2019

Analyzing Twitter Feeds to Facilitate Crises Informatics and Disaster Response During Mass Emergencies

Arshdeep Kaur
Technological University Dublin

Analyzing Twitter Feeds to Facilitate Crises Informatics and Disaster Response during Mass Emergencies

Arshdeep Kaur

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Analytics)

January 2019

Declaration

I certify that this dissertation which I now submit for examination for the award of M.Sc. in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Arshdeep Kaur

Date: 4th January 2019

Abstract

It is a common practice these days for general public to use various micro-blogging platforms, predominantly Twitter, to share ideas, opinions and information about things and life. Twitter is also being increasingly used as a popular source of information sharing during natural disasters and mass emergencies to update and communicate the extent of the geographic phenomena, report the affected population and casualties, request or provide volunteering services and to share the status of disaster recovery process initiated by humanitarian-aid and disaster-management organizations. Recent research in this area has affirmed the potential use of such social media data for various disaster response tasks.

Even though the availability of social media data is massive, open and free, there is a significant limitation in making sense of this data because of its high volume, variety, velocity, value, variability and veracity. The current work provides a comprehensive framework of text processing and analysis performed on several thousands of tweets shared on Twitter during natural disaster events. Specifically, this work employs state-of-the-art machine learning techniques from natural language processing on tweet content to process the ginormous data generated at the time of disasters. This study shall serve as a basis to provide useful actionable information to the crises management and mitigation teams in planning and preparation of effective disaster response and to facilitate the development of future automated systems for handling crises situations.

Keywords: social media, tweet processing, sentiment analysis, text classification, disaster response, machine learning

Acknowledgments

I would like to express my sincerest thanks to my project supervisor **Dr. David Leonard** without whose guidance and valuable advice, this thesis wouldn't have come about. You are an amazing guide, an excellent teacher and a wonderful person!

I would also like to thank **Dr. Luca Longo**, M.Sc. theses coordinator, for his useful inputs in the formulation and design of research proposal.

Thank you **David Ng** and **Brendan Dier** for helping me big time with the laptop issues. Thank you **Victor Santiago** for those exciting discussions on history and European culture.

Lastly, love and best regards to **my family**, particularly my ***PAPA*** for being with me throughout my tough times and for providing me all the strength and courage to endure this journey.

I thank God for all the memories I made here in this place, bitter-sweet, they shall remain forever in my heart.

Contents

List of Figures

List of Tables

List of Acronyms

ROC	Receiver Operating Characteristic
PRC	Precision Recall Curve
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
SMO	Sequential Minimal Optimization
QAG	Quality Assurance Guidelines
IAB	Interactive Advertising Bureau
QCRI	Qatar Computing Research Institute
WEKA	Waikato Environment for Knowledge Analysis
ARFF	Attribute Relation File Format
TP	True Positive
FP	False Positive
CNN	Convolutional Neural Network
MLP	Multi Layer Perceptron
TF-IDF	Term Frequency Inverse Document Frequency
LDA	Latent Dirichlet Allocation
SVM	Support Vector Machine
NLP	Natural Language Processing
ICT	Information and Communication Technology
AIDR	Artificial Intelligence for Disaster Response
CRISP-DM	Cross Industry Standard Process for Data Mining

Chapter 1

Introduction

With a user base of more than 157 million daily active users, Twitter has become one of the most pervasive medium for social networking and micro-blogging today. Twitter is gaining popularity as a wealthy research tool for various social science and data science problems. It has successfully been used as a data source for text analytics, sentiment and opinion mining, topic modeling, text classification and summarization etc. The use of such user-generated content is no longer limited to classical social media research and analysis but also has been effectively tried and tested in new and exciting domains emerging these days, such as, disease tracking, modeling in epidemics, estimating revenues, generating insights into the personalities of customers, news analytics, polls, predicting stocks and so on. Even though the use of Twitter micro-blogging service to disperse important information during natural hazard emergencies dates back to the late 90s, the prevalence of its use for coordinating disaster response began around the year 2007 during the raging wildfires that took place near San Diego, California (Imran, Castillo, Diaz, & Vieweg, 2015; Palen & Liu, 2007). The use of Twitter as a resource for extracting useful information during hazard events is a challenging task, owing to the issues related with data quality and reliability of the posted content; it facilitates the preparation and planning of relief operations for disaster response and management. Processing of social media messages during time and safety-critical situations help to reduce the risk to human and property by accelerating casualty evacuations, providing donations and volunteering services, coordinating

medical responses and arranging well-timed supplies of food and other essentials to the affected population. Analyzing Twitter feeds during hazard events is easier and faster than other sources of information because of its real-time rapid data transmission. Crises response using social media information has turned into an active area of research over the past few years and teams involved with formal response efforts are continually incorporating that information into their processes and procedures.

1.1 Research Focus

The focus of this project is two-fold: It sets out with processing and analysis of textual content of twitter feeds collected during five different natural disaster events with an aim to generate key insights from them, specifically, in terms of sentiment polarity, subjectivity and sentiment scores, and secondly to automatically classify tweets into different categories of information useful for humanitarian organizations. While the first task of sentiment analysis is helpful in determining and assessing the sentiments of people during emergencies to obtain an understanding of their concerns, panics, and emotions regarding various issues related to the disaster, the second task of tweet classification is helpful to address specific information needs of response teams to expedite disaster mitigation.

The research project concludes with a characterization and evaluation of classification performance using various machine learning techniques to classify tweets based on the type of information they carry.

1.2 Background

There is a long history signifying the use of Internet and Web technologies to gather and disseminate disaster-relevant information during such events to facilitate stakeholders and disaster management bodies, for the planning and preparation of disaster response, dating back to the beginning of 21st century. The setting up of disaster portals and websites that bring in information from various sources have been existent for

a long time as detailed in (Palen & Liu, 2007) suggesting the role of Information and Communication Technology (ICT) in generating warnings and planning of response activities during the course of a natural disaster. This has gained serious momentum post the year 2010 with citizen participation going on the rise, during crises situations, (Sakaki, Okazaki, & Matsuo, 2010; Imran, Elbassuoni, Castillo, Diaz, & Meier, 2013; Truong, Caragea, Squicciarini, & Tapia, 2014) to publish, share, communicate, collect and spread information to aid and accelerate the response efforts. As a result, it has become a mainstream practice for the affected population and other concerned people to increasingly use social media platforms, during such times, to post textual information as well as other useful multimedia content (images and videos) to provide updates about injured, missing, found or dead people, infrastructure and utilities damage, donation needs or volunteering services requested etc. Studies reveal that this on-line information, if processed timely and effectively, is extremely useful for humanitarian organizations to gain situational awareness and plan relief operations supporting decision-making and coordinating emergency-response actions (Vieweg, Castillo, & Imran, 2014; Imran et al., 2015).

Several systems to aid disaster response during crises situations exist, for instance, Tweet4act (Chowdhury et al., 2013), Artificial Intelligence for Disaster Response (AIDR) (Imran, Castillo, Lucas, Meier, & Vieweg, 2014), Aerial Clicker AIDR (Ofli et al., 2016), SensePlace2 and SensePlace3 (MacEachren et al., 2011) etc. SensePlace2 and 3 are recent initiatives of GeoVista Lab at Pennsylvania State University which provide a support tool for geo-visual analytics and crises management. Specifically, they help to characterize and compare the space-time geography associated with topics and authors in tweets. Details about them can be found in the links:

https://www.geovista.psu.edu/SensePlace2/ and

https://www.geovista.psu.edu/SensePlace3/.

Since these tools focus more on the visual overview of place and time of tweets matching a user's query than the tweet classification itself, their scope lies outside the domain of current work, as a result, their description isn't provided in this report. An interested reader can look into the web-links provided above for a detailed understanding.

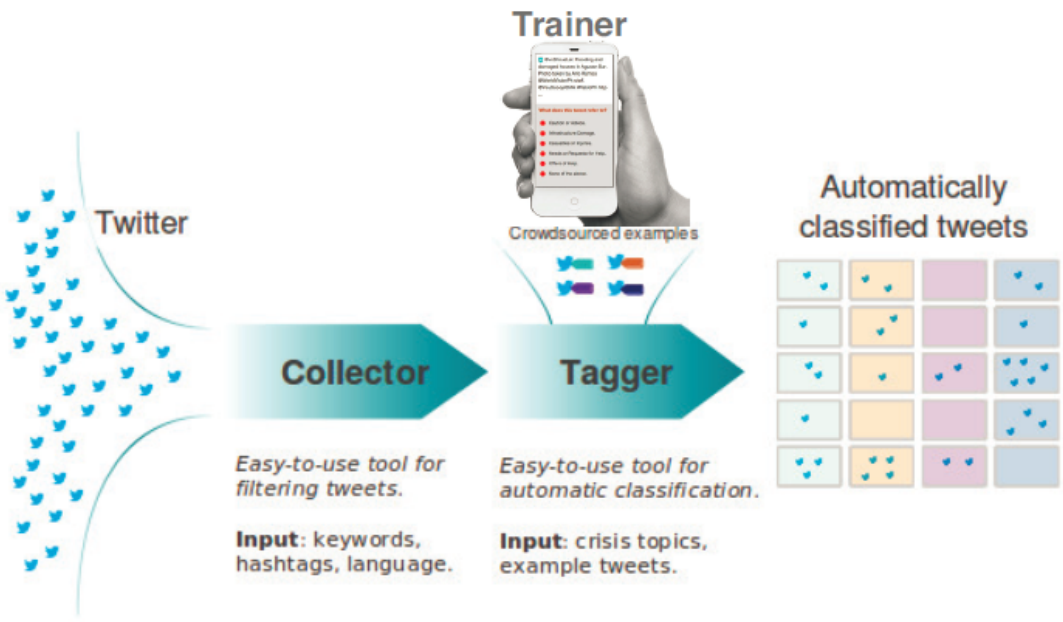


Figure 1.1: Artificial Intelligence for Disaster Response (AIDR) platform for automatic tweet classification. Reprinted with permission from (Imran et al., 2014)

The AIDR platform which is all about automatic filtering and classification of tweets during disaster events is presented here. AIDR is a free and open source service (Imran et al., 2014) designed by researchers at Qatar Computing Research Institute (QCRI), Doha to automatically filter and classify social media messages related to emergencies, disasters, and humanitarian crises. AIDR combines the best of human and machine intelligence to automatically tag up to thousands of tweets per minute. Figure 1.1 depicts the overall design of an AIDR system. Specifically, AIDR's *Collector Module* collects crisis-related messages from twitter, *Tagger Module* provides a sub-set of collected messages to a crowd-sourcing platform (like Crowd-Flower or Figure-Eight) to label them, and the *Trainer Module* trains a machine learning classifier based on the labels. The accuracy of a classifier improves as more labels become available. It has been indicated in a study by Imran (Imran, Elbassuoni, Castillo, & Diaz, 2013) that the classification accuracy of a machine learning classifier which is trained on a pre-existing dataset is not particularly high, even though there is a similarity in writing style of tweets for various disaster events. This leads to the conclusion that

crisis-specific labels offer much higher accuracy than *generic labels* obtained from past disaster events as detailed in (Imran, Elbassuoni, Castillo, & Diaz, 2013; Imran et al., 2015). Crises-specific labels obtained from tweets corresponding to different disaster events can thus be utilized effectively for disaster response actions.

The supervised learning techniques for machine classification of tweet text which are frequently being used today depend upon the availability and quality of labeled dataset. Much of the related work performed on twitter disaster datasets involve the use of standard statistical and machine learning approaches for text classification. Some of these techniques include parametric methods of *Naive Bayes, Simple Neural Network, Logistic Regression*, and non-parametric methods of *Random Forest, Support Vector Machine, Rule Induction, Decision Tree* (Imran, Castillo, Diaz, & Vieweg, 2018; Imran et al., 2015; Thangaraj & Sivakami, 2018; Chowdhury et al., 2013) and recently using *Convolutional Neural Network* (CNN) (Nguyen, Mannai, et al., 2017) for text classification. As tweet text is short, the performance of machine learning classifiers is reported as low in comparison with their longer counterparts (longer text). This is due to the fact that limited tweet length fails to provide sufficient data within the body of the target text classes. The machine learning classifiers use different schemes for text representation such as *Word Vectors, Count Vectors, Term Frequency - Inverse Document Frequency (TF-IDF), Word Embeddings, Text or Natural Language Processing (NLP)* based features or *Topic Models* as features for text classification.

Although the tasks of twitter sentiment analyses and twitter text classification have been done separately, very little to no specific information is available relating to the use and/or improvement of classifier performance including sentiment based features (obtained from sentiment analyses) for text classification. The work done in this domain till date handles the task of sentiment analyses independently of text classification. The exact nature of relationship between sentiment analyses and text classification of tweets in terms of classification performance has remained unclear, with no evidence of any documented work combining the two.

The author of the current project believes that this is the first work demonstrating

the use of sentiment based features for text classification (in a disaster scenario) with an aim to enhance the accuracy of performance. This work opens up an avenue for improving classification performance of disaster-related tweets by enriching the dataset with additional sentiment features to be used in conjunction with regular word vector features for training machine learning classifiers. The reasonable intuition to formulate the hypothesis sits on the hope that augmenting the word vector features to also include tweet sentiments could *improve* the classifier performance.

1.3 Research Problem

The main focus of this work is defined by the research question:

Does the accuracy of classifying disaster-related tweets improve by including tweet sentiments in addition to regular word vectors as features for text classification?

This question can be sub-divided into four parts which are investigated in the designed experiment:

Research Sub-Question A - What kind of *multi-dimensional textual content analyses* can be performed on disaster-related tweets?

Research Sub-Question B - Is there a difference in classification performance of disaster-related tweets using different sizes of *token lengths*?

Research Sub-Question C - Does the inclusion of *tweet sentiments* as features for classifying disaster-related tweets impact the accuracy of performance of machine learning classifiers?

Research Sub-Question D - Which text classifier performs the best in terms of highest *weighted average precision, recall* and *F1 score* for classifying disaster-related tweets?

These questions are more formally stated as an experimental hypothesis in the next section.

