International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 5 – APRIL 2018 (SPECIAL ISSUE).

EXTRACTION OF DISASTER BASED HASH-TAGS IN TWITTER

[#1] S. Angela angelsroy97@gmail.com
[*2] C. Blessy Daffodil blessydaff25@gmail.com
[*3] D.Sterlin Rani sterlinrani@gmail.com
[#1,2] *U.G. Student, Department of Computer Science and Engineering, Kings Engineering College, Chennai.*
[#4] *Assistant Professor, Department of Computer Science and Engineering, Kings Engineering College, Chennai*

.

Abstract

Twitter is a online news service where users can post messages called tweets. These tweets are restricted to 280 characters. Registered users can post their tweets whereas the unregistered users can only read the tweets. Twitter provides tweets about all the real time trending topics. In this paper we extract the disaster based hashtags and classify the extracted tweets as affected and unaffected. The sentiment of the tweets is analyzed. Hash-tag is a word or phrase preceded by a hash sign (#), used on social media websites and applications, especially Twitter, to identify messages on a specific topic. The extracted tweets are preprocessed to remove the stop words and to perform stemming. The duplicate tweets are removed in-order to prevent the repetition of tweets. The Naive Bayes algorithm is used to classify the tweets. This project is implemented using Python and the experimental result show that it produces accurate result.

Keywords – twitter, preprocessing, sentiment analysis, classification.

I. INTRODUCTION

Twitter is the most popular micro blogging site where the users search for the recent real time information such as breaking news, trending topics, etc. Tweets are short text messages which are restricted to 280 characters. Twitter was launched in 2006 and its popularity is dramatically increasing. The tweets can be retrieved using keywords or hashtags. Hashtags is a word that is preceded by a hash sign (#). The trending topic names may or may not be indicative of the kind of information that people are tweeting about unless it is associated with a hashtag. For example, #Disaster indicates that the people are tweeting about disaster. Our goal is to extract all the live tweets that contains the disaster hash-tag and to classify it into two category.

(i) those who are affected and need help
(ii) those who are unaffected.

Then the sentiment of each tweet is identified. To classify the tweets we use supervised machine learning algorithm called the Naive Bayes. It can be used for both classification and regression challenges. However, it is mostly used in classification problems. Naive Bayes is a supervised machine learning algorithm which is used to classify data into predefined classes.

The remainder of this paper is organized as II. Relatedworks, III.System analysis, IV.System Development,V.Experimentalresults,Conclusion

II. RELATED WORKS

A number of recent papers has been studied.

DR. S. Vijayarani [1] defined about the preprocessing methods. It is the first step in the text mining process. In this paper, the three key steps of preprocessing namely, stop words removal, stemming and TF/IDF algorithms are defined. Text mining is the process of seeking or extracting the useful information from the textual data. It tries to find interesting patterns from large databases.

Shanshan Zhang[2] described the problem of retrieving disaster-related tweets shortly after the onset of a disaster is addressed. In such a scenario, we can expect to have access to a very limited number of labeled tweets. The accuracy of classifiers trained on such small data would be limited. To remedy this problem, a semi-supervised approach that can utilize a large unlabeled corpus of tweets to create word clusters and use them as features for classification is proposed. Experiments on Twitter data from 6 disasters strongly indicate that the proposed semi-supervised approach could most often result in the improvements in accuracy as compared to the traditional supervised learning approach that uses feature selection on the bag of words features. This study also provides useful insights into different modeling choices when using the proposed approach.

Hassan Saif [3] defined a novel approach of adding semantics as additional features into the training set for sentiment analysis. For each extracted entity (e.g. iPhone) from tweets, semantic concept is added to it (e.g. “Apple product”) as an additional feature, and measure the correlation of the representative concept with negative or positive sentiment. This approach was applied to predict sentiment for three different Twitter datasets. The results show an average increase of F harmonic accuracy score for identifying both negative and positive sentiment of around 6.5% and 4.8% over the baselines of uni-grams and part-of-speech features respectively. We find that semantic features produce better Recall and F score when classifying negative sentiment, and better Precision with lower Recall and F score in positive sentiment classification. Results indicates that the semantic approach is more appropriate when the datasets being analyzed are large and cover a

wide range of topics, whereas the sentiment-topic approach was most suitable for relatively small datasets with specific topical foci. adding such features to the analysis could help identifying the sentiment of tweets that contain any of the entities that such concepts represent, even if those entities never appeared in the training set.

