International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

An Effective Knowledge-Based Pre-processing System with Emojis and Emoticons Handling on Twitter and Google+

Tripti Agrawal, Archana Singhal

Abstract: ***Social networks, nowadays, are an imperative source for users to express their opinions freely on diverse topics including various brands and services. Presence of millions of users results in the generation of gigantic volume of data. The generated data are actually a treasure trove for business organizations to understand public sentiments towards their brand and services. This data is dynamic and highly unstructured due to varying writing patterns such as use of slangs, misspelled words, emojis and so on. The unstructured data makes pre-processing an underlying and challenging step in sentiment analysis. Therefore, authors have thoroughly explored a series of pre-processing steps on two social networks and observed that the sequence order of pre-processing steps plays an important role in improving overall pre-processing results. Hence, an improved ordered sequence of pre-processing steps has been proposed. It has also been observed that the presence of emojis in the text act as a pivot in determining users' sentiments. Therefore, a detailed handling of emojis has also been included in the proposed pre-processing steps. New dictionaries have been compiled to provide a language to the emotional contents carried by emojis and emoticons. Few existing dictionaries have also been extended to make them more comprehensive for lookup task. Additional pre-procesing steps for handling multiword usernames and hashtags have also been incorporated in the proposed work. Further, experiments have been carried out to compare the proposed system with the existing ones. Results show that the proposed system outsmart the existing approaches mainly due to implementation of pre-processing steps in an ordered sequence and handling of emojis.***

Index Terms: Twitter, Google+, Sentiment Analysis, Emojis, Emoticons, Pre-processing

I. INTRODUCTION

Social networking sites provide a convenient platform for millions of users to express their views and experiences freely and independently about various topics, brand products, and services. The data generated on social networks are a great source for different organizations to get a better insight into users' sentiments. However, the presence of slangs, misspelled words, hash tags, numbers, emoticons along with uninformative data such as URLs, HTML tags, usernames, etc makes this data highly unstructured and due to this, the task of effective sentiment classification becomes a challenging research area. Thus, before the sentiment classification task, an efficient data pre-processing is needed to filter out irrelevant features without the loss of users' context. The appropriate selection of features is crucial as it will have a huge impact on the accuracy of the classifier. In this paper, data pre-processing has been implemented exhaustively on two different social networks - Twitter and Google+. Twitter with over 326 million (Statista 2018) monthly active users is a microblogging site that has set a maximum limit of 280 characters for tweets whereas, Google+ with 395 million monthly active users has a maximum limit of around 100,000 characters for Google+ posts. This limit of Google+ is even greater than that of Facebook's maximum character limit which has been set to 63,206 characters. The aim for selection of these two different social networks is to analyze the impact of a microblogging social network (such as Twitter) along with a detailed discussion providing platform (such as Google+) on sentiment analysis of same domain data but from two different sources. It has been observed that the character limit restriction affects writing style of users on different social networks. For example, on Twitter, users use acronyms, slangs, URLs, emojis, emoticons, etc. quite frequently, in order to convey more in fewer words in comparison to Google+ users. Apart from the maximum character limit, another point of difference between these two social networks lies with different username rules. On every social network, all registered users have unique usernames. Anything directed towards an intended user is indicated by mentioning his or her username in the text. On Twitter, every username starts with '@' symbol whereas Google+ username starts with '+' symbol. On one end, Twitter does not allow any whitespace in between a username, hence, every username on Twitter is a single word. On the other hand, Google+ allows whitespace in between a username, therefore, every Google+ username can either be a single word or a multiword. Replacing single-word usernames is pretty straight forward with the help of regular expressions whereas replacing multiword usernames in the text is a challenging task. In the present paper, a method for replacing multiword username has been suggested and implemented on Google+ data. Another difference between these two social networks lies in the selection of language while collecting data from APIs. From Twitter API, language can be set as 'English' in the search query but no such language setting option is provided by the Google+ API. This results in the collection of code mixed data and such data has been handled by the proposed system. Beside these differences between the two networks,

Manuscript published on 30 September 2019.
***Correspondence Author(s)**
Tripti Agrawal, Department of Computer Science, University of Delhi, Delhi-110007, India.
Archana Singhal, IP College for Women, Department of Computer Science, University of Delhi, Delhi-110054, India.

it has been observed that the latest trending linguistic feature frequently used by netizens on social networks is the abundant usage of emojis and emoticons. Users make use of emojis and emoticons to express their feelings and to add a more personal touch to their content. These emojis and emoticons also represent a digital way of humanizing communication by facilitating visualization of the writer's feelings in the reader's mind. However, emojis and emoticons are two different things and thus, a distinction needs to be made between these two. An emoticon, such as ;-), is a representation of facial expression, formed by various combinations of keyboard characters such as punctuation, angle brackets, digits, arithmetic operators, alphabets, parenthesis, and special symbols. Whereas, emojis are the pictures or pictograms that have a resemblance to an actual physical object and thus, facilitates more expressive messages than emoticons. Another point of difference between these two is that emoticons have the ability to represent only the nuances of facial expressions whereas emojis can represent a wide range of concepts, ideas, events, activities, and locations as well. Hence, emojis are the advanced new generation of emoticons that have grown by leaps and bounds over the years since their existence from 1982 onwards. In the present work, data has been collected from Twitter and Google+ using their respective APIs. In the collected data, both emojis and emoticons are present. In Twitter data, emojis are present as Unicodes (example, <u+0001f601>) whereas, in Google+, they are present as mojibakes (such as ðÿ˜). For this reason, separate emoji handling dictionaries have been compiled. In the collected tweets, a total of 4,520 Unicodes are present. Out of these, there exist 500 unique Unicodes that are part of the Unicode emoji standard and 50 other unique Unicode characters that are not part of the Unicode emoji standard. A separate dictionary has been compiled for these 50 unique Unicodes. Among these 50 Unicodes, few Unicodes comprise of currency codes, company logos, direction arrows and many more. Since, the presence of emojis and emoticons in the text, are a rich source for understanding users' sentiments, rigorous efforts have been made to compile comprehensive dictionaries for emojis and emoticons handling. The aim of these dictionaries is not merely the replacement of emoji or emoticons with their English names, rather excessive efforts have been made to provide sentimental language to these non-informative codes for emojis and emoticons. In a nutshell, in this paper, pre-processing steps have been implemented rigorously on two different social networks' data. Various existing dictionaries have been modified to make them more comprehensive for a successful lookup task. Also, new dictionaries for handling of emojis and emoticons have been compiled in an attempt to provide language to the sentiments expressed by unspoken emojis or emoticons, present in the text. Also, a procedure for handling multiword usernames in Google+ has been proposed and implemented on Google+ data. After the proposed pre-processing sequence, experiment results have been presented to validate the significance of the proposed work.Rest of the paper is organized as follows - Section 2 outlines the existing work on sentiment analysis and existing pre-processing approaches. Section 3 provides an overview of the proposed system. Section 4 shows the experimental results from the comparison of existing pre-processing approaches with the proposed system. Finally, section 5 concludes the paper.