1.4 Research Objectives

The aim of this work is to analyze the data collected during five different natural disasters to understand and utilize various types of actionable information available on social media to facilitate disaster-response organizations. A multi-dimensional analyses of textual content of tweets then follows. This involves analyzing tweets for understanding tweet sentiments (also called *tweet polarity*), generating numeric sentiment scores, finding out the subjectivity of opinion expressed in a tweet (also called *tweet subjectivity*), extracting important *named-entities* from tweets etc. The accuracy of a machine learning classifier to correctly classify tweets into one of the predefined humanitarian categories depends not only upon the availability and quality of the labeled data but also the presence (or absence) of relevant features to be used for classification. These relevant features can be generated from a given piece of text in a number of ways, such as by counting words, term and document frequencies etc. Performing *sentiment analyses* on tweet text can also help to generate additional features, that might be useful for text classification. In this regard, a *null hypothesis* is constructed suggesting no improvement in classification performance after including additionally generated sentiment features from tweets. This is the hypothesis to be tested in this work. Expressed concisely, the aim of this research is to determine whether the use of additional features (*tweet sentiments*) improve the predictive power of machine learning models for correctly classifying tweets into one of the predefined humanitarian categories.

Null Hypothesis: Using additional features obtained from *sentiment analyses* of disaster-related tweets does not affect the accuracy of performance of the text classification task.

Alternative Hypothesis: Using additional features obtained from *sentiment analyses* of disaster-related tweets improves the accuracy of performance of the text classification task.

The research objectives corresponding to each research sub-question are as described:

Research Objective A- Perform *multi-dimensional textual content analyses* on disaster-related tweets.

Research Objective B- Observe and analyze changes in classification performance at varying *token lengths*.

Research Objective C- Measure and analyze changes in the relative performance of text classifiers using *sentiment features* in addition to regular word vectors.

Research Objective D- Compare and evaluate the performance of different text classifiers in terms of *weighted average precision, recall* and *F1 score*.

The resulting experimental tasks undertaken to achieve the research objectives are:

1. Obtain and prepare the natural disaster dataset from twitter.
2. Perform multi-dimensional textual content analyses on disaster-related tweets.
3. Generate sentiment based features from the results obtained after performing multi-dimensional textual content analyses of tweets.
4. Train and test the classification performance of different machine learning classifiers using original disaster dataset (without including sentiment features).
5. Observe the performance of different machine learning classifiers on original dataset using different token sequences.
6. Select the best performing token sequence (yielding highest accuracy) for final modeling.
7. Add additionally generated sentiment features into the original dataset (also called, dataset enhancement).
8. Train and test the classification performance of different machine learning classifiers on new (enhanced) dataset using the selected best token sequence as obtained from step 6.

9. Measure, analyze, compare and report the results of classification performance using both the datasets (original and enhanced).

1.5 Research Methodologies

The research conducted in this project is *secondary* as it relies on a dataset collected and maintained by Qatar Computing Research Institute (QCRI)'s *CrisisNLP project* on mass emergencies and disaster situations. The *CrisisNLP* team at Qatar provides resources and datasets to research communities and technologists to facilitate research on humanitarian and crisis computing by developing new computational models, innovative techniques, and systems useful for humanitarian aid. This research is *quantitative* as it deals with statistical, mathematical and numerical analysis of data using objective measurements.

The current research project involves multi-dimensional textual content analyses of tweets as well as multi-class tweet-text classification experiments using the crises dataset (with and without adding sentiment features) in an attempt to examine the impact of *feature extension* on short-text classification performance of disaster-related tweets.

As the performance accuracies of different machine learning classifiers will be compared against each other using both the datasets, the obtained results are verifiable by observation rather than purely by logic or theory. This research is *empirical* in nature as it focuses on testing the feasibility of the suggested solution using empirical evidence. This research follows a *deductive approach* as it starts with a proposed theory, progresses to a hypothesis and ends with a rejection or acceptance of the hypothesized solution.

The broad outline of Cross Industry Standard Process for Data Mining (CRISP-DM) model is followed in designing this research. In this context, CRISP-DMs *Business Understanding* phase may be considered analogous to the *Literature Review* covered in Chapter 2. The *Data Understanding* and *Data Preparation* phases of CRISP-DM are covered in Chapter 3 under *Design and Methodology*. Chapter 4 details on

Data Modeling phase discussing *Implementation and Results* of the designed experiments while *Model Evaluation* phase is covered in Chapter 5 under *Model Evaluation* and *Analysis*. Lastly, the end of the CRISP-DM cycle, *Deployment* phase corresponds to the *Discussions and Conclusions* which are outlined in Chapter 6.

1.6 Scope and Limitations

The scope of this work is strictly limited to the examination of changes in text classification performance of several state-of-the-art machine learning classifiers using both the datasets (original and enhanced). Dataset enhancement is done by adding sentiment-based features to the original dataset with the supposition that it could affect (and hopefully improve) the accuracy of classifying disaster-related tweets into one of the humanitarian categories for effective utilization of information by crises response organizations. The performance of the machine learning classifiers is evaluated in terms of *Precision, Recall, F-Score* and *% Accuracy of Correctly Classified Tweets*. *Feature engineering* (more specifically, *feature selection*) is performed by *ranking* the word vector features by the level of their informativeness (metric is called *Information Gain*) and using only the most informative features for training the text classifiers.

Additionally, different *n-gram tokenizers* are used to observe changes (if any) in the accuracy of performance. Specifically, 8 different token-levels (lengths of tokens) are tried and tested in the current experimental set-up i.e. *alphabetic, word, uni-gram, bi-gram, tri-gram, uni+bi-gram, bi+tri-gram* and finally *uni+bi+tri-grams*, in relation to the task of feature engineering, to analyze the impacts they have on the classification performance. The baseline classifier(s) is selected based on the best performing tokenizer scheme. This classifier is then expanded upon by adding sentiment features such as *sentiment scores, sentiment polarity* and *sentiment subjectivity* and is evaluated for any improvement in the classification accuracy. The same is performed using different machine learning classification algorithms and the results are compared in terms of accuracy obtained in the correct classification of tweet text. Specifically, the most commonly used state-of-the-art machine learning classifiers in this problem do-

main are exploited for the current experiment. No attempt is made to optimize or tune the classifier performance than to use additional features (*feature extension*). The use of different machine learning classifiers is undertaken to demonstrate the applicability of the findings, if any, and to rule out any effect that may arise from the use of any specific classifier. To this end, 15 different classifiers are chosen in their most basic configurations. It should be noted that the modeling is performed on labeled twitter dataset obtained from *CrisisNLP project*. The labeled data is classified into 9 different tweet categories based on the information content of each tweet. The labeling is done by paid workers and volunteers working for the crowd-sourcing platform called *Figure-Eight*, formerly known by the name *Crowd-Flower*. It is also to be noted that the multidimensional textual content analyses of the tweets is done using the available tools meant for the job in a similar or related domain, there is no way to guarantee the quality of results generated from them in the absence of a pre-labeled sentiment dataset. The accuracy of the results obtained thus greatly depend on the accuracy of the tools used to perform the task.

1.7 Document Outline

There are five chapters remaining in this report. Below is presented an outline of the content covered in each chapter ordered by the chapter number:

Chapter 2 - Review of Existing Literature: This chapter provides a comprehensive coverage of various approaches to crises analytics and disaster response planning and formulation using twitter data. It discusses the application of social media data in the field of disaster management. As this is relatively new and currently an active area of research, most work that has been done till date is provided and critically analyzed. It also summarizes the state-of-the art techniques and methods used as well as their strengths, weaknesses while also pointing at the way forward.

Chapter 3 - Experiment Design and Methodology: This chapter summarizes the project approach in terms of design, experimental set-up, methodology and systematic presentation of work-flow and information processing stages. It discusses all of the

major steps taken that form the basis of the study and their methodical execution. The project approach and design used for this work has been informed and influenced by the findings obtained after surveying existing literature. Specifically, it covers the dataset description, exploration, preparation, preprocessing and feature engineering to conduct the experiment. It also points out relevant data quality issues that can limit the performance of machine learning approaches used subsequently. Overall, this chapter focuses on design aspects of the major components of the project and how they work.

Chapter 4 - Implementation and Results: This chapter provides an in-depth explanation of the specific components of the experimentation performed. It focuses on the individual model implementation including model training, tuning and performance. Initial results are also documented and briefly discussed. More precisely, the implementation and results of sentiment analyses and tweet text classification is presented in this chapter.

Chapter 5 - Evaluation and Analysis: This chapter covers the performance testing and evaluation of the approaches used by analyzing the results of the experiments conducted. It helps to conclude that the work done has produced sound results and that the experimentation has worked as intended and to measure its performance in terms of various performance metrics, specifically precision, recall, accuracy and error reports. The model providing the best accuracy is considered and proposed to be applied to real-life disaster situations. The chapter concludes with a discussion of the strengths and limitations of the findings.

Chapter 6 - Conclusion: This chapter covers the overall achievements of the project and the weaknesses that could be expanded upon in the future. It provides a conclusion and a review of the contribution of this experiment to the literature. Suggestions are also put forward for direction of future work.

Chapter 2

Review of Existing Literature

Social media platforms such as Twitter are an active form of communication channels during mass emergencies situations like natural calamities and disaster events. Research suggests that a rapid sifting through social media messages in order to look for relevant actionable information may turn out to be tremendously useful for disaster management teams and responders to obtain valuable insights into the disaster situations as they unfold. There is an increased use of twitter during hazard events because of the speedy rate of real-time information supply. Citizen participation during a disaster event has been encouraged multiple number of times making use of publicly available data streams by responders whereby each citizen posting something about the disaster event is seen as a sensor (Sakaki et al., 2010) for detection of earthquake events in Japan.

Extracting useful information from social media messages involves a combination of intermediary information processing stages like filtering, parsing, ranking, classifying, summarizing etc. depending upon the nature of the task. Utilization of textual content from tweets poses certain challenges for information extraction and classification because of the irregular structure and form of content published on-line as well as the presence of noise. This causes a significant drop in the performance of such tasks because of misspellings, slangs, hash-tags, URLs, improper use of language and excessive use of emoticons. To this end, different state-of-the-art machine learning techniques including supervised, semi-supervised and un-supervised are being adopted.

2.1 Social Media during Crisis Situations

Social media is increasingly being used as a tool by responders providing volunteering services and offers for disaster relief and crisis management. The main applications of social media in disaster management can be summarized as: generating situational awareness and for sharing important actionable information. Situational awareness starts with the identification of aspects relating to a disaster, followed by processing of disaster situation and lastly, understanding the dynamics of interaction between the causalities and disaster-hit location. Huge amounts of time-critical and useful information that is posted during the event can be processed to reveal important insights into the event as the situation unfolds (Stowe, Paul, Palmer, Palen, & Anderson, 2016). On the other hand, information sharing enables a common person to have access to social media platform allowing them to direct and send requests of useful commodities needed to the required authorities in front of the public-eye. This accelerates disaster response and minimizes both human and property risk at the time of disaster.

It was shown by (Imran & Castillo, 2015; Imran et al., 2015, 2018) that the widespread use of Twitter during crises and emergency situations significantly improves the planning and execution of disaster management bodies enabling faster disaster response. In effect, it reduces human loss (by saving those in need) and minimizes infrastructure and utilities damage. The affected population posting useful information about missing and injured people, assistance needed in terms of food, shelter and medicine and updated reports on infrastructure damage etc. can be used by several humanitarian organizations.

In addition to analyzing textual content from social media, recent studies (Alam, Ofli, Imran, & Aupetit, 2018; Nguyen, Ofli, Imran, & Mitra, 2017; Alam, Ofli, & Imran, 2018a; Nguyen, Alam, Ofli, & Imran, 2017; Imran, Alam, Ofli, & Aupetit, 2017; Alam, Ofli, & Imran, 2018b) are continuously making use of images and other multimedia files published on them enabling crises management teams to boost disaster response significantly. Although the use of imagery for disaster response is on the hike, limited work is focused on combining both content types owing to the lack of labeled

image datasets. A recent work on multimedia content analyses of Hurricane-related tweets (Alam, Ofli, Imran, & Aupetit, 2018) targeting specific information needs of humanitarian organizations was performed using both the textual and imagery content. Sentiment analysis of tweets was performed using Stanford Sentiment Analysis classifier classifying tweets into 5 polarities ranging from very negative to very positive. Classification into predefined humanitarian categories was done using decision tree based Random Forest with an overall accuracy of 66%.

Deep learning has become another popular choice for crises response (Nguyen, Joty, Imran, Sajjad, & Mitra, 2016) using CNN, On line Learning, Word-Embeddings that are effective for sentence-level classification tasks. In order to perform classification of disaster-related tweets using word-embeddings, a skip-gram model of word2vec (Imran, Mitra, & Castillo, 2016; Lilleberg, Zhu, & Zhang, 2015) has been used recently for binary and multi-class classification with a reported accuracy of 73% and around 60% respectively. (Alam, Joty, & Imran, 2018) proposed a graph-based semi-supervised variant of CNN to classify disaster-related tweets in the absence of labeled crises datasets and showed promising results. This was done using a k-nearest neighbor similarity graph improving the performance of baseline significantly when applied to Nepal Earthquake and Queensland Flood events.

Using images for disaster response pose challenges due to inclusion of duplicated and irrelevant images. An image filtering pipeline (Nguyen, Alam, et al., 2017) using transfer learning and perceptual hashing to detect irrelevant and redundant image content for better tweet classification has been proposed.

2.2 Analyzing Sentiments from Twitter Micro-texts

Sentiment Analysis is the broad task of assigning sentiment-class labels to a given text in consideration with an aim to generate polarity of the opinion expressed by it. The text mostly derives from social media websites, blogs and product reviews etc. The task of analyzing sentiments in a given piece of text is also commonly known by the name, opinion mining, and is employed to analyze peoples sentiments, atti-

tudes and opinions about different things and entities. There is a constant upsurge in studies related to sentiment analyses due, in part, to the advancement and popularity of machine learning approaches for natural language processing, computational linguistics, information extraction and retrieval as well the ready access to massive and open-utility social media datasets, making sentiment analyses one of the most favored research domain for social media. Sentiment analysis can be broadly categorized into three main levels on the basis of their depth of operation. These are: *Document Level*, *Sentence Level* and *Entity or Aspect-Level* as mentioned in (Beigi, Hu, Maciejewski, & Liu, 2016; Pawar, Shrishrimal, & Deshmukh, 2015):

Document Level: The task at this level is classifying sentiments for the entire document. It is important to note that for this type of analysis, the documents should correspond to a single topic, multiple topics can't be accommodated in this case as this level assumes document singularity for its operation.

