained 85.39% precision on positive tweets, 87.65% on negative, 88% on neutral, 83.52% recall on positive set, 84.18% recall on negative and 94.96% recall on neutral set. Balahar [8] found improvement in accuracy because of pre-processing especially due to the replacement of intensifiers and sentiment words with unique labels. Bhadane et al. [9]. observed an increase in 4.89% accuracy due to StopWords removal and 12.2% due to stemming when tested on 41 reviews. Potdar et al. [10] proposed software named Samiksha (review bot) which generates factual summarization of product reviews. The proposed system has four phases-fetching of user reviews, pre-processing using NLP, analysis of pre-processed reviews and then summarization of review. Jianqiang and Xiaolin [11] observed improvement in accuracy and F1 when negations are replaced and acronyms are expanded but performance hardly changed on the removal of StopWords, URLs and numbers. They found the removal of URLs, StopWords, and numbers as effective for noise reduction in tweets. This is in contrast with Bao [13] result which showed performance improvement on URLs reservation.

Krouska et al. [12] applied various pre-processing methods on 3 different datasets and then for each sentiment analysis is done by using 4 algorithms: NB, SVM, KNN, C4.5 and presented a comparative analysis to determine the impact of pre-processing on accuracy. Zhao [14] performed the similar experiment as [11] except that he did only binary classification task (positive and negative) and evaluated the impact of various pre-processing methods on classifier accuracy only. Rajasree et al. [15] performed twitter sentiment analysis of electronic products by using 4 classifiers- SVM, NB, MaxEnt and ensemble and presented comparative analysis of their performance. Their results show that NB classifier has better precision but lower recall and accuracy than other 3 classifiers. 90% accuracy was obtained for SVM, MaxEnt and ensemble and 89.5 for NB.

Gokulakrishnan et al. [16] explored various pre-processing methods in detail. Results show that basic NB classifier failed in showing sufficient accuracy but acceptable performances were obtained by SMO, SVM, random forest and variants of NB. They also found an increase in accuracy by use of SMOTE technique for handling skewness of data. In future, they would handle sarcasm too. Appel et al. [17] proposed the hybrid approach which is composed of NLP techniques, lexicon and fuzzy set for determining polarity and intensity of polarity. They obtained high accuracy of 88.02 and precision of 84.24 on twitter set which proved the effectiveness of their hybrid approach better than traditional machine learning algorithm like NB, MaxEnt etc. In future, they would be handling sarcasm and also investigate the possibility of using SenticNet. Tripathy et al. [18] analyzed the performance of 4 different classifiers (NB, MaxEnt, SGD, and SVM) for sentiment analysis of IMDB review dataset. They observed that with NB classifier better result was obtained using bigram as a feature but with MaxEnt, SVM and SGD unigram gave better performance. In future, they would try different features also.

III. METHODOLOGY

Data pre-processing is one of the crucial and most important steps in analyzing sentiment from tweets as it helps in the understanding of unstructured data in the better way. Some of the linguistic peculiarities of a tweet such as URLs, usernames, hashtags, elongated words, emoticons, slangs, misspelled words etc. if employed properly might improve classifier performance.

So in our paper, appropriate steps are employed for handling such linguistic peculiarities. Such steps are known as tweet pre-processing for normalizing language and generalizing the employed vocabulary. Before doing pre-processing we have performed tokenization and Part of Speech Tagging with the help of CMU ARK Twitter Part-of-Speech Tagger [19] which tokenize the tweets and then tagged each token with its part of speech also. So for each tweet, we have tweet tokens and pos tokens. For e.g. “Grasshopper holds a survey to show #DeMonetisation was a disaster but from the voting he must be disappointed now :P https://t.co/GXISJ3Pk57N V D N P V # V D N & P D V O V V V R E U”. Now using tweet tokens and pos tokens we have performed pre-processing in 2 phases.

A. Basic Cleaning Operations

First of all, we have done some basic cleaning of tweet text that is denoising because some of the symbols and words have no impact on the orientation of the text.

So all URLs, usernames, hashtag symbol, additional white spaces are removed from tweets. We have also removed StopWords (frequently used words i.e. a ,an ,the, for etc.), html characters(<>,& etc.), prepositions (whose pos token is P), proper nouns (whose pos token is ^ for e.g. modi, bjp), foreign words (whose pos tag is G for e.g. investiga), numeric words (2000, 20k, 20.50, 2,000, $2000 etc.) and punctuations (except exclamation !, question marks ? and asterisk *) from tweets. For removal of StopWords we have created a custom StopWords list with the help of nltk Stop Word dictionary. We have included apostrophe words like it's, he'll into our customized list of StopWords but excluded negation apostrophe words like can't, don't etc. so that negation can be handled later on during phase 2 of pre-processing.