II. RELATED WORK

Sentiment analysis [1] is an evolving field of Natural Language processing mainly due to the increasing demands by industries to automate data processing and sentiment classification task. Authors in [20] explored the potential of various social networks for target brand marketing. The huge volume of data expressing opinions and the informal language used by netizens on social networks, make sentiment analysis a challenging research area for many researchers. In the past, many researchers carried their research on Twitter data only. In fact, Go, Bhayani and Huang were the first who conducted their research in sentiment analysis on Twitter data and thereafter, their work served as a baseline for many researchers. In sentiment analysis, data pre-processing is a crucial step due to the high presence of informal language in tweets. Hence, researchers in this field implemented various pre-processing steps on Twitter data in order to achieve higher accuracy in sentiment classification results. In the initial years of sentiment classification of tweets, basic pre-processing steps such as removal of retweets, numbers, replacement or removal of usernames and URLs, stop words removal, etc. have been implemented by authors in [2]–[5]. Later on, many researchers realized the importance of emoticons in altering the sentiment polarity of tweets. Authors in [6]–[9], included emoticon handling as an additional pre-processing step in their work. In [7]–[9], authors have grouped various emoticons into 5 different sentiment polarities - "Positive", "Extremely-Positive", "Neutral", "Negative" and "Extremely-Negative" and replaced emoticons with their corresponding polarities whereas in [6], authors have categorized emoticons into 6 different categories - "EMOT_CRY" , "EMOT_SMILEY", "EMOT_LAUGH", "EMOT_LOVE", "EMOT_WINK", and "EMOT_FROWN" and replaced emoticons with their corresponding categories. However, these categories are not sufficient to retain the nuances of different emoticons. For this reason, in the present work, each emoticon has been given a proper meaning to express its emotional nuances clearly and precisely. In [10]–[14], authors conducted a series of experiments to verify the effectiveness of various pre-processing methods on Twitter data. From the experimental results, authors in [11] concluded that there is no improvement in accuracy after the removal of Stop words, URLs and numbers whereas, experiments results showed improvement on URLs reservation in [10]. In the paper [15], in addition to the basic text pre-processing steps, authors have expanded negations and acronyms using online Slang dictionary. Whereas, in the proposed system, authors have modified existing slang dictionary by adding more acronyms and frequent misspelled words observed in the collected data. Also, a dictionary for non-negated Apostrophe look-up, in addition to the negation contractions dictionary has also been compiled. The proposed system differs from pre-processing approach in [9] in many ways.

Retrieval Number: K13520981119/19©BEIESP
DOI: 10.35940/ijitee.K1352.0981119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The dictionary for expanding acronyms is more comprehensive. Another difference between the two is in the handling of emoticons. In [9], emoticons are replaced with their sentiment polarities whereas, in the proposed work, emoticons have been replaced with their emotional content. Also, combining splitted acronyms, dictionaries for emoji handling and expanding HTML codes are a few additional steps of the proposed system. Above all, the proposed system has been implemented and experimented on two different social networks data whereas existing approaches have been implemented and experimented mainly on Twitter data.

The proposed work is an extension of authors work in [16]. In the proposed system, hashtags are retained in the data as well as the pre-processing sequence has also been modified. The proposed work also extends the authors' previous work in [16] by the inclusion of additional pre-processing steps such as handling of code-mixed data, detailed handling of emojis along with emoticons and handling multiword usernames. In the present scenario, pre-processing of social networks' data is incomplete without handling of emojis due to their frequent usage by netizens. Many researchers have contributed their work towards handling of emojis. Authors in [17] provided the first emoji sentiment lexicon for 751 frequently used emojis where each emoji is assigned a discrete 3 valued variable sentiment label in the range of -1,0,+1, using discrete probability distribution and the sentiment score is then computed as the mean of the probability distribution. However, a single emoji can have multiple interpretations and depending on the context, sentiment polarity of emojis changes. Authors in [12], have introduced the first machine-readable sense inventory for emojis called EmojiNet. It links Unicode of emojis with their English language meanings extracted from four different web resources and integrates these resources with BabelNet to infer sense definitions. However, the description available on these resources are quite lengthy and the linking process must be time-consuming to the best of our knowledge. Also, EmojiNet fails to link emojis in Google+ data as they are available as mojibake instead of Unicode. Moreover, many emojis are often misconstrued by users and the resources used by EmojiNet fails to incorporate such misinterpretations. In the proposed work, each emoji is replaced with their emotional content and multiple possible interpretations have also been incorporated in the newly compiled dictionaries of emojis. Rigorous efforts have been made to get all possible interpretations of an emoji in a minimal set of words. Also, the proposed system is capable of handling emoji mojibakes present in Google+ data. Next section explains the detailed data pre-processing steps of the proposed work.

III. PROPOSED WORK

A. Data Gathering

In the present work, data has been collected from two different social networks using their respective APIs. Authors have collected around 80,853 unique tweets of eight different electronic brands namely Motorola, Apple, Samsung, Oppo, HTC, Xiaomi, Nokia and Huawei. Similarly, around 1,09,563 users comments from Google+ have been retrieved for the same electronic brands.

Data collected from APIs are raw texts that need extensive pre-processing before sentiment classification task. Details of pre-processing steps implemented on the raw text are explained in the next subsection.

B. Data Pre-processing

To prepare data for sentiment classification, the following are the pre-processing steps that have been implemented in a sequence on the raw text as the sequence order pre-processing affects the overall results of pre-processing [16]. The pre-processing steps of the proposed system are explained below in the same sequence order as suggested by the authors.

1) Handling of Hashtags: This is the first pre-processing step of the proposed sequence. A hashtag is a keyword or an unspaced phrase that is always preceded by a hash mark '#'. Use of hashtags, allows people to easily follow topics they are interested in. These hashtags are mainly used to index subjects (for example, #MotoG) or trending topics (for example,#TrumpUKVisit) or an event (example, #TripToDubai).

This has been observed that hashtags are also used by users to express their experience or feelings (for example, #AwesomeWeather, #fun, #love, etc.). In this pre-processing step, hashtags have been retained in the data after stripping off hash mark '#'. Also, hashtags with unspaced compound words (example, #NewYear) have been split into independent words on the basis of their letter cases via regular expression. For example, after hashtag handling, the hashtag, #AwesomeWeather, gets replaced by "Awesome Weather" in the appearing text.

Due to retaining letter case requirement, hashtags handling is the first pre-processing step and converting text to lowercase is the second step in the sequence.

2) Converting to lowercase: Usually, users on social networks use different letter cases for referring the same thing. For example, users may write lol or LOL or Lol or LoL or lOl, depending on their mindset at that particular point of time. Thus, in this step, the raw text is first converted to lowercase so that dictionary lookup doesn't fail just because of the difference in letter cases.

After this step, translating code mixed data into English follows next in the sequence.