Kevin Stowe [4] defined an annotation schema for identifying relevant tweets as well as the more fine-grained categories. This paper focuses on Twitter data generated before, during, and after Hurricane Sandy, which impacted New York in the fall of 2012. Focusing on the 2012 Hurricane Sandy event, this paper presented classification methods for
(i) filtering tweets relevant to the disaster, and
(ii) categorizing relevant tweets into fine-grained categories such as preparation and evacuation. This type of automatic tweet categorization can be useful both during and after disaster events. During events, tweets can help crisis managers, first responders, and others take effective action. After the event, analysts can use social media information to understand people's behavior during the event.

Jaime T. Ballena [5] has build some machine learning models that can automatically detect informative disaster-related tweets. Two machine learning algorithms, Naive Bayes and Support Vector Machine (SVM), were used to build models for the automatic classification of the tweets, and these models were evaluated across the metrics of accuracy, precision, recall, area under curve and F-measure. Relevant tweets shared by users are a vital source of information and is useful in understanding and visualizing the situation of affected parties. During the Habagat incident, subscribers used this medium to broadcast both informative and uninformative tweets. Based on the Habagat statistics, uninformative tweets outnumbered informative tweets with a 65% to 35% ratio. Subscribers expressed their opinions and emotions through their posted tweets. Although more uninformative tweets were posted, the informative tweets were rapidly and repeatedly sent because these were re-tweeted. This evidently implies that the informative tweets contain vital and urgent information that can provide significantly needed information for situational awareness of the public and disaster response units. Moreover, disaster-related tweets can be automatically classified using the bag of words approach and classifying algorithms SVM and Naïve Bayes.

III. SYSTEM ANALYSIS

Existing System

In the existing system the tweets about the disaster are gathered and analyzed. The gathered tweets are classified as relevant and irrelevant using support vector machine algorithm.

Disadvantage

1. The accuracy of the classified tweets is low.
2. The opinion of the tweet is not analyzed.

Proposed System

In our proposed system we gathered the disaster related tweets and classified it as affected and unaffected using Naive Bayes algorithm and the opinion of each tweet is analyzed as positive, negative or neutral.

Advantage

1.The AUC, F1 score and recall was high.
2.The opinion of each tweet is analyzed

IV. SYSTEM DEVELOPMENT

Modules

1. Extraction of tweets
2. Preprocessing
3. Classification
4. Sentiment Analysis

Extraction of tweets

In order to extract the tweets from twitter , it is necessary to create a twitter application and obtain the API keys and secret keys. The steps for creating twitter application is as follows.
1.Go to https://dev.twitter.com/apps/new and log in, if necessary. Click on create new application.
2. Enter your Application Name, Description and your website address. You can leave the callback URL empty.
3. Accept the TOS, and solve the CAPTCHA.
4. Submit the form by clicking the Create your Twitter Application
5. Copy the consumer key (API key) and consumer secret from the screen into your application