Sentence Level: This provides a detailed sentence-level analysis for each line in the document. Each sentence is evaluated to determine the polarity of opinion expressed by it ranging from negative to positive. Neutral class may or may not be included for a sentence.

Entity or Aspect Level: Aspect level or entity level deals with each entity that a sentence talks about. It can be thought of as contextual sentiment analyses as it needs to have an understanding of how many entities a sentence has and what kind of sentiment words (adjectives or adverbs to denote their quality) are being used. A single sentence might have two totally unrelated entities with opposing opinions. As an example, consider the sentence: "This book is brilliant but is too lengthy to read". There are two aspects in this case with differing sentiment polarities. Aspect level sentiment analyses is more detailed in approach and thus can be highly reflective of the sentiment expression but is complicated and can vary significantly across domains. Again, the sentiment word "frightening" will be positive for a movie review (horror genre) but when used in context of a product review, say, a car, it totally changes the connotation and meaning. Thus, domain adaptability is one of the main limitations of this finer level sentiment analysis approach.

Sentiment analysis can be performed in a number of ways depending upon the domain, type and nature of text and possible applications. In a review article by (Beigi et al., 2016), sentiment analysis is classified into two groups - language processing based sentiment analysis and application-oriented sentiment analysis.

Language Processing Based Sentiment Analysis - This group includes sentiment dictionaries (also called lexicons) to perform the sentiment analysis. It makes use of grammar constructs and rules of language words and semantics to properly classify a sentence into a positive or a negative class. Lexicons can be generated based on a language dictionary or a domain-specific corpus. Dictionary-based approaches are more comprehensive and exhaustive as they involve bootstrapping while corpus-based approach is a bit restrictive and non-transferable to other domain areas. Sentiment lexicons are known to improve the performance of polarity and subjectivity classification for sentences in a given text.

Application-Oriented Sentiment Analysis - This group deals with the application area where the sentiment analysis is applied. Due to the massive available of online information from social media, several application-oriented sentiment analysis tasks have been performed including classifying movie and product reviews, App reviews, for predicting stock market and customer trends on the basis of their likes and dislikes of certain items. A wide range of tools are available which perform application-oriented sentiment analyses while machine learning techniques like SVM, Naive Bayes, Maximum Entropy etc. are equivalently popular choices.

2.2.1 Sentiment analyses of Twitter Data using Different Techniques

Sentiment Analysis on Twitter Data (Dattu & Gore, 2015) is done using three main techniques: Lexical analysis, Machine learning based analysis and Hybrid/Combined analysis. These are described here briefly before we go ahead with the sentiment that we performed on our dataset.

Lexical analysis: This technique uses a dictionary of pre-tagged lexicons. The

dictionary can vary across different applications. The working principle is simple: Take the input text and break it down into tokens using a certain token sequence (word-level, uni-gram, bi-gram etc.) and match every token with the contents of the dictionary. If there is a match found, then score the token with a corresponding value of that sentiment word, else generate no score for a given token. Similarly, one can have a polarity based lexical analysis, instead of calculating the sentiment scores, this approach only looks for a match of a token into either of the two classes - positive word list and negative word list and classifies the incoming token sequence on the basis of the number of matches found in the text. This surprisingly simple approach does produce good quality sentiment classification results. This is one of the earliest approaches to sentiment classification and reaches an accuracy of as much as 80% on single phrases using adjectives.

Machine learning based analysis: As machine learning gets incorporated in everything, there is a lot of interest in using the state of the art machine classification approaches to sentiment analysis in twitter (Psomakelis, Tserpes, Anagnostopoulos, & Varvarigou, 2014). The main reason this technique is favored is because of its domain-adaptability and high level of accuracy. In case of labeled sentiment datasets, the supervised machine learning classifiers are one of the choicest methods to perform sentiment analysis. It is possible to use uni-grams, bi-grams and tr-gram sequences as feature vectors corresponding to single word, two consecutive and three consecutive word phrases respectively. Higher order n-grams are useful in cases where more adjectives or adverbs are expected in a dataset. Also, the significance of bi-grams increases in case of negations and indirect word references. Example, if using a unigram, the sentence 'This is not good' might be classified as positive because of the word 'Good', however, using bigrams, 'not good' is classified as negative sentiment. Feature selection is performed before word features are fed as input into the classifiers. Lists of equally sized positive and negative words (Kumar & Sebastian, 2012) are supplied are input features to maximize performance. Most common supervised techniques employed for sentiment classification include Support Vector Machine, Naive Bayes, Random Forest, Maximum Entropy Classifier (Jaderberg, 2016; Wakade, Shekar, Liszka, & Chan,

2012). An accuracy ranging from 60% to 80% is observed for classification using these supervised techniques. The main challenges in designing a classifier in this case depend on the availability of training data, contextual understanding of the word phrase and its surroundings as well as the size of the data corpus. Owing to limited availability of pre-labeled dataset in some cases, there is a growing interest in using distant supervision (Go, Bhayani, & Huang, 2010) as well to improve the classification accuracy of sentiment analysis (da Silva, Hruschka, & Hruschka, 2014), ensemble approaches have been proposed.

Hybrid analysis: Lastly, hybrid approaches (Asghar, Kundi, Ahmad, Khan, & Khan, 2018) which bring the best of both the previous approaches - lexical analysis and machine learning are used to enhance the capabilities of the classifiers. These have high accuracy as well as faster speed. Any base classifier such as Naive Bayes, Random Forest, SVM can be coupled with a lexical component to build the hybrid scheme of sentiment analysis. Several algorithmic approaches have been tried and tested in Twitter to conduct sentiment analyses. A study on comparison of algorithms for twitter sentiment analyses (Whipple, 2017) suggest that weighted combination of predictive models yield a higher accuracy than any one method alone.

2.2.2 Enhancing General Sentiment Lexicons and Semantic Features for Domain-Specific Use

Word Lexicon refers to a list of words used in a particular language or subject. To enhance the capabilities of sentiment analysis, various domain specific lexicons have been developed from time to time. A sentiment lexicon is essentially a combination of sentiment words and phrases (idioms) characterized by sentiment polarity, positive or negative, and by sentimental strength. A sentiment lexicon is developed by selecting words and assigning scores to the words, and the performance of sentiment analysis depends on the quality of the assigned scores.

Building domain specific sentiment lexicons combining information from many sentiment lexicons and a domain specific corpus Hugo Hammer (Hammer, Yazidi, Bai, &

Engelstad, 2015) emphasized the fact that most appropriate score assigned to a word in the lexicon is dependent on the domain. In this paper, the author developed a method to construct domain specific sentiment lexicons by combining the information from many pre-existing sentiment lexicons with an unannotated corpus in the domain of interest. Results show that the best sentiment lexicon is the one that is constructed by combining the information from both the source sentiment lexicons and the product review corpus.

To build a domain specific lexicon (Labille, Gauch, & Alfarhood, 2017), author used Amazon product reviews for 15 different categories. Two generic lexicons of SentiWordNet and Generic-Spec were compared against the constructed domain-specific lexicon and it was observed that domain-specific lexicons outperform both the generic lexicons with an average accuracy of 90.09% in their appropriate domain. Likewise, domain-specific lexicons average an F1-Score of 0.94 against 0.87 and 0.91 for both generic lexicon.

In relation to crises-related scenario, a disaster lexicon (Olteanu, Castillo, Diaz, & Vieweg, 2014) called crisis lex was created by using frequently used words during crisis to automatically identify new terms to describe the crisis event. This lexicon showed an overall improvement in recall when added to a set of manually chosen key words enhancing the capabilities of a general sentiment lexicon for crises situations. Similarly, Sentpro was used by (Kreutz & Daelemans, 2018) to enhance DuOMan (a general purpose lexicon) for domain specific use.

Performance of enhanced lexicon is increased when used in a in-domain classification task and performance worsens when used in an out-domain setting. This shows that adaptation to other domains is not possible as expected. Web directories (Minocha, 2012) have also been used to generate a sentiment lexicon for a specified domain type using sentiment scores on top of an ontological structure. Again, a better performance in comparison to a general purpose lexicon was observed.

2.3 Sentiment Analyses of Social Media Data in Disaster Relief

Social media has pervasively played an important role in providing individuals and communities with warnings about evacuations, volunteering services, humanitarian aid and fund-raising during disaster events. It is a common practice for people to post their experiences, ideas, needs and opinions regarding an event (incident) in the form of text, images, videos etc. to generate situational awareness, request and present donation needs, locate, help and support those in need. Sentiment analysis of disaster related tweets is reflective of the emotional states, feelings, panics and concerns (Beigi et al., 2016; Alam, Ofli, Imran, & Aupetit, 2018) of the affected population and of those concerned to improve decision making of humanitarian organizations during mass emergencies. Current social media visualizations at the time of disasters (MacEachren et al., 2011; W. Wang & Stewart, 2015; W. Wang, 2014) focus only on spatial and temporal aspects of the geographical phenomena with no consideration to include sentiments. Sentiment information when combined with visual analytic methods could communicate real-time situation during hazards in a more readable and interpretable way.

Most common methods to analyze sentiments during a disaster event employ machine learning techniques using SVM, Naive Bayes, Maximum Entropy, Random Forest, Swarm Intelligence etc. Both polarity and subjectivity of tweets can be extracted as linguistic features to analyze the evolution of a social sentiment. SentiWordNet and AFINN packages have been extensively used on datasets gathered from social media posts (Beigi et al., 2016) to perform such an analysis. The different packages and tools can process emotions into different categories. For instance, Senti-Strength can classify tweets into positive and negative on the basis of calculated sentiment scores while Sentiment Treebank (Socher et al., 2014) provides a five class classification of sentiments into *Very Negative, Negative, Neutral, Positive* and *Very Positive*. Specifically, user-defined classes expressing anger, happiness, sadness, surprise, disgust, fear and other psychological states can also be added to the list for richness of emotional

expression.

One of the major limitations in this area relate to the absence of domain-specific sentiment labels as they are extremely hard to generate, more so for a data of this staggering size as obtained from social media. Various unsupervised and semi-supervised approaches are being utilized currently to address this problem (Alfarrarjeh, Agrawal, Kim, & Shahabi, 2017; Alam, Ofli, Imran, & Aupetit, 2018) but these studies are still not mature enough and need further research. Another area that needs to be investigated is advancing domain specific sentiment-lexicons so that they can be leveraged with classifiers to improve their classification and augmenting the results with geo-spatial visualization to facilitate crises response in real-time.

2.4 Short-text Classification in Twitter

To classify a piece of text into binary or multi-class labels requires a machine learning algorithm to understand the document. There are various ways to represent a document, the simplest of which is a bag of words approach, which simply tokenizes each word of the written text and uses them as features for text classification. Another most commonly used way is to weigh the features using a term-frequency inverse document frequency (TF-IDF) score such that higher value of TF could infer a higher feature weight. Machine learning classifiers can be applied to these representation schemes to perform text classification.

The rising popularity of on-line short message communication using SMS, Twitter and other social media platforms however, do not work well with traditional text representations because of the reduced text length thereby causing way too small word occurrence in a document to offer any meaningful context. Most short-text classification tasks therefore rely heavily on using web searches, Wikipedia and WordNets (Sriram, Fuhry, Demir, Ferhatosmanoglu, & Demirbas, 2010; Li, He, & Ma, 2017) to enhance their semantic knowledge.

2.4.1 Feature Engineering for Text Classification

Feature Engineering refers to data pre-processing steps with an aim to achieve dimensionality reduction. This is useful because all the attributes/variables are not equally relevant for predictive modeling, some of the variables offer most predictive capabilities to a data model while other correlated or redundant variables simply enhance the data-size without adding anything to the prediction, such variables need to be eliminated so that model complexity as well training time can be minimized thus enhancing performance. Feature engineering is most often done using feature selection methods, particularly suited for such tasks as classification, clustering and regression.These techniques find extensive usage in text mining, computer vision, industries, bio-informatics and other application domains working with big data.

Principal Component Analysis, Linear Discriminant Analysis and *Multidimensional Scaling* are some of the most useful feature extraction methods which differ from feature selection methods as the former involves transformation of original variables into a new feature set while the latter is simply a process of picking up a subset of features without any transformation (Jovic, Brkic, & Bogunovic, 2015; Chandrashekar & Sahin, 2014). Among many ways to represent natural language text as features for text classification, bag-of-words have been one of the earliest and most commonly used approaches. Other approaches to capture syntactic and semantic relationships between words include phrases, synonyms and hypernyms of word forms. Early work in this area (Scott & Matwin, 2001) performed on *Reuters-21578* dataset using RIPPER algorithm reported very slight improvement in classification performance as compared to bag-of-words approach.

A number of techniques are used to perform feature selection including use of standard filters, wrappers, embedded methods as well as hybrid approaches, to name a few. Common filters used are Information Gain, Correlation, Chi-squared. Wrapper methods differ from filter approaches in the way they generate a subset of data based on classifier performance, example Naive Bayes, SVM etc. for text classification (Jovic et al., 2015) while embedded approaches rely on feature selection during model execution embedded in the algorithm itself, they have been used along with logistic regression,

random forest and their variants. Hybrid methods, on the other hand, offer the best capabilities from both filter based and wrapper approaches.