3) Translating code mixed data into English: This step is applicable to Google+ data only. From the Google+ data, it has been observed that netizens, who have knowledge of two or more languages, prefer to mix English text with another known language while posting their comment(s) on this network.

4) This results in the generation of code mixed data which is an individual research area in itself. In the present work, a python library named 'TextBlob[1]' has been used to translate non-English text to English.

Table I shows examples of non-English texts, followed by their original text language and, finally, their corresponding translated English text via TextBlob.

After this pre-processing step, the next step is - Expanding HTML codes.

[1] https://textblob.readthedocs.io/en/dev/quickstart.html#translation-andlanguage-detection

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table I: Examples of Non-English Text Translated to English via TextBlob

Original Text	Language	Translated Text in English
mein handy ist in 70 minuten voll.	German	my phone is full in 70 minutes.
see mon prã©fã©rã©	French	see my favorite
isse mobile kharab to nahi hoga	Hindi	this will not spoil the mobile
me encanta	Portuguese	i love it
bien putos caros	Spanish	good fucking expensive

5) Replacing HTML codes: In this pre-processing step, HTML codes (such as "&","'", etc.) are replaced with their meaning with the help of a dictionary compiled by authors. This pre-processing step is needed for two purposes. First, for expanding acronyms, for example, "q&a" becomes "q&a" after HTML code replacement and this "q&a" will be expanded as "question and answer" by the acronym dictionary. Second purpose is for expanding contractions. For example, "can't" becomes can't after replacing HTML code which in turn would be expanded as "cannot" by the negation contractions dictionary.

After expanding HTML codes, the next step is Combining Acronyms and letters.

6) Combining Acronyms and letters: The fourth step of the proposed system is combining acronyms. It has been observed that sometimes users use spaces in between acronyms such as "i l u", "w o w". Due to spaces, splitted acronyms either would not be expanded to their standard form or are wrongly expanded. To handle such cases, splitted acronyms are combined by removing in-between spaces so that they can be correctly expanded to their original standard form. It has also been found from the collected data that there exist words, with a space in between each letter of the word, example, "c u r s e d", "k i s s i n g", etc. Such letters individually, do not carry any meaning in their own, but after combining letters, the formed word makes sense. For such cases, splitted letters have been combined by removing in-between spaces. Without this step, such individual letters would be treated as acronyms and could be wrongly expanded by the next step.

7) Expanding acronyms and slangs: Expanding acronyms and slangs is an important pre-processing step as their expansion provide detailed information of the user's context and even emotions such as lol (laugh out loud). For this purpose, modified Internet Slang Dictionary[2] has been used for expanding acronyms and slang present in the text to their original standard form. Although, this dictionary is quite comprehensive in itself, still, netizens style of writing acronyms sometimes led to unsuccessful matches. For example, the dictionary has lookup available for "jj" only, but if a user uses j/j or j.j. in their text then it would not be expanded at all because such forms of acronyms are not present in the dictionary. Therefore, the dictionary has been modified for each acronym to handle such type of cases as well. Many additional acronyms and slangs found in the collected data, have also been added to the dictionary along with their various possible forms so as to make it more comprehensive for lookup. For example, the phrase, "toodalooo" with its possible forms - "toodaloo" or "too-da-loo" or toodle-oo" or "toodleoo" or "too-a-loo" has been added to the dictionary. Also, few acronym meanings have also been modified as per the domain data terminology, for example, "gb" originally in the dictionary stands for "going back" but in the computer terminology "gb" stands for "gigabytes". Similarly, "pos" originally stands for "parent over shoulder" and has been modified as "point of sale" in the modified dictionary.

In the modified dictionary, many misspelled words have also been incorporated. Examples of misspelled words and acronyms/slang that are now part of the modified dictionary are shown in Table II and Table III respectively. These tables show unique sample entries added in the dictionary.

Table II: Examples Of Additional Misspelled Words Added To The Slang Dictionary

Misspelled Words	Corrected Spelling
Sammy	Samsung
Pattetic	Pathetic
Suffisant	sufficient
Troubleshat	troubleshoot
Essier	easier
Continuednued	continued
Galaxy	galaxy
Pixle	pixel
Coppyd	copied
Xcited	excited

Table III: Examples Of Additional Slangs/Acronyms Added To The Slang Dictionary

Slang/Acroym	Meaning
Spec	Specification
Comfy	Comfortable
too-da-loo	Goodbye
Eeek	dislike or slight fear
Icarp	regretting of wasting money in buying apple device
Uidesign	user interface design
Ios	iphone operating system
Navbar	navigation bar
Pos	point of sale
Uhhh	feeling annoyed

After expanding acronyms and slangs, "Negation and non-negation apostrophe lookup" follow the next.

7) Negation and non-negation apostrophe lookup: In this step, every negation contraction is expanded to its original form with the help of a dictionary compiled by authors. For example, "aren't", "can't've", "havn't" have been expanded as "are not", "cannot have" and "have not" respectively. The same dictionary also facilitates non-negation apostrophe lookup. For example, "that'll" will be expanded to "that will".

After this pre-processing step, the next step is - Emojis and Emoticon handling.

[2] https://www.internetslang.com/

8) Emojis and Emoticons handling: Emojis and emoticons handling is the core step of the proposed pre-processing sequence. Emojis and emoticons are a form of self-expression that allows users to express themselves in a way they wish people to perceive their emotions. In fact, these emojis and emoticons are a rich source for understanding users' sentiments. As per the latest trend on social networks, the usage of new generation emojis has surpassed usage of old generation emoticons to a great extent due to emoticons limited ability to express nuances of emotions.

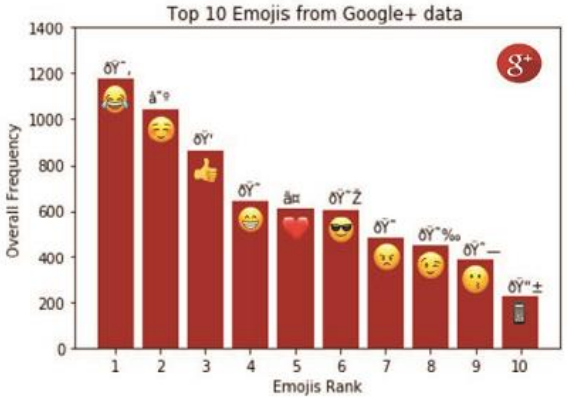
Table IV shows the usage distribution of emojis and emoticons in the collected Twitter and Google+ data.

Table IV: Usage Distribution of Emojis and Emoticons in Twitter And Google+ Data

Social Networks	Total Collected Data	Numberof Emojis	Numberof Emoticons
Twitter	80,853	23,133	97
Google+	1,09,563	9,967	218

From Table IV statistics, it is evident that emojis are used more frequently in comparison to emoticons. Hence, in the present work, emojis have been handled in detail. Emojis in the Twitter data are present as Unicodes whereas, in Google+ data, they are in the form of mojibakes . Hence, two different dictionaries have been compiled to handle emojis of these two social networks. It has been observed that Twitterati are more creative and frequent in the usage of emojis in comparison to Google+ users. As per the collected data statistics, there exist around 500 unique emoji Unicodes along with 1397 unique Unicode combinations for Twitter data. Whereas for Google+ data, only 177 unique emoji mojibake exists along with 928 unique emoji mojibake combinations. Fig. 1 shows the top 10 emojis in the collected data from Twitter and Google+.