Fig. 1 shown below showcase the basic cleaning steps for Twitter corpus and partially cleaned corpus now act as input for second phase of pre-processing.

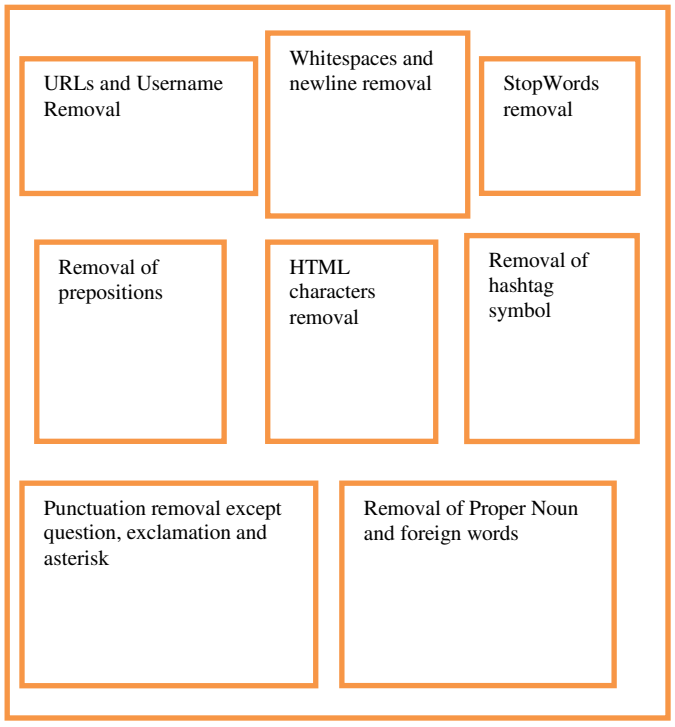

Fig. 1. Pre-processing phase 1: Basic cleaning operations

B. Normalization of tweet

Tweets often contain non-standard words, ill-formed words, negation words and out of vocabulary words. Such words need to be replaced to their canonical forms. Following is done for normalization:

- Acronym look up: Acronyms are the informal language and are replaced by their canonical forms using acronym file having the translation of 343 acronyms. Acronyms are collected from various online resources [20]. For e.g. LOL is replaced by laughing out loud.

TABLE I: Part of Acronym file

Acronym	English Expansion
ilu	i love u
np	no problem
lol	laughing out loud
Ty	thank you

- Apostrophe look up: Apostrophe contractions are also taken into account for normalization while the words with hyphen are kept intact such as pre-processing. So we have compiled a list of apostrophe words except negation apostrophe words (didn't, can't etc.) taken from online resource [21] and add them to the list of StopWords for their removal from tweets.

TABLE II: Part of StopWords file having apostrophe contractions also

StopWords
about
i've
it's
she'll

- Misspell (nonstandard) words replacement: We have also compiled the list of misspelled words taken from online resource [22] and then again by look up mechanism such words are replaced by corrected form. This file has 3432 misspelled words with their corrected form.

TABLE III: Part of Misspelled file

Misspelled word	Normalized Word
actin	Acting
coz	Because
wlcme	Welcome
askin	Asking

- Elongated words normalization: Users often repeat a letter in the word to stress on it for expressing the opinion. Such words then become unavailable in lexicon dictionaries such as WordNet or SentiWordNet. So needs to be normalized. Hence if a letter repeats more than 3 times in a word then replace it by 3 occurrences only. For ex- huuuury is reduced to huuury

- Emoticon replacement: We have created a resource text file with the list of emoticons enumerated on Wikipedia [23] for replacing the emoticons with their emotional polarity by look up in the text file.
- Emoticons are categorized into one of 5 polarities: Extremely-Positive, Positive, Neutral, Negative and Extremely- Negative. So if an emoticon from the text file is found in a tweet it is replaced by its corresponding polarity. For e.g. ":)" is labelled as Positive whereas ":D" is labelled as Extremely-Positive. We have removed the tweets having both positive and negative emoticons. Emoticon replacement will increase weightage of such features during classification.

TABLE IV: Part of Emoticon file

Emoticon	Polarity
:-) :) :o) :]	Positive
:D C: :)) :)))	Extremely-Positive
:| >;) >:-)	Neutral
:-(:(:c :<	Negative
:-((:-(((	Extremely-Negative

- Replace negation words: Negation words can change the sentiment of tweet completely. For e.g. "This movie is not good" Here good indicates positive sentiment but the presence of word "not" before word "good" inverse the sentiment of above sentence. Negation plays an important role as it can reverse polarity or change the strength of polarity. For e.g. "Terrible" is a negative sentiment word but when negated it become mildly negative instead of positive. Negation scope is important to define that is which all words are affecting by negation like "not a very good idea" depicts distant negation while "not good" depicts immediate negation. It is necessary to handle negation otherwise we may get incorrect classification that would affect accuracy. So we firstly we have created a list of negation words such as not, never, n't, can't, don't, barely etc. and if a negation word is identified in tweet then it is replaced with tag "negation" so that in future for doing sentiment analysis we can handle negation words easily. For e.g. "can't" is replaced by " negation" and "This movie is not good" becomes "This movie is negation good".