While common feature selection strategies of *Subspace and Uniform Sampling, Document Frequency* and *Information Gain* are popular for text classification, (Dasgupta, Drineas, Harb, Josifovski, & Mahoney, 2007) proposed an SRLS Algorithm to perform feature selection on three datasets of *Tech-TC, NewsgroupS* and *Reuters-RCV2* where the best performing feature selection method of Information Gain produced almost comparable results using SRLS on the same datasets. (Rogati & Yang, 2002) in a similar work reports high performance using chi-squared filter as opposed to using information gain for certain types of datasets using different classifiers such as K-Nearest Neighbors, Naive Bayes, Rocchio and Support Vector Machines (SVM). Feature Selection methods applied in conjunction with supervised and unsupervised classification learning (Garnes, 2009) using *Naive Bayes* and *SVM* showed high quality results with *Chi-squared, Information Gain* and *Mutual Information* for supervised while *Term Frequency Inverse Document Frequency (TF-IDF)* and *Collection Frequency* showed best results for unsupervised learning respectively.

2.4.2 State-of-the-art Approaches to Twitter Text Classification

Twitter is being increasingly used for short-text classification and categorization into a set of pre-defined labels or topics. Most common techniques for such reduced length text classification include the supervised machine learning techniques of Support Vector Machines, Random Forest, Naive Bayes and Logistic Regression etc. A lot of recent work has started to look at shallow and deep learning neural networks for tweet text classification with very good classification performance (Choudhary & Sain, 2016; Gharavi & Bijari, 2017). Message classification of tweets into six of the commonly used topical categories of *Sports, Politics, Entertainment, Education, Technology* and *Business* have been performed using deep neural network (Sahoo, 2017) producing an accuracy as high as 80%. Similar work has been reported by (Sriram et al., 2010)

using Naive Bayes, Sequential Minimal Optimization and C4.5 for classifying tweets into user-defined classes of *News, Opinions, Deals, Events* and *Private messages* on a tweaked feature set (with additional features).

While the most common form of text representation for tweets is to use PCA, word embeddings and Wikipedia-trained word2vec models to provide context and understanding necessary for short text classification, recent approaches are focusing on using dense representations using topic models (Li et al., 2017), specifically Latent Dirichlet Allocation (LDA) and their integration with pre-trained word2vec models used on a supervised text classifier like SVM. Cascading the topic features with word vectors improves the semantic representation of the feature set resulting in improved classification accuracy.

2.5 Gaps of Literature

Even though twitter has been used extensively during disaster events for information sharing and generating situational awareness, thereby improving the capabilities of disaster response, the classification of tweet text has been performed separately from sentiment analyses. While the tweet text classification uses various machine learning classifiers to categorize tweets into pre-defined tweet labels, all the work (to the author's awareness) has only used tweet content as features for text classification with no inclusion of sentiment based features. Sentiment analyses on crises-related tweets is another domain which works in complete isolation of tweet text classification. The use of sentiment analyses during crises to raise awareness about people's panics and concerns has been attempted as a related task, but integrating the two has not been done yet. This work aims to perform sentiment analyses on disaster related tweets and use its output (sentiment scores, polarity, subjectivity) as additional features for text classification with a hope that it will improve the classification performance.

Chapter 3

Experiment Design and Methodology

This chapter discusses the underlying project approach and detailed design aspects of the experiments conducted as a part of this study. This also includes the statistical treatments of the experimental results produced. An overview of the experimental design, specifications of hardware and software used and documentation of the data source and contents is also provided.

3.1 Project Approach

The aim of the current research is grounded in measuring the classification performance of twitter disaster dataset using sentiment features (generated as a result of textual content analyses of tweets) which is described in Section 3.4 of the current chapter. The dataset with additional features, on which to perform the text classification, will henceforth be referred to as *Enhanced Dataset* for the remainder of the thesis. Several state-of-the-art machine learning text classification techniques are applied to this *Enhanced Dataset*. The overall project can be sub-divided into two broad tasks: *Multidimensional Textual Content Analyses Of Disaster-Related Tweets* and *Classifying Disaster-Related Tweets Using Original And Enhanced Dataset With Additional Features Using Several Machine Learning Classifiers*.

The differences, if any, in classification performance using *Enhanced Dataset* with additional features, as measured by weighted average Precision, Recall and F1 score for accuracy of classification, will be analyzed to determine if their impact on the classification performance is statistically significant or not. Specifically, the aim is to answer the four research sub-questions as presented in Chapter 1:

- What kind of *multi-dimensional textual content analyses* can be performed on disaster-related tweets?
- Is there a difference in classification performance of disaster-related tweets using different sizes of *token lengths*?
- Does the inclusion of *tweet sentiments* as features for classifying disaster-related tweets impact the accuracy of performance of machine learning classifiers?
- Which text classifier performs the best in terms of highest *weighted average precision, recall* and *F1 score* for classifying disaster-related tweets?

3.2 Design Aspects

The overall system can be viewed as two-entity process decomposed into *Textual Content Analyses* and *Tweet Text Classification*. These two entities are covered in detail in Section 3.3 under the header *Detailed Design and Methodology* where each of these components are in turn viewed as a system in their own right and decomposed further.

Figure 3.1 presents a broad work-flow digram modeling the interactions between the main components of the experimental set-up and an accompanying explanation of the overall design at a high level. This experimentation was undertaken using a Lenovo Laptop with an Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz, 4 Logical Processor(s), Intel(R) HD 4000 graphics card and 8 gigabytes of on-board RAM.

Raw tweets are pre-processed and cleaned using R programming language. This includes removing stop-words, URL's, hash-tags, twitter handles etc. The cleaned up tweets are then used to perform two tasks - *Textual Content Analyses* and *Tweet Text*

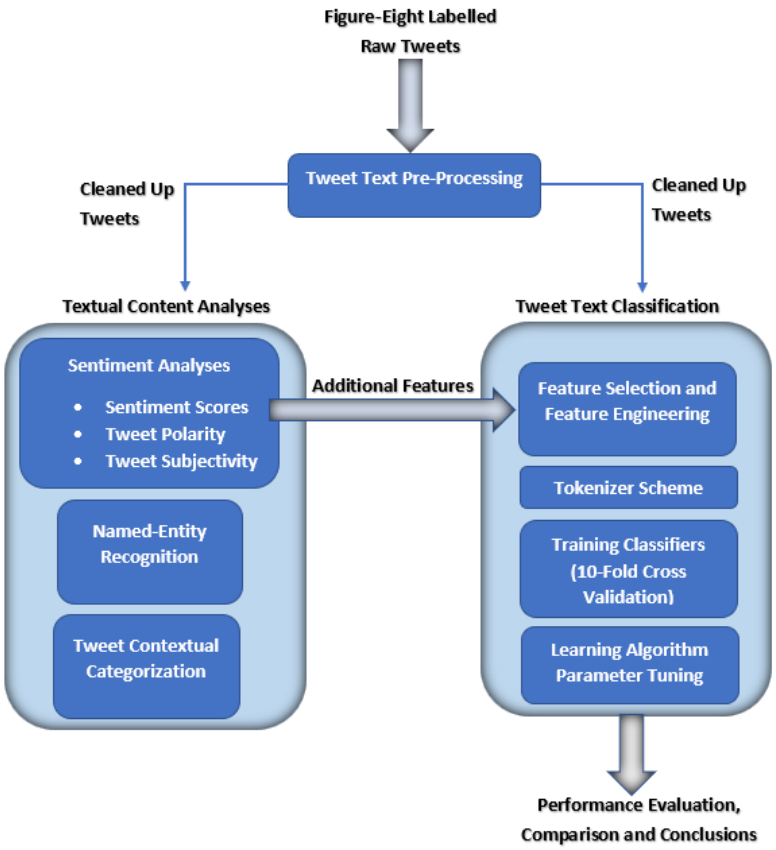


Figure 3.1: Broad methodology for the experiment.

Classification. Textual content analyses is a multi-dimensional analyses of textual content of the cleaned up tweets, which includes sentiment analyses, named-entity recognition and contextual categorization of tweets. Tweet text classification, on the other hand, is the automatic categorization of tweets into one of the nine pre-labeled categories with the help of several state-of-the-art machine learning classifiers.

10-fold cross validation is used to get a realistic picture of the modeling results in terms of their generalizability. It is important to understand that the output of textual content analyses (specifically, *sentiment analyses*) is used as an input into the features used for text classification to compare and evaluate the performance of the different classifiers. Section 3.3 covers the details of each process.

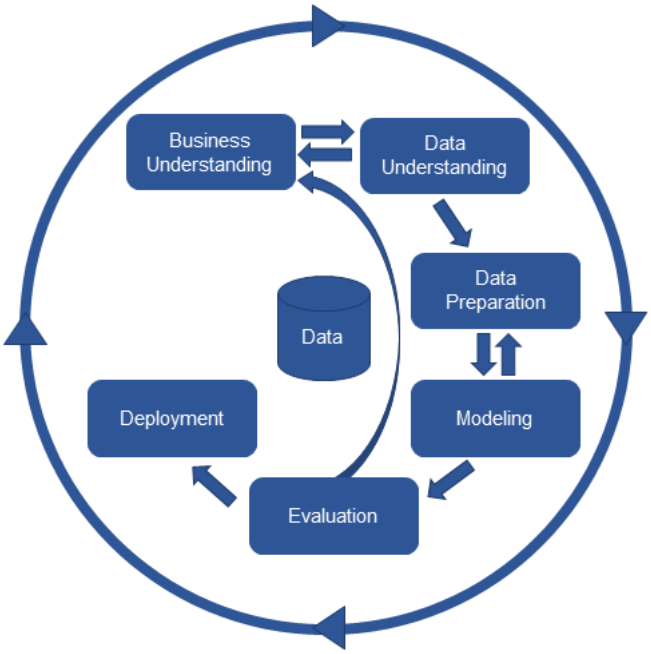


Figure 3.2: CRISP-DM Process Model for Data Mining.

3.3 Detailed Design and Methodology

This section provides a detailed methodology based on the CRISP-DM (Cross Industry Standard Process for Data Mining) process model as shown in Figure 3.2. The CRISP-DM process model provides a structured approach to planning and designing a data mining project as well as organizing the experimental set-up.

Most of what is covered in chapters 1 and 2 (Introduction and Literature Survey respectively) account for the business understanding part. That involves understanding the research objectives and requirements from a business perspective which includes steps like refining the research objectives into a specific data mining problem definition and specifying the data mining goals and success criteria. The focus of the current chapter, however, is on devising a preliminary plan to achieve the objectives by outlining a step-by-step action plan for the project as well as initial assessment of the tools and techniques. This is done after reviewing the available data, also called *Data Understanding*. This involves gathering data, describing and exploring it and most importantly, verifying the data quality. As the raw data obtained from an on-line source is generally not suitable to be used directly in analytics and machine learning

applications, it needs to be cleaned, pre-processed, profiled, validated, transformed, formatted, organized and prepared. All this comes under the broad range of things called *Data Preparation* as discussed in this chapter. This then leads to the design and development of analytic models, *Data Modeling* step which includes selecting appropriate modeling techniques, configuring and setting up of modeling parameters, designing tests, building models and assessing them. This step is followed by *Evaluation* which involves evaluating results, reviewing the processes and findings, highlighting any concerns that require immediate attention or those steps that were overlooked or that should be revisited, along with determining the next steps. This concludes by reviewing and reporting final results and outputting the deliverable, also called *Deployment*. The *Data Modeling, Evaluation* and *Deployment* stages are covered in Chapters 4, 5 and 6 respectively of this report.

3.3.1 Multi-Dimensional Textual Content Analyses Of Tweets

Figure 3.3 describes the detailed methodology for performing multi-dimensional textual content analyses of disaster-related tweets. The labeled tweets from *Figure-Eight* need to be pre-processed before any type of sentiment analyses or entity extraction or categorization takes place. R Studio has been used to clean up the raw tweets by removing the hash tags, URL's, stop-words etc. The details of tweet pre-processing is elaborated in Section 3.4 of the current chapter.

Extracting named entities from tweets is useful because it can help to rapidly assess the disaster situation by providing detailed information about the names of people, locations and organizations in the tweets enabling crises management teams to quickly go through thousands of tweets while discarding unnecessary and irrelevant tweets. It is very common to find completely off-topic tweets, having to go through them individually is a big wastage of time and resources. Entity extraction thus, saves time and helps to identify most necessary and frequent people, organizations and locations from the data. In this work, named entity recognition is done using a text analysis extension of *Rapid Miner* software known by the name *'Rosette'*. Not only is extracting important named entities from tweets significant

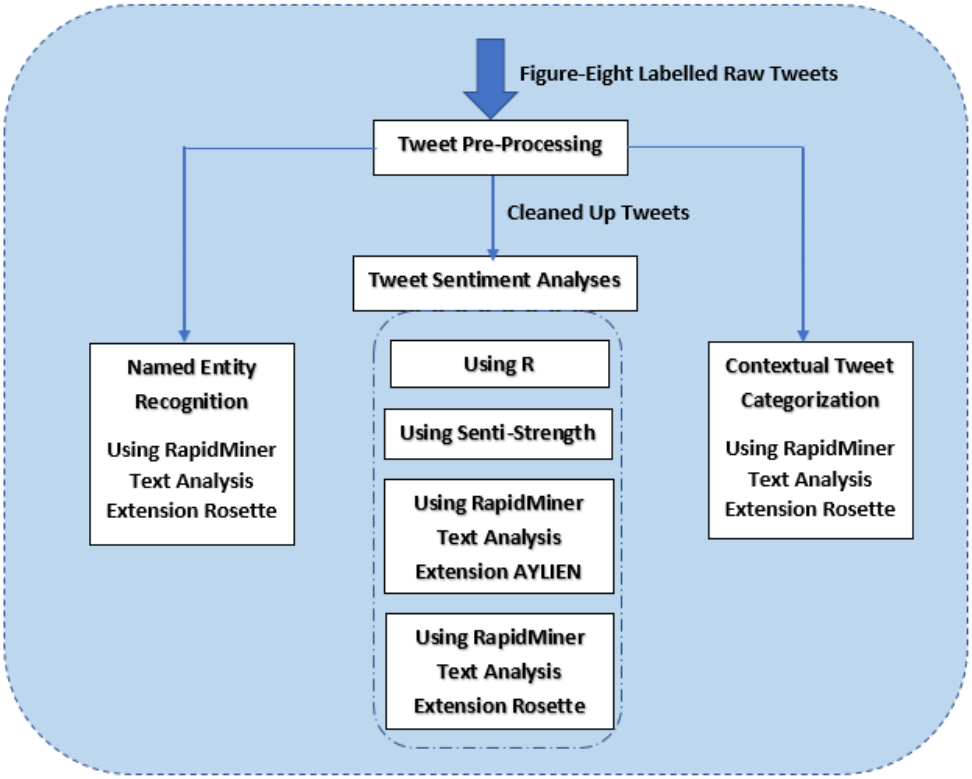


Figure 3.3: Methodology for textual content analyses of disaster-related tweets.

but also is to learn about the concerns and panics of people, commonly expressed through emotions and sentiments at the time of disaster. It helps responders establish stronger situational awareness of the disaster area. To this end, *sentiment analyses* is performed using a variety of techniques (R, Senti-Strength, Rapid-Miner AYLIEN, Rapid-Miner Rosette) to obtain scores like sentiment strength, sentiment polarity and sentiment subjectivity etc. providing useful insights about the crises situation. Specifically, tweet categories which correspond to anger, frustration, fear and negative sentiments are analyzed. This helps humanitarian organizations to keep an eye on public sentiment and issues affecting people and to plan the disaster recovery process in a timely manner. Lastly, *contextual categorization* of tweets based on Interactive Advertising Bureau (IAB) Quality Assurance Guidelines (QAG) taxonomy is conducted. The IAB provides a list of several Tier 1 and Tier 2 categories based on the context of data. These categories help to classify tweets on the basis of topics of discussion in a particular tweet. Tier 1 is a relatively broad categorization of tweet

topics whereas Tier 2 is highly specific categorization. More details about this can be found in the link: https://www.iab.com/guidelines/iab-quality-assurance-guidelines-qag-taxonomy/iab-quality-assurance-guidelines-qag-taxonomy/. This is also performed using *Rapid Miner* Text Analysis Extension *'Rosette'*.