While compiling dictionaries, the real intention had been to provide a language to the feelings expressed by emojis and



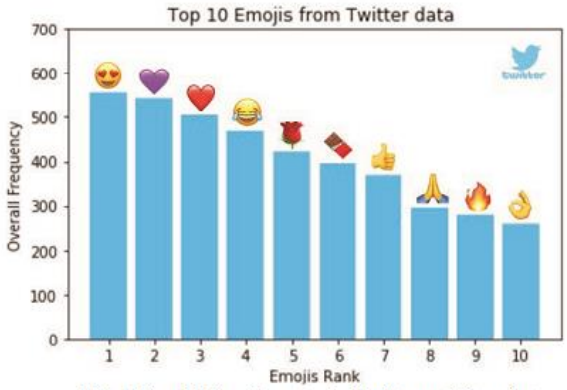


Fig 1: Top 10 Emojis usage in Twitter and Google+

Table V: Emoji Polarity Comparison Of Standard Cldr Names With Emoji Meanings In The Proposed System

Emoji	CLDR Name	Emoji Polarity	Meaning in Proposed System	Emoji Polarity
💪	Flexed Biceps	Neutral(0.0)	strong and powerful	Positive(.73)
❤️	Red Heart	Neutral(0.0)	love heart	Positive(.64)
🙏	Folded Hands	Neutral(0.0)	expressing request, respect or thanks	Positive(.72)
💞	Revolving Hearts	Neutral(0.0)	growing love feelings	Positive(.71)
💕	Two Hearts	Neutral(0.0)	love in the air	Positive(.64)
🤬	Face with symbols on mouth	Neutral(0.0)	angry face with symbols	Negative(-.51)
🤪	Zany Face	Neutral(0.0)	amusing zany face	Positive(.38)
😛	Face with tongue	Neutral(0.0)	fun, excitement, silliness, cuteness, happiness or joking	Positive(.91)
💙	Blue Heart	Neutral(0.0)	trust,loyalty,attraction, love	Positive(.64)
🏴‍☠️	Pirate Flag	Neutral(0.0)	danger warning, symbol of death	Negative(-.86)
🤮	Face Vomiting	Neutral(0.0)	intense disgust	Negative(-.56)
🙌	Raising Hands	Neutral(0.0)	joyful celebration with raising hands	Positive(.60)
💝	Heart with Ribbon	Neutral(0.0)	gifting love heart romantically	Positive(.78)
🙄	Face with rolling eyes	Neutral(0.0)	feeling frustration or boredom	Negative(-.60)
👍	Thumbs up	Neutral(0.0)	agreed, approval, encouragement, all the best	Positive(.79)

emoticons in the text. This has been observed that emoji 'CLDR Short Name' provided by Unicode consortium does not always exhibit the real sentiments of users and thus, require their meanings to be decoded in order to extract sentiments.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

For example, the CLDR name of ❤️ emoji is the red heart and this name does not exhibit the actual construal of the emoji and that is 'love'. For this reason, these emojis and emoticons have not been simply replaced with their CLDR names in the compiled dictionaries rather, with their decoded emoji meanings derived from various online sources such as emojipedia.com, emojis.wiki, urbandictionary.com, emojimeanings.net, and many more. The challenging task during the compilation of these dictionaries was to get meanings in the least possible words which otherwise, would lead to an increase in vocabulary size during Sentiment Analysis.

Table V shows examples for comparison of emoji polarity on the basis of their CLDR names versus on the basis of emoji meanings in the compiled dictionaries by authors. The emoji polarity of emojis has been computed with the help of lexicon and rule-based sentiment analysis tool - VADER [19] that is specifically attuned to sentiments expressed in social media and has been found to be quite successful when dealing with social media texts. In table V, emoji sentiment polarity is accompanied by a polarity score which is a compound score given by VADER. This normalized compound score ranges from -1 to +1.

According to [19], the typical threshold values for detecting sentiment polarity are:

positive sentiment: compound score >= 0.05

neutral sentiment: (compound score > -0.05) and (compound score < 0.05)

negative sentiment: compound score <= -0.05.

This scoring consideration is compatible with the Emoji data dictionaries compiled by the authors as with this scoring, expected results match the actual results.

Also, there are few emojis that are frequently used by netizens but in the wrong context i.e., contrary to their real meaning. Table VI shows examples of such misused emojis with their actual meanings and misinterpreted meanings by the users. In the proposed work, the misused interpretations of emojis have also been incorporated in the compiled dictionaries to retain users sentiments. During the dictionary lookup, the best interpretation based on the context of the text is selected. In case, if only emojis are present, then emojis are replaced with their first place meaning in the dictionary and that is their standard meaning.

Table VI: Examples Of Commonly Misused Emojis

Emoji	Actual meaning	Misinterpretation meaning
😤	Face with look of triumph	To convey someone is angry or even fuming
🙏	Folded hands to express respect or thanks	High-five
😗	Kissing face	Whistling face
😫	Tired face	Showing whining or Frustration
💩	Pile of poo	Scoop of chocolate ice-cream

During emojis and emoticons handling, it has been noticed that the presence of emojis in the text affects the sentiment polarity of text. To illustrate this observation, Table VII shows text examples and their text polarity with and without emoji consideration. It is evident from Table VII that the presence of emoji(s) in the text are the primary essence of users' sentiments and have the ability to change entire text sentiment polarity.

In sentiment analysis, facial emoticons (such as :), :(, :D and so on) are considered as a primary source of sentiments. For this reason, many researchers have focused their work primarily on emoticon handling only. However, in the present work, it has been noticed that non-face emojis (such as, 🌴, 📱, etc.) are also important to be handled during sentiment analysis. As per the authors' observation, it has been found that the meaning derived from the combination of two or more non-face emojis can be categorized into two categories - first, providing information only and second, providing information and sentiment both. Both categories are needed to be handled as information is necessary for retaining users' context and the sentiment is necessary for determining sentiment polarity. Therefore, handling non-face emojis are also an integral part of the compiled dictionaries.