TABLE V: Part of Negation words file

Negation Words
not
don't
can't
rarely

At the end of the second phase of pre-processing, we get cleaned and normalized twitter corpus which is now ready to undergo sentiment classification process. Above mentioned steps of pre-processing would improve the performance of classifiers for sentiment analysis due to dimensionality reduction. Fig. 2 below portrays complete normalization process of our proposed pre-processing methodology.

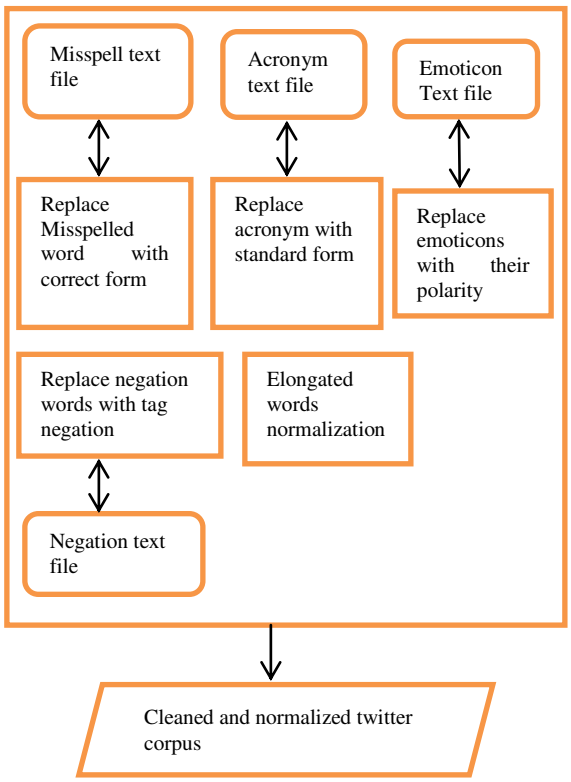

Fig. 2. Pre-processing phase 2: Normalization

IV. EVALUATION

A. Corpus

We have collected 19000 tweets on demonetization and 6000 tweets on iphone7 by use of twitter search API and annotate them automatically into positive, negative and neutral classes by count of positive, negative words and presence of emoticons.

B. Experiments

In order to evaluate effectiveness and accuracy of our proposed algorithm experiment have been conducted on first 1000 tweets of demonetization dataset. For experimentation we have done automatic pre-processing of first 1000 tweets by the use proposed algorithm and then manually pre-processing is also done for the same 1000 tweets to validate the accuracy of the proposed algorithm. Some of the examples for

exemplifying our evaluation approach of proposed pre-processing method are described below:

1) Raw tweet: Post #Demonetisation frst tym used @SBI app. Amazng improvement! Nw chnge the rules where v need to visit SBI branch to change our passwrds!

a) CMU ARK tagger output (tweet tokens and POS tokens): Post #Demonetisation frst tym used @SBI app . Amazng improvement ! Nw chnge the rules where v need to visit SBI branch to change our passwrds !N # A N V @ N , A N , G V D N R P V P V ^ N P V D N ,

b) Automatic pre-processing output (cleaned tweet): Post Demonetisation first time used application amazing improvement ! change rules need visit branch change passwords !

C) Manual pre-processing output: Post Demonetisation first time used application amazing improvement ! change rules need visit branch change passwords !

d) Result: correctly pre-processed by proposed method

2) Raw tweet: VOTE NOW: Are you still supporting PM @narendramodi on #DeMonetisation? #___ Takr the Poll. https://t.co/Q9ZI6BFK1S

a) CMU ARK tagger output (tweet tokens and POS tokens): *VOTE NOW:* Are you still supporting PM @narendramodi on #DeMonetisation ? #___ Takr the Poll. *https://t.co/Q9ZI6BFK1S* V R , V O R V N @ P ^ , # V D N , U

b) Automatic pre-processing output (cleaned tweet): VOTE still supporting PM DeMonetisation ? Takr Poll

c) Manual pre-processing output: VOTE still supporting PM DeMonetisation ? Take Poll

d) Result: incorrectly pre-processed by proposed method

Finally, accuracy of proposed algorithm is calculated as below

$$Accuracy = \frac{\text{correctly pre} - \text{processed tweets by algorithm}}{\text{total tweets}} \times 100$$

We have achieved 87.6% accuracy which is considerably good. It means out of 1000 tweets 876 are correctly pre-processed by the proposed algorithm.