3.3.2 Tweet Text Classification

As the information needs of different humanitarian organizations vary in accordance with their responsibilities in the disaster response and the level at which they operate - local, regional, national etc., it is important to provide them with the relevant actionable information and not junk. Some humanitarian organizations need a high-level overview information about the crises, i.e. the scale of the disaster, urgent needs of the affected people, infrastructure and utilities damage, economic issues while others such as police forces, fire-fighters, municipalities etc. seek information related to immediate individual emergencies such as reports of missing, trapped or found people in need of food, shelter and medical supplies etc (Alam, Ofli, Imran, & Aupetit, 2018). Therefore, only specific type of requisite information aligned with the specific response priorities should be provided. This can be done by automatically classifying the incoming tweets into different information categories. The pre-defined information categories based on the tweet text are covered in Section 3.4 of the current Chapter.

Figure 3.4 represents the detailed methodology for performing multi-class tweet text categorization. The cleaned up tweets generated with the help of R are used to perform the task of text classification. The automatic classification of tweets into one of the different pre-defined information categories is performed using several state-of-the-art supervised machine learning approaches. As can be seen from the detailed methodology, the word vectors that are generated after text transformation, stemming and tokenization, are used to perform the text classification. All the word vector features are not important for the classification of tweets into different humanitarian categories, there are a few relevant features which provide the best splits for tweet classification, they should be utilized while others should be discarded. It is therefore useful to rank the word vectors in terms of their informativeness (*information gain*)

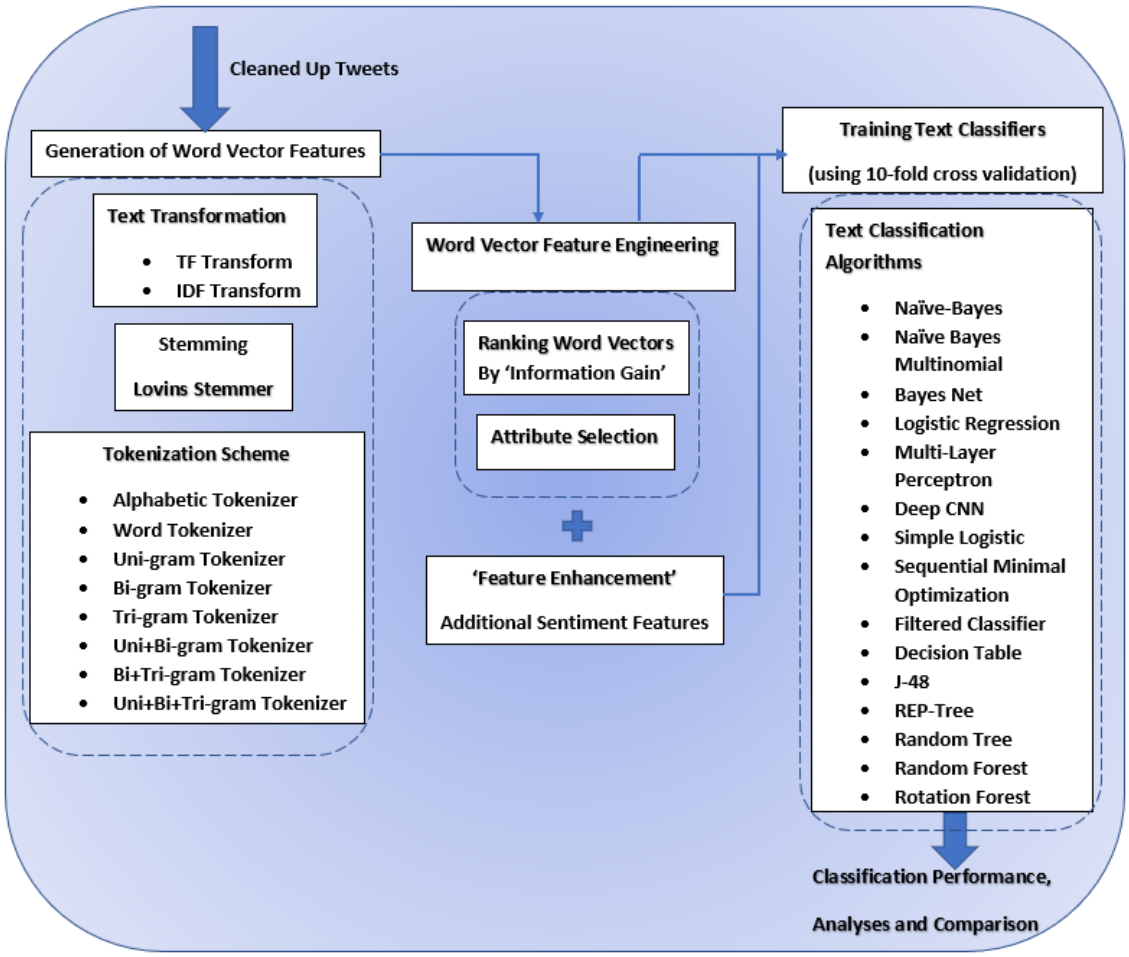


Figure 3.4: Methodology for automatic classification of tweet text.

for training the machine learning classifiers. In this work, the best (most informative) word vector features are selected and later enhanced by adding sentiment-based features as described in the previous section. The classifiers are then run, evaluated and compared against each other as well with the results obtained without using sentiment based features.

3.4 Data Description

Although large amounts of twitter datasets are freely available on-line from various sources, it was observed that the datasets on crises and disaster situations were very few. Specifically, two websites were found to be dedicated for the development,

maintenance and upkeep of disaster related datasets extracted from twitter: *Crisis Lex* and *Crisis NLP* which can be found in the web-links: http://crisislex.org/ and http://crisisnlp.qcri.org/ respectively. Both of these resources are a repository of crisis-related social media data and tools. While Crisis Lex includes collections of crisis data (in different languages) and a lexicon of crisis terms, it was initially released in the year 2014 with the idea of collecting and filtering micro-blogged communications related to crises events. It also includes tools to help create crises lexicons and data collections.

A *lexicon* essentially refers to a catalog of language words which includes *morphemes* (morphological unit of a language which may or may not stand alone as words) in computational linguistics. As most of the domain-specific collections of written texts, also called *corpus* (plural is corpora) contains a specific set of lexicon terms which are domain-specific and not generic, the Crisis Lex team produced a list of such *Emergency Management* (EM) terms (Temnikova & Castillo, 2015) containing up-to 7,000 word descriptors used in Twitter to describe various crises events. This resource has been used by practitioners to search for relevant messages in Twitter during crises, and by computer scientists to develop new automatic methods for crises handling. The Crises Lex is a rich source of manually labeled disaster-related tweets (Olteanu et al., 2014) into one of the many predefined information categories useful for humanitarian organizations. There is a total of 26 different disaster events in this dataset in various languages. This multi-language dataset was not used in the current project as it would have unnecessarily complicated the sentiment analyses as well as the classification task.

The dataset for this project was taken from the *Crisis NLP* project (Imran et al., 2016) available from the link: http://crisisnlp.qcri.org/lrec2016/lrec2016.html. This is a humanitarian organization at Qatar Computing Research Institute (QCRI), Doha which has been collecting and generating meaningful twitter corpora corresponding to major natural hazard events happening world over since the year 2011. The crisis computing team at QCRI collects data for research on humanitarian and crises computing. This is a publicly-available dataset collected using domain-specific application programming interfaces (APIs) and consists of *English-only tweets*. The Crisis NLP

group at QCRI provides resources for research on *Crisis Informatics* to help researchers and technologists in developing new computational models, innovative techniques, and systems useful for humanitarian aid.

The Crisis NLP *Crowd-Flower* labeled twitter dataset consists of several thousands of manually annotated tweets collected during major natural disasters including earthquakes, hurricanes, floods, typhoons and cyclones that happened between the years 2013 and 2015 around the world. A team of paid workers and volunteers from the *Crowd-Flower* (now called *Figure-Eight*) crowd-sourcing platform were used to perform the labeling of tweets (Imran et al., 2016) into one of the 9 pre-defined humanitarian categories catering to different information needs of the response organizations. At least three different workers were required to agree on a label before a task was finalized and no worker was allowed to perform more than 200 labeling tasks. A tweet was categorized solely on the basis of the tweet content, no attention was paid to the URL's in the tweets. A total of 9 categories were used in this task, as described:

1. Injured or dead people—Reports of casualties, fatalities and/or injured people.

2. Missing, trapped, or found people—Reports and/or questions about missing or found people.

3. Displaced people and evacuations—People who have relocated due to the crisis, even for a short time (includes evacuations).

4. Infrastructure and utilities damage—Reports of damaged buildings, roads, bridges, or utilities/services interrupted or restored (e.g. power lines, water pipes etc.).

5. Donation needs or offers or volunteering services—Reports of urgent needs or donations of shelter and/or supplies such as food, water, clothing, money, medical supplies or blood; and volunteering services.

6. Caution and advice—Reports of warnings issued or lifted, guidance and tips, cautions, and advice about the disaster useful for other vulnerable people or humanitarian organizations.

7. Sympathy and emotional support—Prayers, thoughts, and emotional support towards the victims of the disaster.

8. Other useful information—Other useful information that helps understand the situation and can be potentially important for the humanitarian organizations.

9. Not related or irrelevant—Unrelated to the situation or irrelevant for humanitarian response.

3.5 Data Exploration

There were individual CSV files corresponding to a single disaster event which were joined together to perform the initial exploratory analyses. Each CSV file contained roughly around 2,000 tweets pertaining to a single disaster event. The initial data exploration was done using *Tableau* software. The cumulative file generated after joining all the individual CSV files contained exactly 19,112 tweets in English language. The original dataset also consisted of Middle-East Respiratory Syndrome (MERS) and EBOLA diseases as disaster events. However, they were eliminated in the current exercise as they were categorized into different humanitarian categories (not among the 9 categories defined above), this would have impacted the classifier model and its accuracy, if included. In this work, tweets corresponding to five different types of natural hazard events: earthquake, flood, hurricane, cyclone and typhoon are considered.

Figure 3.5 represents the distribution of tweets by different disaster events. It can be observed that a total of around 9,000 tweets were collected for earthquake events, this is the case because there were 4 earthquake events in the dataset - US Earthquake 2014, Pakistan Earthquake 2013, Chile Earthquake 2014 and Nepal Earthquake 2015 respectively. Again, a total of around 4,000 tweets is seen for flood events because it included India and Pakistan Floods for the year 2014 respectively. The remaining tweets were for Cyclone Pam in Vanuatu 2015, Typhoon Hagupit in Philippines 2014 and Hurricane Odile in Mexico 2014. It can be observed that there is an equitable number of tweets per disaster event in the dataset, this is helpful when training the classifier models for automatic categorization of tweets coming from different disaster events.

Figure 3.6 depicts the distribution of tweets across the nine predefined humanitar-

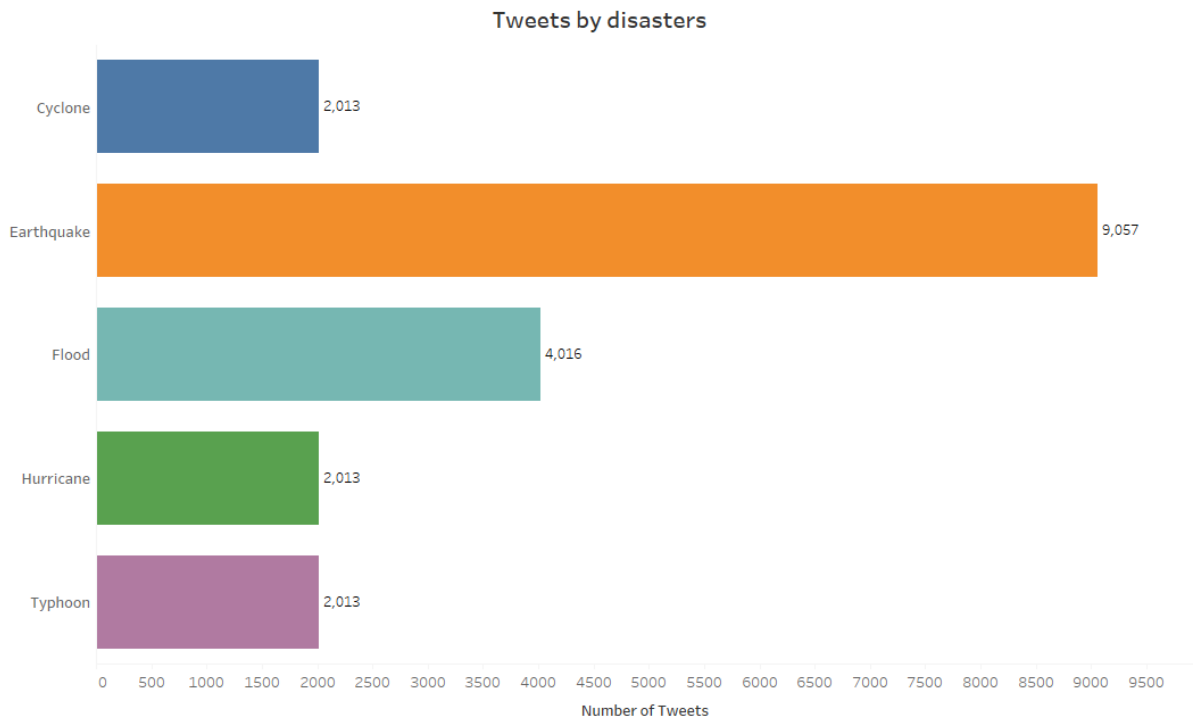


Figure 3.5: Number of tweets in each disaster type.