Table VII: Text Polarity With And Without Emoji(S)

Pre-processed Text	Emoji Meaning	Polarity with Emoji	Polarity without Emoji
note5 to note7 gear to gear 3 upgradelife 😊	Smiling with Blush	Positive (.46)	Neutral (0.0)
samsung, i see what you did there 😏	Unpleasant Smirk	Negative (.48)	Neutral (0.0)
Camera sam-sung Galaxy s10 😀👍	smilingface, agreed, approval, encouragement, all the best	Positive (.91)	Neutral (0.0)
i got my new iphone 👍❤️	smiling face, love heart	Positive (.80)	Neutral (0.0)
samsung galaxy s10+ will have 6gb ram 😃	happy face with big eyes	Positive (.57)	Neutral (0.0)

Table VIII shows examples of non-face emojis with their above-stated categories and the sentiment polarity with and without including emojis.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table VIII: Examples Of Non-Faced Emojis and Their Effect On Sentiment Polarity

Text	Meaning with Emoji	Sentiment Polarity with Emoji	Category	Sentiment Polarity without Emoji
We have a visitor in the harbor 🐳	we have a visitor in the harbor whale	Neutral(0.0)	Information only	Neutral(0.0)
Samsung 📱🔋💪	samsung mobile phone battery strong and powerful	Positive(.73)	Information+ Sentiment	Neutral(0.0)
📱🔋📱 sucks	mobile phone battery charger sucks	Negative(-.36)	Information only	Negative(.36)
🤳📱👰❤️⛰️	selfie mobile phone family love heart mountain	Positive(.64)	Information+ Sentiment	Neutral(0.0)

During emoji Unicode handling, it has been observed that there exist Unicode characters in the Twitter data that have no emoji version and are intended to be displayed only as black and white graphics symbols on most platforms. In the collected tweets, there are such 50 Unicode characters. To handle these Unicode characters, a separate dictionary has been compiled by the authors. Table IX shows a few examples of such Unicodes.

Table IX: Examples of Non-Emoji Unicodes

Unicode	Graphic Symbol	Short Description
U+F8FF		Apple Company logo
U+20B9	₹	Indian Rupee Symbol
U+26FE	⛾	Cup on Black Square
U+2713	✓	Checkmark
U+265B	♛	Black Chess Queen
U+266A	♪	Musical Note
U+2606	☆	Star
U+2698	⚘	Flower

In the present work, along with emojis, emoticons have also been handled in detail. For this, a comprehensive dictionary has been compiled where each emoticon corresponds to its emotional content. A sample from compiled emoticons dictionary is shown in Table X.

Table X: Emoticons With Their Meanings

Emoticons	Meaning
:j	Smirk
<3	love heart
:s	Confused
:$	Embarrassed
:>	mischievous smile
:')	tears of happiness

The next step after Emojis and Emoticons handling is the removal of URLs from the text.

9) Removal of URLs: Social network users use URLs and shortened URLs to link detailed information about the topic in discussion. From the observations, it can be stated that URLs standalone does not provide any meaningful information that can affect the sentiment polarity. Hence, URLs have been removed from the data during pre-processing.

After removal of URLs, the next step in sequence is the removal of HTML tags.

10) Removal of HTML tags: The purpose of HTML tags is to format the display of content on the web. The observations show that these HTML tags by far have no impact on sentiment polarity and therefore, they have been removed via regular expression during pre-processing.

After removal of HTML tags, the next pre-processing step is handling usernames of social networks.

11) Handling usernames: On every social network, every user has a unique name. Anything directed towards a user is indicated by mentioning the username in the text. On Twitter, every username is preceded by '@' and no whitespaces are allowed in-between. Thus, with the help of regular expression, replacing username is pretty straightforward. On the other hand, every Google+ username is preceded by '+' and whitespaces are also allowed. Thus, Google+ allows single word as well as multiword usernames such as "+shahroz ahmed", +"mohammed al saied" and many more. Presence of multiword usernames in Google+ data makes replacing usernames a challenging task. Hence, an algorithm has been proposed and implemented for this task. In the proposed algorithm, replacing multiword usernames has been divided into three subtasks -"PERSON" Entity Identification, followed by extracting usernames and finally username replacement in the text.

For Entity Recognition, initially, both Stanford Named Entity Recognition (NER) and Spacy NER model failed to recognize Google+ usernames as "PERSON" entity. Hence, Spacy NER model has been trained on Google+ sample data so as to enable the model to recognize Google+ usernames correctly. After training, the model is now able to identify Google+ usernames as "PERSON" entity.

After entity recognition, the next subtask is to extract usernames. Usually, there are three possibilities for usernames - username consisting of first name only (example, +akimatzu) or username consisting of first name and last name (example,+jonathon hafner) or username consisting of first name, middle name and last name (example,+lionel khutso mallela). To cover all these cases, authors have considered unigrams, bigrams, and trigrams for extraction of usernames from the text. Any unigram or bigram or trigram identified as "PERSON" by the model is extracted as the resultant username.

However, there is also a possibility of the presence of more than one username in the same text. For example, "+shahroz ahmed going to laugh at that +themoises1213 dudes been so broke". The use of n-grams generates a list for entire text and for the present subtask, these n-grams are needed for extracting usernames only. The considered n-gram lists for the above example are as follows-

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Unigrams = ['+shahroz', 'ahmed', 'going', 'to', 'laugh', 'at', 'that', '+themoises1213', 'dudes', 'been', 'so', 'broke']
Bigrams = [('+shahroz', 'ahmed'), ('ahmed', 'going'), ('going', 'to'), ('to', 'laugh'), ('laugh', 'at'), ('at', 'that'), ('that', '+themoises1213'), ('+themoises1213', 'dudes'), ('dudes', 'been'), ('been', 'so'), ('so', 'broke')]
Trigrams = [('+shahroz', 'ahmed', 'going'), ('ahmed', 'going', 'to'), ('going', 'to', 'laugh'), ('to', 'laugh', 'at'), ('laugh', 'at', 'that'), ('at', 'that', '+themoises1213'), ('that', '+themoises1213', 'dudes'), ('+themoises1213', 'dudes', 'been'), ('dudes', 'been', 'so'), ('been', 'so', 'broke')]

Instead of entire list, the desired results for username extraction are - ['+Shahroz', ('+Shahroz Ahmed'), ('+Shahroz Ahmed going'), '+themoises1213', ('+themoises1213 dudes') and ('+themoises1213 dudes been')]. For the desired result, an index variable, 'indx' has been used to store the index of the string starting with '+' and from the index onwards, unigram, bigram and trigram are extracted as username.