Again, we test our proposed method on iphone7 tweets in order to evaluate effectiveness of our proposed method on another domain. For this, we have randomly selected 200 tweets of iphone7 dataset and then calculate accuracy of the proposed method with the same procedure described above.

We obtained 90.5% accuracy on 200 tweets of iphone7 and 88.08% combined accuracy of proposed method on 1200 tweets (1000 from demonetization dataset and 200 from iphone7 dataset).

TABLE VI: Evaluation results

Dataset	No. of tweets for evaluation	Correctly pre-processed by proposed algorithm	Accuracy obtained
Demonetization	1000	876	87.6
Iphone7	200	181	90.5
Demonetization+Iphone7	1200 (1000+200)	1057	88.08

V. CONCLUSION AND FUTURE WORK

Usage of slang and dubious grammar leads to increase in pre-processing requirement exponentially so in this research we have discussed the pre-processing framework for standardizing tokens in a tweet which in turn leads to the reduction in computational complexity. Comprehensive application of proposed pre-processing algorithm makes us attain satisfactory accuracy.

In future, we would continue our evaluation by using more number of tweets, different acronym list, stoplist and splitting of attached words. We could also analyze the impact of pre-processing methods on classifier performance.

REFERENCES

[1] Haddi, Emma, Xiaohui Liu, and Yong Shi. "The role of text pre-processing in sentiment analysis." Procedia Computer Science 17 (2013): 26-32.

[2] Singh, Tajinder, and Madhu Kumari. "Role of Text Pre-processing in Twitter Sentiment Analysis." Procedia Computer Science 89 (2016): 549-554.

[3] Kotsiantis, S. B., D. Kanellopoulos, and P. E. Pintelas. "Data preprocessing for supervised leaning." International Journal of Computer Science 1.2 (2006): 111-117.

[4] Elgamal, Mahmoud. "Sentiment Analysis Methodology of Twitter Data with an application on Hajj season."

[5] Angiani, Giulio, et al. "A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter." KDWeb. 2016.

[6] Singh, Rajni, and Rajdeep Kaur. "Sentiment Analysis on Social Media and Online Review." International Journal of Computer Applications 121.20 (2015).

[7] Ishtiaq, Munazza. "Sentiment analysis of twitter data using sentiment influencers." Journal of Intelligent Computing 6.1 (2015): 17.

[8] Balahur, Alexandra. "Sentiment Analysis in Social Media Texts." WASSA@ NAACL-HLT. 2013.

[9] Bhadane, Chetashri, Hardi Dalal, and Heenal Doshi. "Sentiment analysis: measuring opinions." Procedia Computer Science 45 (2015): 808-814.

[10] Potdar, Aarti, et al. "SAMIKSHA-Sentiment Based Product Review Analysis System." Procedia Computer Science 78 (2016): 513-520.

[11] Jianqiang, Zhao, and Gui Xiaolin. "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis." IEEE Access 5 (2017): 2870-2879.

[12] Krouska, Akrivi, Christos Troussas, and Maria Virvou. "The effect of preprocessing techniques on Twitter sentiment analysis." Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on. IEEE, 2016.

[13] Bao, Yanwei, et al. "The role of pre-processing in twitter sentiment analysis." International Conference on Intelligent Computing. Springer, Cham, 2014.

[14] Jianqiang, Zhao. "Pre-processing boosting Twitter sentiment analysis?." Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on. IEEE, 2015.

[15] Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013.

[16] Gokulakrishnan, Balakrishnan, et al. "Opinion mining and sentiment analysis on a twitter data stream." Advances in ICT for emerging regions (ICTer), 2012 International Conference on. IEEE, 2012.

[17] Appel, Orestes, et al. "A hybrid approach to the sentiment analysis problem at the sentence level." Knowledge-Based Systems 108 (2016): 110-124.

[18] Tripathy, Abinash, Ankit Agrawal, and Santanu Kumar Rath. "Classification of sentiment reviews using n-gram machine learning approach." Expert Systems with Applications 57 (2016): 117-126.

[19] http://www.ark.cs.cmu.edu/TweetNLP/

[20] http://www.illumasolutions.com/omg-plz-lol-idk-idc-btw-brbjk.htm#top(1)

[21] http://www.grammarmonster.com/lessons/apostrophes_replace_letters.htm (3)

[22] http://www.hlt.utdallas.edu/~yangl/data/Text_Norm_Data_Release_Fei_Liu/Test_Set_3802_Pairs.txt

[23] https://en.wikipedia.org/wiki/List_of_emoticons(2)