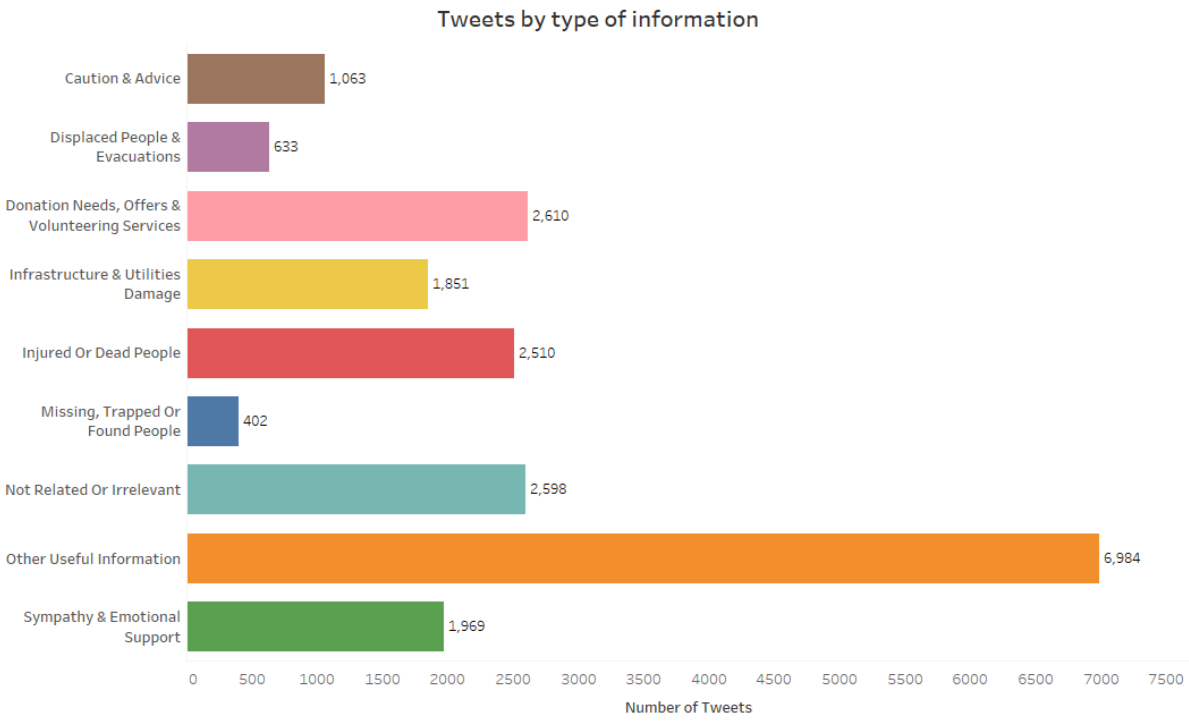


Figure 3.6: Number of tweets corresponding to each humanitarian category.

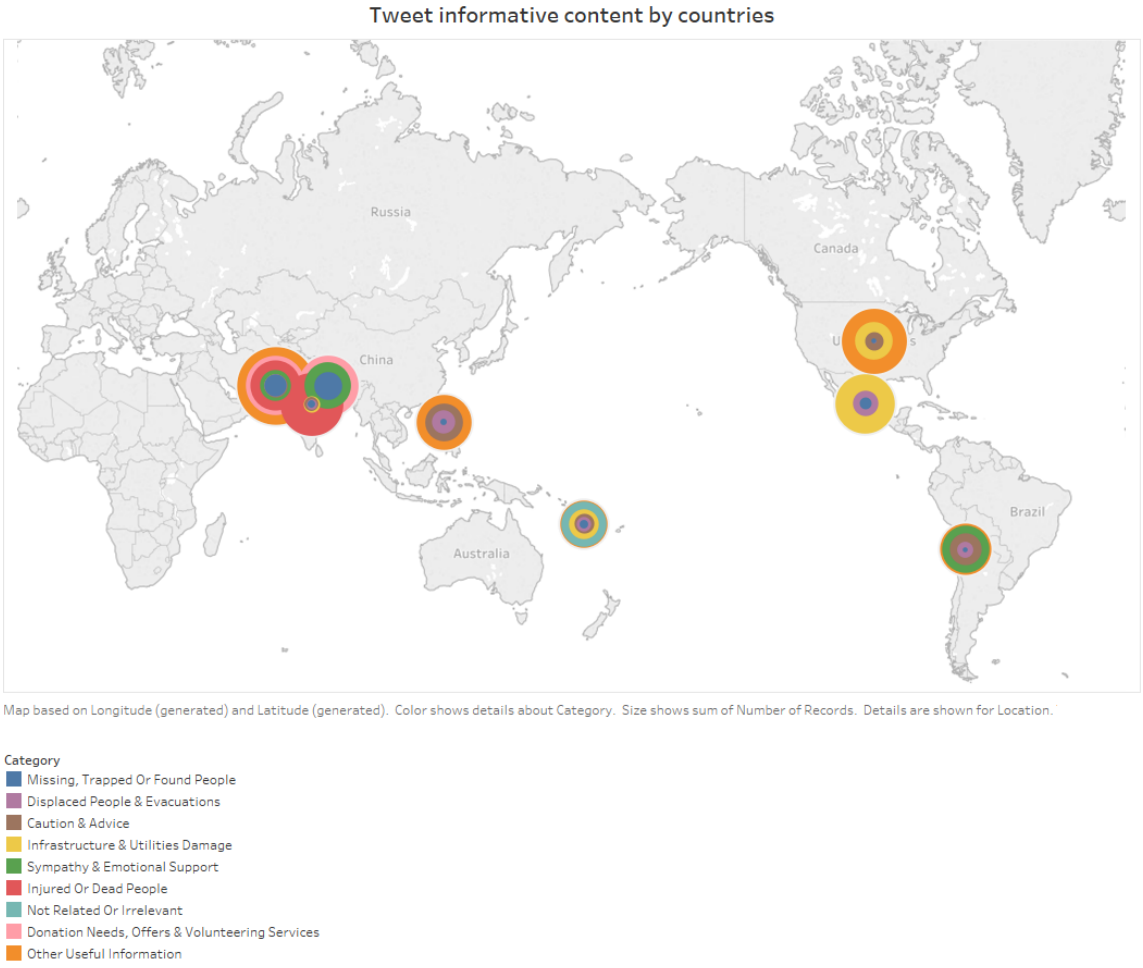


Figure 3.7: Geographic distribution of tweets by humanitarian categories.

ian categories. From the figure, it can be seen that a vast majority of tweets (nearly 35 percent) are classified as *Other Useful Information*. Approximately 2,500 tweets are classified as *Donation Needs, Offers & Volunteering Services, Injured or Dead People* and *Not Related or Irrelevant* respectively. It is also observed that roughly around 2,000 tweets are classified as *Infrastructure & Utilities Damage* and *Sympathy & Emotional Support*. 1,000 tweets are classified as *Caution & Advice* and around half this number of tweets (between 400-600) are classified as *Displaced People & Evacuations* and *Missing, Trapped, or Found People*.

Also, figure 3.7 is provided which depicts the geographic distribution of tweets by their information content. It is important to note that the size of the concentric circles correspond to the number of tweets pertaining to a specific humanitarian category for a

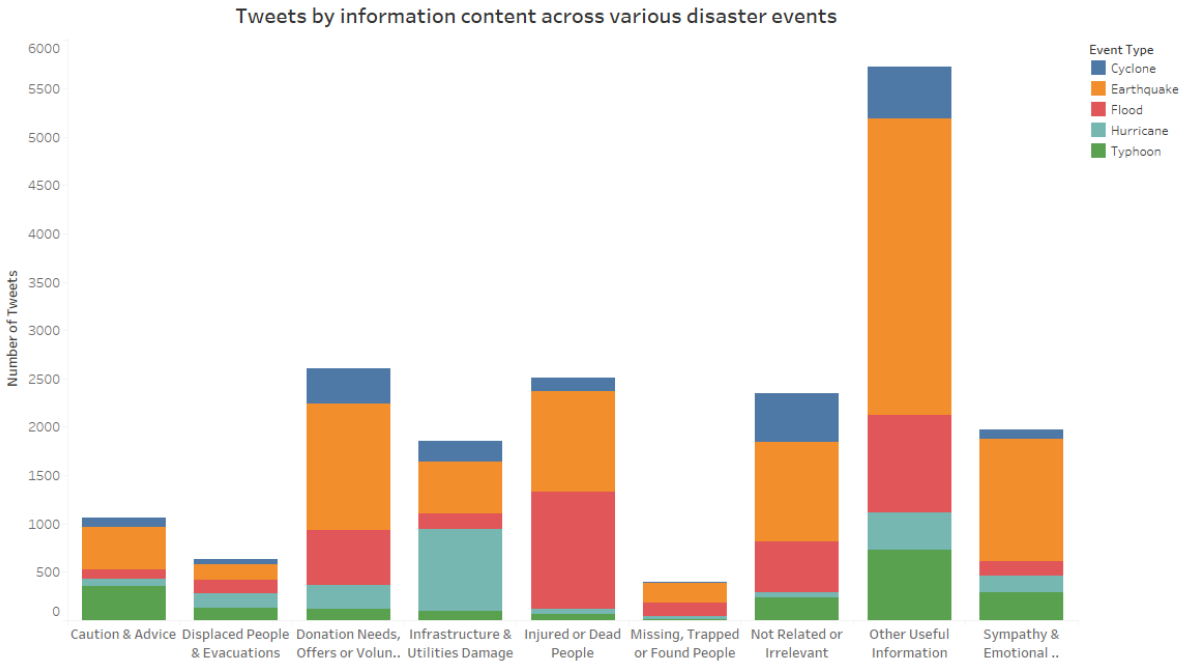


Figure 3.8: Distribution of tweets in each humanitarian category across the five disaster events.

specific disaster type. It can be seen directly from the map that most of all the tweets in case of Pakistan and Nepal Earthquake events contain information on *Missing, Trapped or Found People* than any other disaster events. This consequently leads to a high number of tweets related to *Donation Needs, Offers or Volunteering Services* due to the presence of substantial information on people in need of help. One can also note that most of the tweets classified as *Injured or Dead People* are coming from India Floods and Pakistan Earthquake events. Also, it is observed that there is a lot of infrastructure and utilities damage in Mexico and US Earthquake based on the high number of such tweets for those disaster events. Lastly, a huge number of *Other Useful* tweets are observed for US Earthquake and Philippines Typhoon, while a major share of tweets on Vanuatu Cyclone Pam are classified as *Not Related or Irrelevant*.

From figure 3.8, it is observed that the four earthquake events have highest number of tweets containing *Other Useful Information* while the cyclone at Vanuatu has least helpful tweets as most of them are either off-top or irrelevant. The flood events in India and Pakistan have a majority of tweets classified as *Injured or Dead People*,

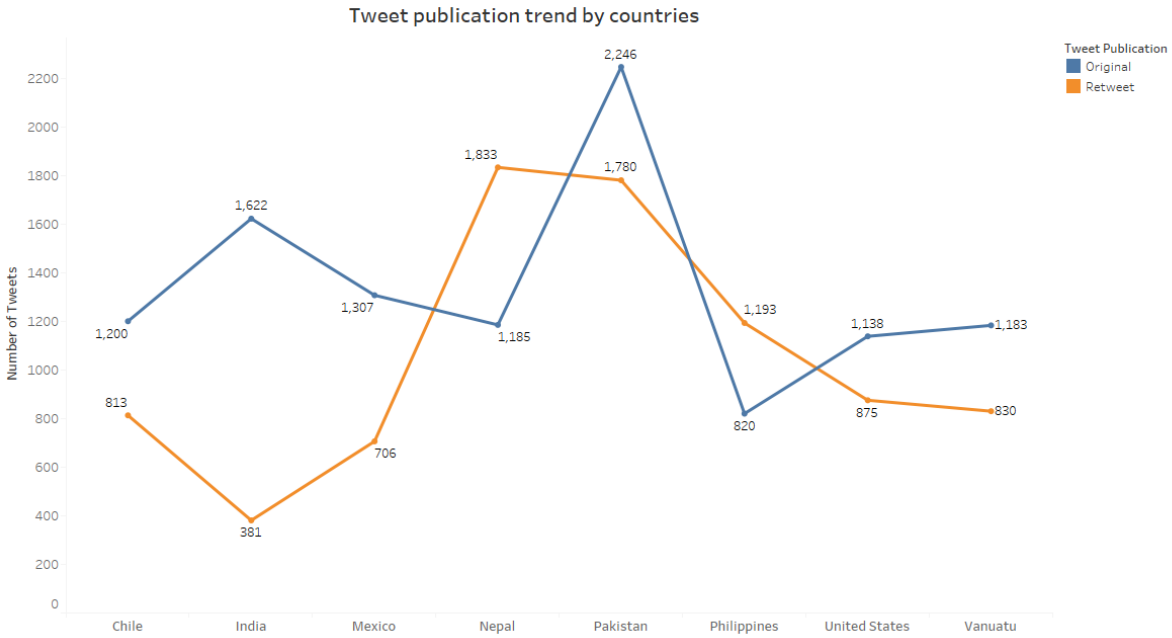


Figure 3.9: Trend of tweet publication among different countries in the dataset.

and a high number of tweets requesting donation needs and volunteering services. This is an immediate actionable information with the potential of saving the lives of missing, trapped and found people if the humanitarian services are lent to them on time. Hurricane Odile has caused maximum infrastructure and utilities damage apart from earthquake events. Emotional support and sympathetic tweets are found for typhoon and earthquake events, for the most part.