After username extraction, the next subtask is username replacement with word "username". However, even the username replacement subtask is not straightforward. In case, if trigram is identified as "PERSON" by the NER model then its corresponding bigrams and unigrams will also be identified as "PERSON". Similarly, if a bigram is identified as a "PERSON" then its corresponding unigrams will also be identified as "PERSON" by the model. For example, if username is '+shahroz ahmed khan' then its corresponding unigram, bigram and trigram lists are as follows-

Unigrams = ['+shahroz', 'ahmed', 'khan']
Bigrams = [('+shahroz', 'ahmed'), ('ahmed', 'khan')]
Trigrams = [('+shahroz', 'ahmed', 'khan')]

From the lists, it is clear that '+shahroz ahmed khan' is a "PERSON", in fact, +Shahroz Ahmed', 'Ahmed Khan', '+Shahroz', 'Ahmed' and 'khan' are all "PERSON" entities as well. Due to this reason, the username replacement in the proposed algorithm has been done only after all three cases are executed. The procedure for handling multiword usernames is as follows -

Algorithm for Multiword Username Replacement

Data: Raw Text
Result: multiword username replacement with word "username"

1 initialization of Boolean variables, uni_flag, bi_flag and tri_flag to 0
/* Entity Recognition and Username Extraction */
2 *for* each word in text *do*
3 *if* word starts with '+' and length of word is greater than 1 *then*
4 find the index of word and store the index value in 'indx' variable;
5 extract substring starting from the index value stored in 'indx';
6 load trained NER model;
7 tokenize text and store tokens in variable 'tokens' ;
8 If word entity type is 'PERSON' then store the word in 'uni_str' variable and set uni_flag=1;
9 If tokens length is greater than or equal to 2 then generate bigrams and store bigrams in a string variable named bi_str;
10 If bi_str entity type is 'PERSON' then set bi_flag=1;
11 If tokens length is greater than or equal to 3 then generate trigrams and store trigrams in a string variable named tri_str;
12 If tri_str entity type is 'PERSON' then set tri_flag=1;
/* Username Replacement */
13 If uni_flag is set then replace uni_str with word "username" in the text;
14 ElseIf bi_flag is set then replace bi_str with word "username" in the text;
15 Else replace tri_str with word "username" in the text;
16 End

part of their spelling (example, cool). In the above example, 'loooove' is replaced by 'looove'. After this pre-processing step, the next step is punctuations filtering.

13) Filtering of Punctuation marks: In this pre-processing step, instead of removing all, punctuation marks have been filtered. Punctuation marks - ".","?", and "+" have been retained in the data. Reason for retaining '.' in the data is the presence of '.' in between digits (example, 5.7 inch display). The '?' (Question mark) has been retained to identify

After handling usernames, the next pre-processing step is filtering repeating letters.

12) Filtering repeating letters: Users often repeat letter(s) in a word to express their opinion by putting stress via repeating letters. For example, '#Iphone looooove you so muchhh'.

In the data collected, though the frequency of repeating characters is not very high, still such words exist. So, this has been added as a part of the pre-processing sequence. In this step, if a letter is repeating more than thrice then it is replaced by exactly three occurrences. The lower bound for replacement of repeating character has been fixed three because there are words that contain two repeating letters as a

Handling of Numerals: From the collected data, it has been observed that the removal of numerals from data can lead to loss of information provided by various users. Also, numerals are needed when there is a comparison between two different versions of a product (example, Samsung note 9 vs Samsung note 10) or when referring to a specific product model (example, Samsung note 9 is awful). Hence, numbers have been retained in the proposed pre-processing sequence.

This step concludes the proposed data pre-processing sequence. The proposed pre-processing ordered sequence has been summarized in the algorithm given below –*14) Stop words Removal:* In this pre-processing step, all non-informative stop words such as "the", "is", "on", "an" etc. have been removed with the help of the nltk Stop Word dictionary. After removing stop words, the next step is dealing with numbers present in the data. differentiating between mobile models (example, motog5s and motog5s+).

The next pre-processing step in the sequence is Stop words removal.

15) interrogative sentences which is a part of authors' future work and '+' (plus) has been retained for the purpose of

Algorithm for Proposed Pre-processing System

Data: Raw Text
Result: Pre-processed text

1. Identify hashtag(s) in the raw text. Remove '#' (hash) symbol from it. If a hashtag is a combination of unspaced compound words then separate compound words into individual words on the basis of their letter cases and proceed to the next pre-processing step. If no hashtags are present in the text, then skip the current step and go to the next step;
2. Convert raw text into lowercase and proceed to the next step;
3. Translate code mixed data into English and proceed to the next step. If no code mixed data is present then skip the current step and go to next step;
4. Replace HTML codes with the help of the compiled dictionary and proceed to the next step. Otherwise, skip the current step and go to the next step;
5. Identify acronyms with spaces and combine them by removing in-between spaces and move to the next step. If no acronyms with spaces exist in the text then skip the current step and go to next step;
6. Expand acronyms with the help of the compiled dictionary and move to the next step. If no acronyms are present in the text then skip the current step and go to the next step;
7. Identify negation and non-negation contractions in the text. If present, expand them with the help of the compiled dictionary and move to the next step. Otherwise, skip the current step and go to the next step;
8. If any emojis or emoticons are present in the text then replace them with their corresponding meaning from the compiled dictionaries and proceed to the next step. Otherwise, skip the current step and go to next step;
9. Remove URLs from the text and proceed to the next step. If no URL exists in the text then skip the current step and go to the next step;
10. Remove HTML tags from the text and proceed to the next step. If no HTML tags are present in the text then skip the current step and go to the next step;
11. Replace usernames starting with '@' or '+' with the word 'username'. If any multiword username is present in the text, call multiword username replacement algorithm, otherwise, replace username via regular expression. After replacement, proceed to the next step. If no username exists in the text then skip the present step and go to the next step;
12. If a letter in a word is repeating more than thrice then replace such letters by exactly three occureces. Otherwise, skip the current step and go to the next step;
13. Remove punctuation marks except ".","?", and "+" from the text and move to the next step. If no punctuation marks are present then skip the present step and go to the next step;
14. Remove all non-informative stop words from the text and proceed to last step. If stop words do not exist in the text then skip the present step and go to the last step;
15. The last step is numerals handling. In this step, all numerals have been retained in the text;

The next section shows experimental results to validate the significance of the proposed work.

IV. EXPERIMENTAL RESULTS

To justify the significance of the proposed work, this section presents its comparison with the existing pre-processing approaches. The proposed system has been compared with the authors work in [9] and [19]. The present pre-processing methodology is more comprehensive than the existing ones in many aspects. First, it has been implemented on more than one social networks. Second, its elaborate handling of emojis and emoticons and third, inclusion of hashtag handling. Moreover, the proposed system has more comprehensive existing dictionaries along with newly compiled dictionaries in comparison to the existing ones. For comparison, the experimental pre-processing results have been demonstrated on two different examples.

To avoid any sort of biasedness in experimental results, the first example has been taken from publicly available SemEval2017 training data. The second example is from our own collected data from APIs and the sentiment polarity of this example has been marked by independent human annotators.

In the second example, Unicode characters,<u+1f64f> and <u+1f60a>, corresponds to emojis 🙏 and 😊, respectively. For demonstration purpose, these Unicode characters have been replaced with their corresponding emojis to maintain readability of text. Table XI shows these two examples with their sentiment polarities.

Table XI: Examples With Their Sentiment Polarities

S.No.	Example Text	Sentiment Polarity
1	Paul Dunne going to make history tomorrow #IrishEyesAreSmiling.	Positive
2	Samsung I o y 🙏😊	Positive

Both examples have been pre-processed separately and the sentiment polarity has been computed by VADER at each pre-processing step. After each pre-processing step, the text ouput becomes an input for the subsequent pre-processing step. Table XII shows the results with their polarity score after every pre-processing step using the proposed sequence.