System Architecture

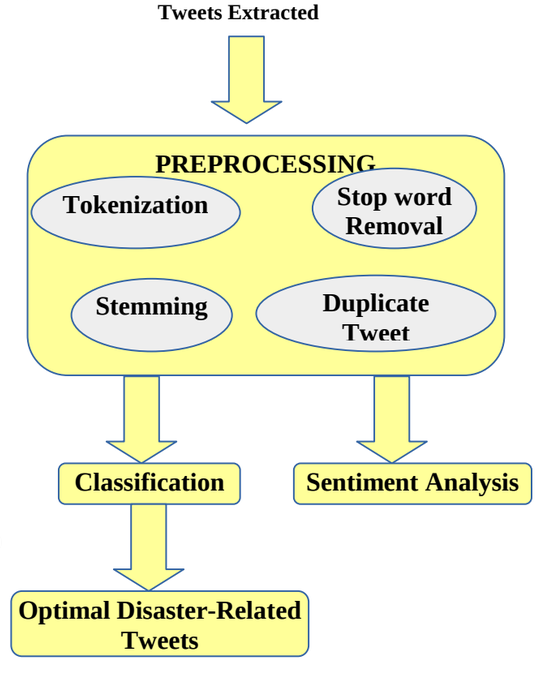

Fig 1: Architecture Diagram

P***reprocessing***

Data preprocessing refers to cleansing of data. For example, before performing sentiment analysis of twitter data, you may want to strip out any HTML tags, white spaces, expand abbreviations and split the tweets into lists of the words they contain. When analyzing spatial data you may scale it so that it is unit-independent, that is, so that your algorithm doesn't care whether the original measurements were in miles or centimeters. However, preprocessing data does not occur in a vacuum. This is just to say that preprocessing is a means to an end and there are no hard and fast rules: there are standard practices, as we shall see, and you can develop an intuition for what will work but, in the end, preprocessing is generally part of a results-oriented pipeline and its performance needs to be judged in context.

Preprocessing has four steps

1. Tokenization
2. stop word removal
3. Stemming
4. Duplicate tweet removal

Tokenization

This is the process of splitting a text into individual words or sequences of words (n-grams) Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining. Tokenization relies mostly on simple heuristics in order to separate tokens by following a few steps:

- Tokens or words are separated by whitespace, punctuation marks or line breaks
- White space or punctuation marks may or may not be included depending on the need
- All characters within contiguous strings are part of the token. Tokens can be made up of all alpha characters, alphanumeric characters or numeric characters only.

Tokens themselves can also be separators. For example, in most programming languages, identifiers can be placed together with arithmetic operators without white spaces.

Stop words removal

Stop word removal is one of the most commonly used preprocessing steps across different NLP applications. Stop words are words which are filtered out before or after processing of natural language data (text). The idea is simply removing the words that occur commonly across all the documents in the corpus. Typically, articles and pronouns are generally classified as stop words. These words have no significance in some of the NLP tasks like information retrieval and classification, which means these words are not very discriminative. Removing stop words reduces the dimensionality of term space. The most common words in text documents are articles, prepositions, and pro-nouns, etc. that does not give the meaning of the documents. These words are treated as stop words. Example for stop words: the, in, a, an, with, etc. Stop words are removed from documents because those words are not measured as keywords in text mining applications

Stemming

Goal of stemming is to reduce variations of each word due to inflection or derivation to a common stem. It improves effectiveness by providing a better match between query and a relevant document. User who is searching for “swimming” might be interested in documents with “swim”.

Algorithm – Porter Stemmer

Porters stemming algorithm is one of the most popular stemming algorithm proposed in 1980. This algorithm has about 60 rules and very easy to understand. it is faster. It has effectively traded space for time, and with its large suffix set it needs just two major steps to remove a suffix

Step 1

• Replace “ing” with “e”, if number of consonant-vowels switches, called measure, is greater than 3.
Eg: liberating --> liberate, facilating--> facilate
• Remove “es” from words that end in “sses” or “ies”
Eg: passes --> pass, cries --> cri
• Remove “s” from words whose next to last letter is not an “s”
Eg: runs --> run, fuss --> fuss
• If word has a vowel and ends with “eed” remove the “ed”
Eg: agreed --> agre, freed --> freed
• Remove “ed” and “ing” from words that have other vowel
Eg: dreaded --> dread, red --> red, bothering --> bother, bring --> bring
• Remove “d” if word has a vowel and ends with “ated” or “bled”
Eg: enabled --> enable, generated --> generate
• Replace trailing “y” with an “I” if word has a vowel
Eg: satisfy --> satisfi, fly --> fly

Step 2

• With what is left, replace any suffix on the left with

suffix on the right- only if the consonant-vowels measure >0
tional → tion
conditional --> condition
ization → ize
nationalization --> nationalize
iveness → ive
effectiveness --> effective
fulness → ful
usefulness --> useful
ousness → ous
nervousness --> nervous
ousli → ous
nervously --> nervous
entli → ent
fervently --> fervent
iveness → ive
inventiveness --> inventive
biliti → ble
sensibility --> sensible

Step3:
Remove trailing "e" if word does not end in a vowel
– hinge --> hing
– free --> free

Duplicate tweet removal:

The extracted tweets contains some tweets that are repeated two or more times. The duplicate tweet removal is used to remove all the tweets that appears more than once.

CLASSIFICATION

Naive bayes is a supervised machine learning algorithm which is used to classify data into predefined classes. It uses the concept of conditional probability to classify the test data.