Figure 3.9 illustrates the publication trend of tweets among different countries. The blue line represents the original tweets while the orange refers to re-tweets. It can be seen from the figure that India and Pakistan have the highest number of original tweets at the time of disaster while Philippines and United States have the lowest. Also, Nepal and Philippines have the highest number of re-tweets than any other country. And, United States, Vanuatu and Philippines seem to have near-similar number of tweets and re-tweets as against India, Nepal and Pakistan where there is a significant difference in the number of original tweets and re-tweets. This could indicate the tweet publication behavior of different countries at the time of a disaster event. However, the dataset is not all inclusive (does not contain the full list of tweets

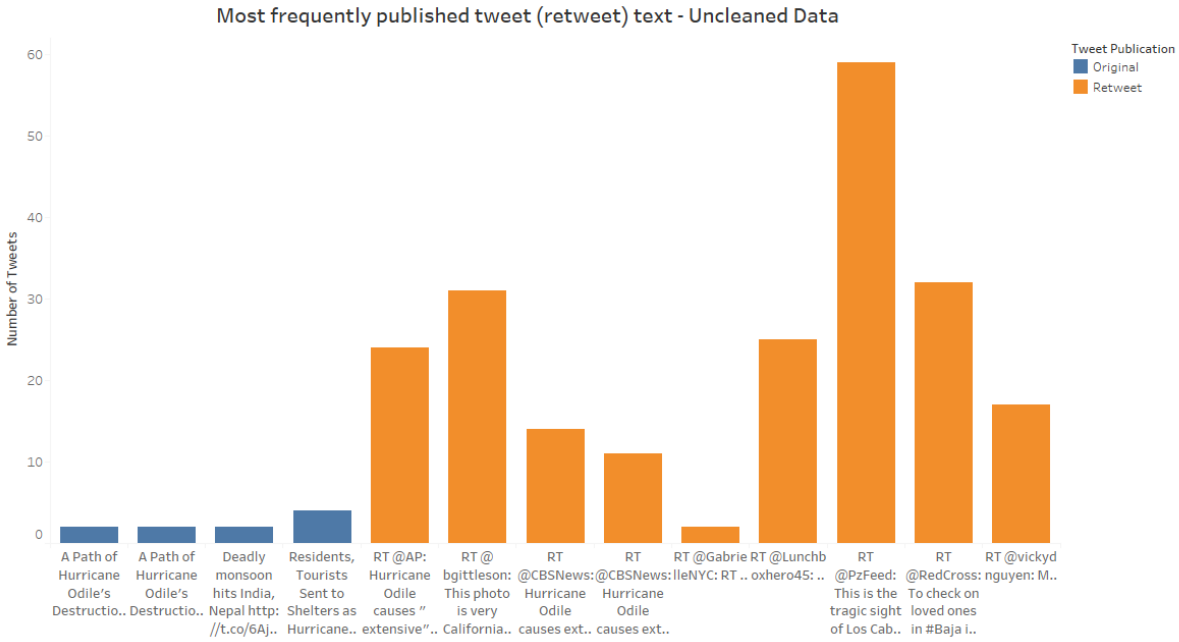


Figure 3.10: Textual content of tweets based on their tweet/re-tweet frequency.

from start to end date of a disaster) and such remarks may or may not hold true. Such an insight is worthwhile for differentiating the relevant and original tweets from irrelevant and duplicated tweets.

On the other hand, Figure 3.10 depicts the most frequent tweets and re-tweets with their textual content. The textual content of the tweet is displayed on the x-axis while the y-axis shows the number of times the tweet was published. It can be seen clearly that the number of times a tweet was re-tweeted ranges from two to nearly sixty times in the dataset while there is a much less duplication of original tweet texts (not occurring over 3 times) in the dataset. This figure was plotted using actual dataset without cleaning the tweet text, so it includes the URL's, hash-tags, twitter handles and symbols etc. as evident from the tweet content in the figure.

3.6 Data Preparation

As the original tweet text contains all sorts of symbols, slang words, twitter handles, hash-tags, URL's, improper grammar etc. owing to limited sentence length, it gets

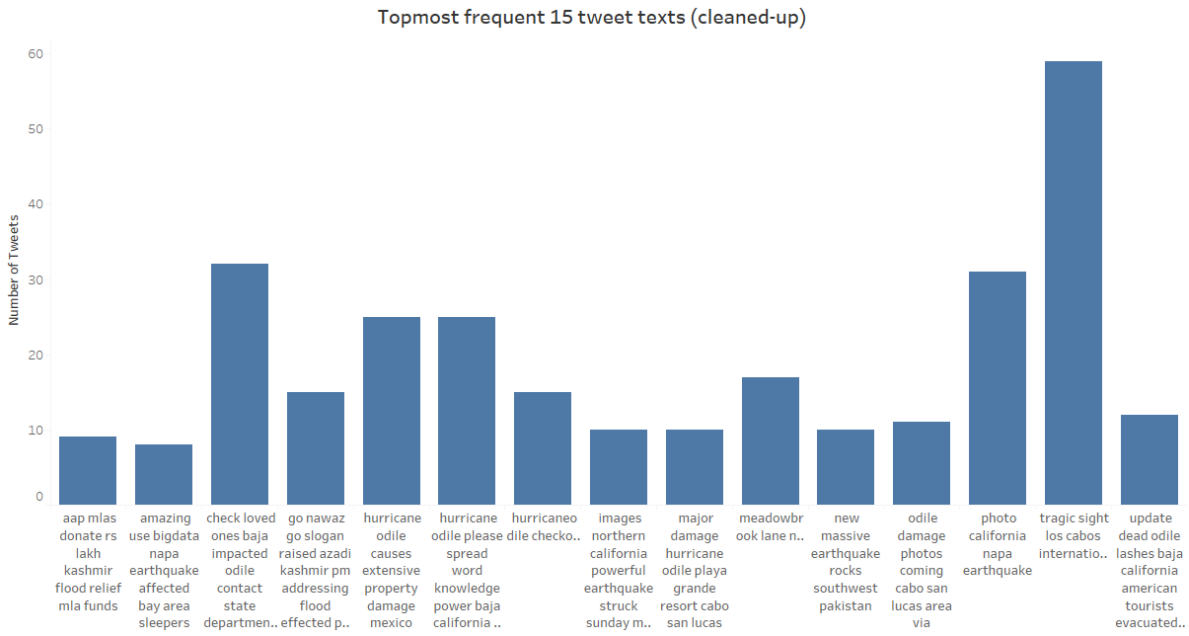


Figure 3.11: Most frequently occurring tweet text (after tweet-cleaning).

difficult to process the tweets and train them in a classifier model to perform the tweet classification based on the tweet text. As the current project intends to classify tweets using several machine learning algorithms into one of the many humanitarian categories and compare them in terms of precision, recall and F-scores, while also trying to use tweet sentiments as one of the features to improve the classification accuracy of the models, it is important to clean the tweets before feeding them into the classifier models as well as before performing sentiment analyses on them.

Tweet data preparation in this case includes the task of removing punctuations, stop-words, numerics, symbols, URL's, and other imprecise & improper language and words within the tweets. This was performed in R Studio using the *Text Mining* (TM) package. The dataset was stemmed, lemmatized, cleaned for URL's, hashtags, @ and other symbols and numerics. The dataset was read as a dataframe in R and was later converted into a plain corpus and finally was outputted as a CSV file.

Figure 3.11 shows the most frequently occurring tweet texts (including both original and re-tweets) after cleaning them. One can clearly see some of the tweet texts repeating from about 10 to 60 times. While the most frequent tweet text certainly

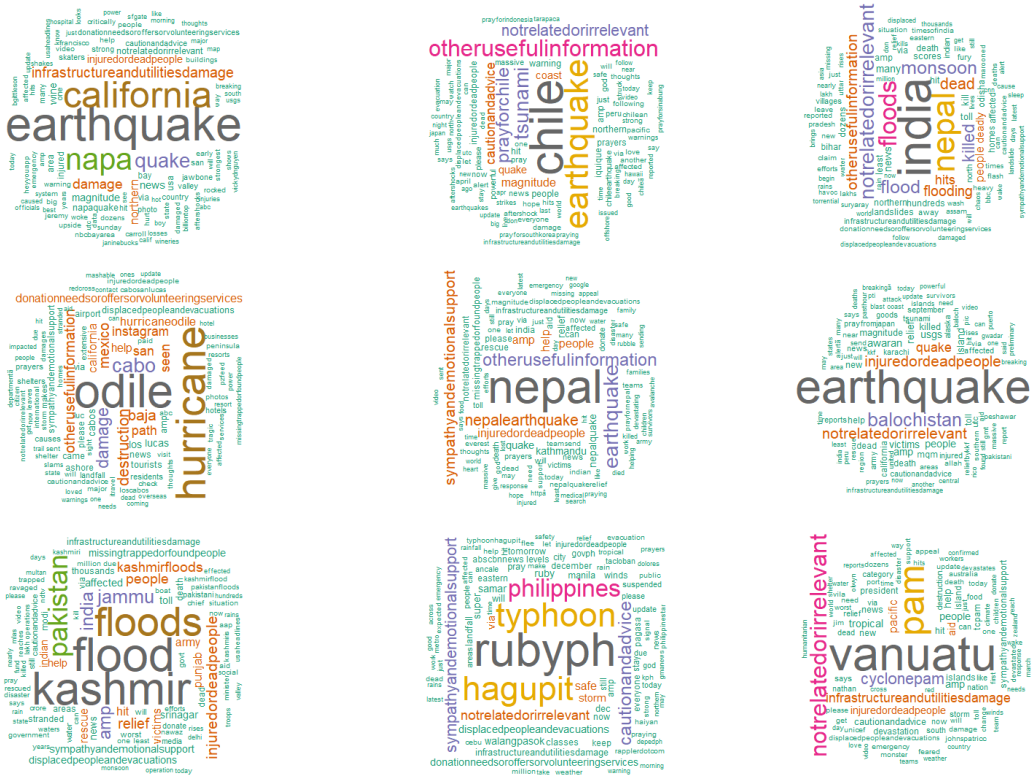


Figure 3.12: Word-clouds corresponding to different disaster events. The size of a word in a word-cloud is indicative of the word frequency.

Note: Only the top 100 most frequent words are plotted.

provides an insight into the structure and form of tweets, the word cloud provides an individual word by frequency map of the words inside each tweet text. The word-cloud was again generated in R Studio and is shown in figure 3.12 where each word-cloud corresponds to a single disaster event. The size of a word in a word-cloud is indicative of the word frequency. The words located in the middle of the word-cloud and with a bigger font are the most frequently occurring words in the tweets. The task was performed on cleaned up tweets and plotted using R programming language.

The cleaned up dataset are then utilized to perform sentiment analyses, named-entity extraction, contextual categorization as well as tweet text classification using several state-of-the-art machine learning classifiers.

3.7 Modeling

The research aims to perform multi-dimensional content analyses of tweet-text. This is done as three separate sub-tasks of sentiment analyses, named-entity recognition and contextual tweet categorization using different tools and techniques. The results obtained from sentiment analyses of tweets are then used as additional features in the task of tweet text classification into one of the nine predefined humanitarian categories. The two tasks are mentioned here in terms of data modeling performed:

3.7.1 Textual content analyses of tweets

Textual content of a tweet is analyzed in terms of sentiment scores, polarity and subjectivity of opinions expressed in a tweet, extracting important named-entities like names of people, locations and organizations, and lastly, the contextual categorization of tweets as per IAB quality assurance guidelines as detailed in section 3.3 of the current chapter. A variety of tools were used to perform these tasks, the choice of tools was influenced by in-depth literature survey of the currently existing methods. Sentiment analyses of tweets was initially done using R package *Tidy-Text*. This package is based on tidy text data frames and supports functions for the conversion of text to and from tidy formats. Sentiment scores for tweets belonging to different humanitarian categories was generated ranging from -5 (negative) to +6 (positive) using *AFINN lexicon*. *Senti-strength* is another popular tool for performing sentiment analyses, especially suited for short-texts, like tweets. This was used to generate sentiment scores in three different settings: *(a) Generate sentiment scores from tweets using all the sentiment words in a sentence, (b) Generate sentiment scores from tweets using the average of all sentiment words in a sentence,* and *(c) Generate sentiment scores from tweets using the strongest of all sentiment words in a sentence.* In addition to this, a GUI based tool for predictive analytics, called *Rapid Miner*, was used to obtain sentiment polarity and sentiment subjectivity from tweets. There are several text analysis extensions like Rosette, General Text Miner, AYLIEN etc. which are used in conjunction with Rapid Miner to obtain useful insights from text. Text

analysis packages *AYLIEN* and *Rosette* were used to obtain sentiment polarity, subjectivity as well as sentiment scores. The sentiment analyses results obtained using these methods: R, Senti-strength and Rapid Miner were then compared against each other. Lastly, named-entity recognition and contextual categorization of tweets was performed using *Rosette* package in Rapid Miner. It is important to note that of all the multi-dimensional textual content analyses of tweets, only *sentiment scores, polarity* and *subjectivity* were used as additional features supplied to machine learning classifiers for tweet classification.