Table XII: Pre-Processing Of Example 1 As Per Proposed Sequence

Example 1: Paul Dunne going to make history tomorrow #IrishEyesAreSmiling			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	Hashtags Handling	Paul Dunne going to make history tomorrow Irish Eyes Are Smiling	0.4588
2	Converting to lowercase	paul dunne going to make history tomorrow irish eyes are smiling	0.4588
3	Translating code mixed data	No change in Text	0.4588
4	Replacing HTML Codes	No change in Text	0.4588
5	Combining Acronyms	No change in Text	0.4588
6	Expanding Acronyms	No change in Text	0.4588
7	Negation and non-negation	No change in Text	0.4588
8	Emojis and Emoticons Handling	No change in Text	0.4588
9	Removing URLs	No change in Text	0.4588
10	Removing HTML Tags	No change in Text	0.4588
11	Replacing Usernames	No change in Text	0.4588
12	Filtering Repeating Letters	No change in Text	0.4588
13	Punctuations Filtering	No change in Text	0.4588
14	Stop words Removal	paul dunne going make history tomorrow irish eyes are smiling	0.4588
15	Handling of Numerals	No change in Text	0.4588

From Table XII, it can be observed that splitting hashtag with unspaced compound words into independent words has the ability to change the sentiment polarity of the entire text. In the proposed work, such hashtags have been splitted on the basis of their letter cases and therefore, hashtag handling is the first preprocessing step in the proposed system. This step is followed by converting text to lowercase. It is important to note here that the sequence order of preprocessing steps is very crucial during data preprocessing. Merely changing the sequence order of first two preprocessing steps will result in different sentiment polarity i.e., in the above example, the hashtag, #IrishEyesAreSmiling will be first converted to lower case as #irisheyesaresmiling and then, the hashtag handling approach will not be able to split the hashtag and the compound score given by VADER will be 0.0 i.e., Neutral.

Table XIII shows the pre-processing of the same example as per the pre-processing approach proposed by Itisha et al. Details of their pre-processing steps can be referred in [9].

Table XIV shows the pre-processing of the same example as per the pre-processing approach proposed Jianqiang et al. Details of their pre-processing steps can be referred in [19].

From the results in Table XII, Table XIII and Table XIV, it can be observed that hashtag handling is important as it can also affects the sentiment polarity of the text.

Table XV, Table XVI and Table XVII show the pre-processing results of example 2 as per the proposed ordered sequence, Itisha et al. and Jianqiang et al. respectively.

In the second example, "I o y" is an acronym whose standard full form is "I owe you", but the existing approaches have treated this acronym as individual stop words, resulting in loss of user's context. Whereas, in the proposed methodology, "i o y" has been combined as "ioy" and then expanded by acronym dictionary as "i owe you". Also, the core of the present work, i.e., emoji handling, has a huge impact in changing the sentiment polarity of text. It can be seen from Table XVI and Table XVII that without emoji handling, sentiment polarity of example 2 is Neutral whereas, with emoji handling, as shown in Table XV, polarity changes from Neutral to Positive.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table XIII: Pre-Processing Of Example 1 As Per Itisha Et Al., 2017

Example 1: Paul Dunne going to make history tomorrow #IrishEyesAreSmiling			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	URLs and Usernames Removal	No change in text	0.0
2	Extra whitespaces and Newline Removal	No change in text	0.0
3	Stop words Removal	Paul Dunne going make history tomorrow #IrishEyesAreSmiling	0.0
4	Removal of Prepositions	No change in Text	0.0
5	HTML characters Removal	No change in Text	0.0
6	Removal of Hashtag	Paul Dunne going make history tomorrow IrishEyesAreSmiling	0.0
7	Punctuation Removal	No change in Text	0.0
8	Removal of Proper Noun and foreign words	No change in Text	0.0
9	Acronym lookup	No change in Text	0.0
10	Apostrophe lookup	No change in Text	0.0
11	Misspell words replacement	No change in Text	0.0
12	Elongated words normalization	No change in Text	0.0
13	Emoticon Replacement	No change in Text	0.0
14	Replace negation words	No change in Text	0.0

Table XIV: Pre-Processing Of Example 1 As Per Jianqiang Et Al., 2017

Example 1: Paul Dunne going to make history tomorrow #IrishEyesAreSmiling			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	Replacing negative mentions	No change in text	0.0
2	Removing URLs	No change in text	0.0
3	Reverting words containing repeated letters	No change in text	0.0
4	Removing numbers	No change in Text	0.0
5	Stop words Removal	Paul Dunne going make history tomorrow #IrishEyesAreSmiling	0.0
6	Expanding Acronym	No change in Text	0.0

Table XV: Pre-Processing Of Example 2 As Per Proposed Sequence

Example 2: Samsung I o y <u+1f64f> <u+1f60a> 🙏 😊			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	Hashtags Handling	No change in text	0.0
2	Converting to lowercase	samsung i o y <u+1f64f><u+1f60a>	0.0
3	Translating code mixed data	No change in Text	0.0
4	Replacing HTML Codes	No change in Text	0.0
5	Combining Acronyms	samsung ioy <u+1f64f><u+1f60a>	0.0
6	Expanding Acronyms	samsung i owe you <u+1f64f><u+1f60a>	0.0
7	Negation and non-negation	No change in Text	0.0
8	Emojis and Emoticons Handling	samsung i owe you expressing request, respect or thanks smiling face with smiling eyes	0.9001
9	Removing URLs	No change in Text	0.9001
10	Removing HTML Tags	No change in Text	0.9001
11	Replacing Usernames	No change in Text	0.9001
12	Filtering Repeating Letters	No change in Text	0.9001
13	Punctuations Filtering	No change in Text	0.9001
14	Stop words Removal	samsung owe expressing request respect thanks smiling face smiling eyes	0.9001
15	Handling of Numerals	No change in Text	0.9001

Table XVI: Pre-Processing Of Example 2 As Per Itisha Et Al., 2017

Example 2: Samsung I o y <u+1f64f> <u+1f60a> 🙏😊			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	URLs and Usernames Removal	No change in text	0.0
2	Extra whitespaces and Newline Removal	No change in text	0.0
3	Stop words Removal	Samsung <u+1f64f><u+1f60a>	0.0
4	Removal of Prepositions	No change in Text	0.0
5	HTML characters Removal	No change in Text	0.0
6	Removal of Hashtag	No change in Text	0.0
7	Punctuation Removal	Samsung u1f64fu1f60a	0.0
8	Removal of Proper Noun and foreign words	u1f64fu1f60a	0.0
9	Acronym lookup	No change in Text	0.0
10	Apostrophe lookup	No change in Text	0.0
11	Misspell words replacement	No change in Text	0.0
12	Elongated words normalization	No change in Text	0.0
13	Emoticon Replacement	No change in Text	0.0
14	Replace negation words	No change in Text	0.0

Table XVII: Pre-Processing Of Example 2 As Per Jianqiang Et Al., 2017

Example 2: Samsung I o y <u+1f64f> <u+1f60a> 🙏😊			
S.No.	**Pre-processing Steps**	**Text After Pre-processing Steps**	**Vader Score**
1	Replacing negative mentions	Samsung I o y <u+1f64f><u+1f60a>	0.0
2	Removing URLs	No change in text	0.0
3	Reverting words containing repeated letters	No change in text	0.0
4	Removing numbers	No change in Text	0.0
5	Stop words Removal	Samsung <u+1f64f><u+1f60a>	0.0
6	Expanding Acronym	No change in Text	0.0

Hence, it can be concluded that emojis are the rich source of sentiments and must be handled during data pre-processing to achieve better sentiment classification results.