Conditional Probability

It helps us to find the probability that something will happen given that something else has happened.
Consider two events A and B, then

P(A and B) = P(A) * P(B | A)

Note that P(B | A) is probability that B happens given that A has already happened.

Bayes Rule

The rule helps us to know how often A happens given that B has already happened P(A | B), when we know how often B happens given that A has already happened P(B | A).

Naive Bayes Classifier finds the probability of every feature then it selects the
outcome with highest probability.

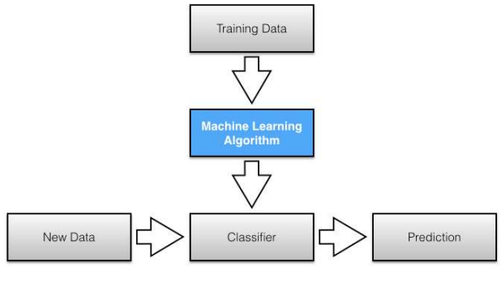

Fig 2: Working of Naive Bayes

SENTIMENT ANALYSIS

Sentiment Analysis (SA) or Opinion Mining (OM) is the computational study of people's opinions, attitudes and emotions toward an entity. There are three main classification levels in SA: document-level, sentence-level, and aspect-level SA. Document-level SA aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic). Sentence-level SA aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, Sentence-level SA will determine whether the sentence expresses positive or negative opinions. Aspect-level SA aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entities and their aspects. The opinion holders can give different opinions for different aspects of the same entity like this sentence. Semantic analysis is derived from the WordNet database where each term is associated with each other. This database is of English words which are linked together. If two words are close to each other, they are semantically similar. More specifically, we are able to determine synonym like similarity. We map terms and examine their relationship in the ontology. The key task is to use the stored documents that contain terms and then check the similarity with the words that the user uses in their sentences. Thus it is helpful to show the polarity of the sentiment for the users.

V. EXPERIMENTAL RESULTS

The experimental result show that the naive bayes works well in our dataset. The area of curve, ca, F1 score and recall was high compared to SVM. But Svm outperformed naive bayes in terms of recall.

Method	AUC	CA	F1	Precision	Recall
Neural Network	0.572	0.577	0.719	0.580	0.946
SVM	0.509	0.510	0.532	0.584	0.489
Naive Bayes	0.552	0.569	0.725	0.569	0.999
Logistic Regression	0.499	0.569	0.725	0.569	1.000

Table 1: Comparision between algorithms

CONCLUSION

Twitter is a medium used by subscribers to broadcast disaster-related tweets. In this paper we used Naive Bayes for classifying the tweets as affected and unaffected. Our results show that Naive Bayes

outperformed SVM in terms of AUC, F1 score and recall and provided an exact result. A real-time system that can detect and filter information from the disaster-relevant tweets is developed for an effective and efficient disaster response management.

REFERENCES

[1]Dr. S. Vijayarani, Ms. J. Ilamathi, Ms. Nithya Assistant Professor , M. Phil Research Scholar Department of Computer Science, School of Computer Science and Engineering,Bharathiar University, Coimbatore, Tamilnadu, India Preprocessing Techniques for Text Mining - An Overview
[2]Beverly Estephany Parilla-Ferrer, Proceso L. Fernanez Jr., PhD, and Jaime T. Ballena IV, PhD Automatic Classification of Disaster-Related Tweets
[3]Kevin Stowe, Michael Paul, Martha Palmer, Leysia Palen, Ken Anderson University of Colorado, Boulder, CO 80309 Identifying and Categorizing Disaster-Related Tweets
[4]Hassan Saif, Yulan He and Harith Alani Knowledge Media Institute, The Open University, United Kingdom Semantic Sentiment Analysis of Twitter
[5]Shanshan Zhang Slobodan Vucetic Temple University 1805 N. Broad Street Philadelphia, PA Temple University Semi-supervised Discovery of Informative Tweets During the Emerging Disasters
[6] Rishabh Upadhyay, Akihiro Fujii Department of Applied Informatics,Hosei University, Tokyo, Japan Semantic Knowledge Extraction from Research Documents
[7]. Yan Huang, Zhi Liu, Phuc Nguyen Location Based event search in social text.
[8]. Hien To, Sumeet agarwal, Seon Ho Kim, Integrated Media Systems Center, University of Southern California, Los Angeles, USA. On identifying Disaster-Related tweets: Matching-based or Learning based ?