Table XVIII shows the experimental results of different approaches whereas, Table XIX shows comparison of sentiment polarities of each approach with expected sentiment polarities. From Table XIX, it is evident that the existing approaches are not producing the desired sentiment polarities whereas the proposed system outshines by producing expected sentiment polarities.

Table XVIII: Sentiment Polarities Of Proposed Work And Existing Approaches

S.No.	**Text**	**Proposed Work**		**Itisha Et Al.**		**Jianqiang Et Al.**	
		Score	**Polarity**	**Score**	**Polarity**	**Score**	**Polarity**
1	Paul Dunne going to make history tomorrow #IrishEyesAreSmiling	.4588	Positive	0.0	Neutral	0.0	Neutral
2	Samsung I o y 🙏😊	.9001	Positive	0.0	Neutral	0.0	Neutral

Table XIX: Sentiment Polarities Comparision of Experimental Results with Actual Polarities

S.No.	Text	Actual Polarity	Proposed Work		Itisha Et Al.		Jianqiang Et Al.	
			Polarity	Result	Polarity	Result	Polarity	Result
1	Paul Dunne going to make history tomorrow #IrishEyesAreSmiling	Positive	Positive	Match	Neutral	Mismatch	Neutral	Mismatch
2	Samsung I o y 🙏😊	Positive	Positive	Match	Neutral	Mismatch	Neutral	Mismatch

V. CONCLUSION

Data preprocessing is a fundamental step in sentiment analysis due to the presence of unstructured data on social networks. For this reason, data of two different social networks (Twitter and Google+) has been analyzed in detail. From the analysis, it was evident that the unstructured data on different social networks presents many preprocessing challenges due to ever changing writing style of netizens. Therefore, to meet these challenges, a new ordered sequence of preprocessing steps has been proposed and implemented. The new ordered sequence has hashtag handling as an additional step along with an algorithm for handling multiword usernames on Google+. Observations show that emojis and emoticons are widely used by netizens due to their ability to visualize communication in the digital world and thereby, affects the sentiment polarity of the entire text. Therefore, in the proposed system, emojis and emoticons have been handled in detail. New dictionaries for emojis and emoticons have been compiled to provide language to the emotional contents expressed by various emojis. Existing dictionaries have also been modified to make them more comprehensive for lookup task. Experiments have been carried out to compare the proposed system with the existing approaches. The experimental results show that the proposed system outsmarts the existing ones due to the handling of hashtags and emojis along with implementation of preprocessing steps in the suggested sequence order.

REFERENCES

1. B. Liu, “Sentiment analysis and opinion mining,” Synthesis lectures on human language technologies, vol. 5, no. 1, pp. 1–167, 2012.
2. A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N Project Report, Stanford, vol. 1, no. 12, 2009.
3. 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.
4. H. Saif, Y. He, and H. Alani, “Alleviating data sparsity for twitter sentiment analysis.” CEUR Workshop Proceedings (CEUR-WS. org), 2012.
5. E. Haddi, X. Liu, and Y. Shi, “The role of text pre-processing in sentiment analysis,” Procedia Computer Science, vol. 17, pp. 26–32, 2013.
6. Y. Garg and N. Chatterjee, “Sentiment analysis of twitter feeds,” in International Conference on Big Data Analytics. Springer, 2014, pp. 33–52.
7. 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.
8. A. Kumar and T. M. Sebastian, “Sentiment analysis on twitter,” International Journal of Computer Science Issues (IJCSI), vol. 9, no. 4, p. 372, 2012.
9. I. Gupta and N. Joshi, “Tweet normalization: A knowledge based approach,” in 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS). IEEE, 2017, pp. 157–162.
10. T. Agrawal and A. Singhal, “An efficient knowledge-based text preprocessing approach for twitter and google+.” in 2019, International Conference on Advances in Computing and Data Sciences (ICACDS). Springer, [Presented], 2019.
11. P. K. Novak, J. Smailovi´c, B. Sluban, and I. Mozeti˘c, “Sentiment of emojis,” PloS one, vol. 10, no. 12, p. e0144296, 2015.
12. S.Wijeratne, L.Balasuriya, A.Sheth, andD.Doran, “Emojinet:Building a machine readable sense inventory for emoji,” in International conference on social informatics. Springer, 2016, pp. 527–541.
13. C. H. E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf, 2014.
14. Y. Bao, C. Quan, L. Wang, and F. Ren, “The role of pre-processing in twitter sentiment analysis,” in International Conference on Intelligent Computing. Springer, 2014, pp. 615–624.
15. Z. Jianqiang, “Pre-processing boosting twitter sentiment analysis?” in 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE, 2015, pp. 748– 753.
16. G. Angiani, L. Ferrari, T. Fontanini, P. Fornacciari, E. Iotti, F. Magliani, and S. Manicardi, “A comparison between preprocessing techniques for sentiment analysis in twitter.” in KDWeb, 2016.
17. A. Krouska, C. Troussas, and M. Virvou, “The effect of preprocessing techniques on twitter sentiment analysis,” in 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA). IEEE, 2016, pp. 1–5.
18. T. Singh and M. Kumari, “Role of text pre-processing in twitter sentiment analysis,” Procedia Computer Science, vol. 89, pp. 549–554, 2016.
19. Z. Jianqiang and G. Xiaolin, “Comparison research on text preprocessing methods on twitter sentiment analysis,” IEEE Access, vol. 5, pp. 2870–2879, 2017.
20. T. Agrawal, A. Singhal, and S. Agarwal, “A comparative study of potential of various social networks for target brand marketing,” in 2016 International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds. IEEE, 2016, pp. 305–311.

AUTHORS PROFILE

Tripti Agrawal is pursuing her Ph.D. in Computer Science from University of Delhi, Delhi. She did her Master of Computer Application (MCA) from Dr. A.P.J. Abdul Kalam Technical University and PGDCA from University of Delhi, Delhi. Her research areas include Social Networks, Natural Language Processing.

Dr. Archana Singhal is working as an Associate Professor, Department of Computer Science, Indraprastha College for Women, University of Delhi, New Delhi. Her research areas include Natural Language Processing, Semantic Web, Multi-agent Systems, Information Retrieval and Ontologies, Secure Software Systems and Social Networks. She has many publications to her credit in reputed journals and International conferences.