3.7.2 Classification of tweet text

The aim of the experiment is to build several predictive models to automatically classify tweets into one of the nine predefined humanitarian categories. Several state-of-the-art supervised machine learning classifiers were trained for this purpose. These classifiers were trained using *10-fold cross validation* to avoid over-fitting. A popular open-source machine learning environment for knowledge engineering developed by University of Waikato, New Zealand, known by the name, *Waikato Environment for Knowledge Analysis* (WEKA) was used to process the tweets to generate word vectors, to rank the word vector features by their level of informativeness, to tokenize the individual tweets using various tokenization schemes (alphabetic, word, uni-gram, bi-gram, tri-gram and their combinations) and to finally train and test the performance of different machine-learning classifiers for tweet text classification. WEKA is a collection of visualization tools and algorithms for predictive modeling and data analysis. There are easily accessible graphical user interfaces to access various functions that work together to perform knowledge engineering on a dataset in WEKA. A total of 15 different classifiers were trained on both the datasets (original and enhanced) in this study. This includes state-of-the-art machine learning approaches for text classification like *Naive Bayes, Decision Trees, Logistic Regression, Neural Networks, Support Vector Machines, Random Forests* etc. which are covered in detail in Chapter 4.

3.8 Evaluation

The accuracy of the classifiers from *10-fold cross validation* is initially assessed. The aim is to obtain a higher accuracy score when using additional features (tweet sentiments) for text classification. The accuracy of the text classification task also depends on the tokenization scheme used. An analysis of the classification performance of various machine learning classifiers using different token lengths (alphabetic, word, uni-gram, bi-gram, tri-gram and their combinations) was also performed in this study which resulted in picking up the token sequence yielding highest classification accuracy. Other metrics used for evaluating classification performance and to compare the results are weighted average *Precision, Recall* and *F-score* along with *% accuracy of correctly classified tweet text*. The best performing model is then analyzed in greater detail.

Evaluation of the results for each of the classifiers on both the datasets (original and enhanced) is finally carried out using a *Paired-T Tester* at a *statistical significance level of 0.05* to determine whether the results produced are statistically significant or not. The paired T-tester has two competing hypotheses, the null and the alternative. The null hypothesis assumes the true mean difference between paired samples to be zero while the alternative assumes the true mean difference between paired samples to be non-zero. If there exists a significant difference in the *percentage accuracy of correctly classified tweet text* using additional sentiment features, a justification to reject the null hypothesis will be made.

Chapter 4

Implementation and Results

This chapter details the execution of the experiment conducted in this study accompanied with an evaluation of the methodology. As the data description, exploration and preparation stages have already been covered in the previous chapter, this chapter jumps straight towards the task of performing multi-dimensional textual content analyses of tweets as well as text classification. The results obtained from each task are summarized towards the end.

4.1 Multi-Dimensional Content Analyses of Tweets

In this section, an attempt to gain an understanding of the textual information posted on social media during disaster events will be made. As the tweets posted at the time of disaster events are one of the most immediate sources of situational awareness and actionable information for several humanitarian organizations, it is extremely useful for disaster management and recovery teams to plan timely evacuation of people in need of volunteering services. This allows decision-makers and responders to analyze and quickly filter out the irrelevant information and plan the time-critical mitigation process. There are three types of content analyses that were performed as a part of this study which include *sentiment analyses of tweets*, *extracting named-entities from tweets* and lastly, *contextual categorization of tweets* into Tier-1 categories as per IAB quality assurance guidelines. Each of them are covered separately in this section.

4.1.1 Sentiment Analyses of Disaster-Related Tweets

Determining the sentiments of people during disasters can provide an overview of concerns, issues, panics and problems faced by public at large helping responders establish stronger situational awareness of the disaster zone. The sentiment analyses of tweets was performed using three different techniques, which are summarized in this section. An analyses and comparison of the techniques then follows.

Sentiment Analysis is the broad task of assigning *sentiment labels* which define the polarity of a given text in consideration. The sentiment polarity is used to detect whether a given sentence is positive, negative or neutral. Some tools perform a binary polarity while others include neutral polarity as well. It is important to understand that sentiment analyses can be performed at *document level* (covering the entire document), *sentence level* (for each sentence in the document) and *entity/aspect level* (aspects from within a sentence). While entity or aspect level analyses is much more detailed in coverage, the current work focuses on sentence level sentiment analyses.

Sentiment analyses on twitter posts is essentially carried out in three capacities: *Lexical analysis, Machine learning based analysis* and *Hybrid/Combined analysis*. Lexical analysis uses a dictionary or sentiment lexicon, machine learning based approach uses machine learning classifiers while hybrid analysis use a combination of both lexical and machine learning approaches to perform sentiment analyses on a dataset.

Sentiment Analysis Using R

R has been extensively used to perform sentiment analysis owing to the availability of various sentiment packages. The *"SentimentAnalysis"* package is a commonly used dictionary-based approach to perform sentiment analysis using a variety of existing domain-specific dictionaries. Furthermore, it can also create customized dictionaries. It is possible to supply a domain-specific dictionary using a *Quanteda* package in R. Although, there were lots of labeled domain-specific sentiment dictionaries available, crises-related lexicons except for *Emergency Management* (EM) lexicon (Temnikova & Castillo, 2015) containing over 7,000 crisis words, were not available. EM terms was the only crises-related lexicon as per the awareness of the author. Since the EM

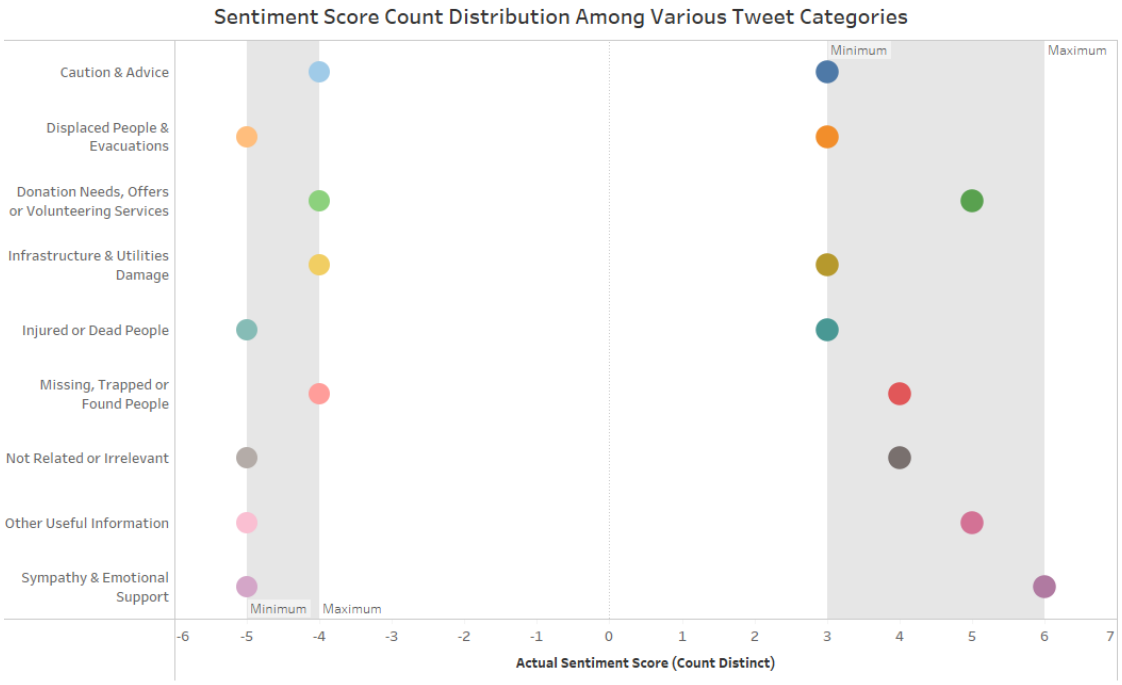


Figure 4.1: Sentiment Score (Distinct Count) across various tweet categories.

lexicon had no labeling of tweets into *Positive* or *Negative* sentiment categories, it wasn't a feasible option to be used.

Tidytext package is the most widely used package for sentiment analysis in R and it gives access to 4 sentiment dictionaries: AFINN, BING, NRC and Loughran. This package was used in this work. The sentiment lexicons provided with this package do not contain every English word because most words in English are neutral in polarity, also these methods are based on uni-grams only and hence cannot catch the mentions of sarcasm or negations. For a twitter-disaster dataset, this isn't much of an issue as the tweets related to disaster events mostly do not involve euphemisms and sarcasms. Another advantage with using this package is that since the dataset is sentence-sized and does not include long paragraphs, the positive and sentiment scores aren't averaged out to zero, hence the sentiment analysis produces good results.

Figure 4.1 illustrates the results obtained. On the x-axis lie the distinct sentiment score counts while the y-axis denotes the 9 tweet categories. The synchronized dual axis-chart demonstrates the distribution of sentiments ranging from -5 (Negative) to +6 (Positive). The gray region shows the maximum and minimum score values. The

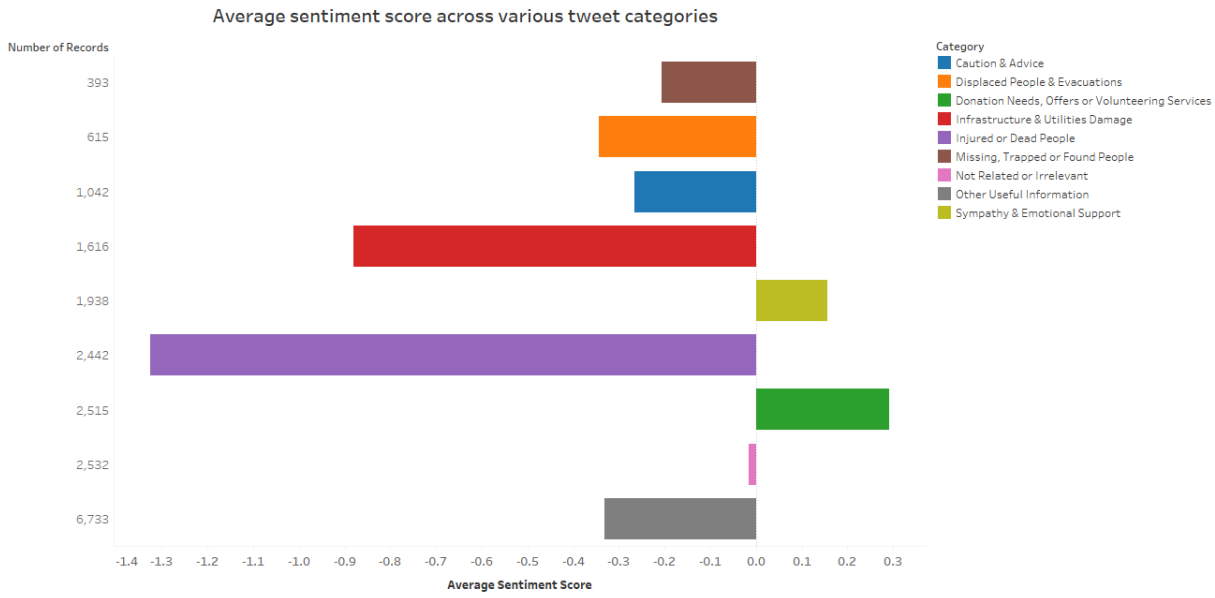


Figure 4.2: Average Sentiment Score across various tweet categories.

negative score ranges from -4 to -5 while the positive score ranges from +3 to +6. Most of the negative score is seen in case of *Injured & dead people* and *Displaced people & evacuations* while most of the positive score is seen for *Sympathy & emotional support, Donation needs, offers & volunteering services* and *Other useful information.*

Figure 4.2 on the other hand, illustrates the averaged sentiment score across the 9 tweet categories. The info-graphic demonstrates the distribution of average sentiment scores ranging from -1.4 (Negative) to +0.3 (Positive). These numbers are generated by averaging the overall positive and negative sentiment scores in the dataset. The y-axis represents the number of tweets in each of the 9 humanitarian categories. Most of the negative score is seen in case of *Injured& dead people* and *Infrastructure & utilities damage* while the only positive score is observed for *Sympathy & emotional support* and *Donation needs, offers & volunteering services*, which is, what was expected. This shows that sentiment analysis using R did a pretty good job identifying the negative and positive sentiments based on the textual content of tweets.

Sentiment Analysis using Senti-Strength

Several studies (D'Andrea, Ferri, Grifoni, & Guzzo, 2015; Abbasi, Hassan, & Dhar, 2018) repeatedly suggest better performance of *Senti-Strength* stand-alone sentiment analyzer tool for twitter datasets than most other tools. The upgraded version of Senti-Strength, called *Senti-Strength 2* has an extended sentiment dictionary covering a variety of data coming from social websites such as MySpace, BBC, Digg and Runners World etc. It is also trained on more than 4,200 tweet texts and around 3,400 texts from Youtube. The number of sentiment terms was increased from 890 to 2,489 in the current version of this tool to facilitate working on many different social websites. The greatest advantage is that it has been tried and tested on 6 different social media datasets, hence it is quite robust and accurate in sentiment classification. It can detect positive and negative sentiment strengths in short informal text quite easily. Domain-specific algorithms which are trained on a specific data type are more accurate than generic sentiment algorithms. The main issue with a domain-specific sentiment classifier is its inability to adapt well to another domain, example, a sentiment analyzer trained on movie reviews can't work work well for book reviews. This problem is called the problem of *domain adaptation*. As mentioned in the previous section, there was no labeled training data related to disaster-domain, sentiment-analyses of tweets using Senti-Strength was thus performed using the pre-existing tweet datasets that this tool was initially trained on.

The performance of Senti-Strength is comparable to state-of-the-art machine learning algorithms making it a great choice of a sentiment analyzer tool. It is important to note that Senti-strength provides the output sentiment score in a range of 1 to 5 where 1 & -1 represents weakest emotion respectively (including both polarities) and 5 & -5 represents the strongest emotions. For example, the sentiment scores of the following words is:

ache = -2, coolest = 3, dislike = -3, excruciating = -5, encourage = 2, hate = -4, lover = 4

Senti-strength uses a *sentiment word strength list* which is essentially a list of all commonly observable emotion words (appearing in short social media texts) along with

their sentiment scores. It is important to note that the data going into Senti-strength is translated and corrected for spelling, repeated words and other common errors. A list of *booster words* (adverbs) is used to alter the sentiment strength, example, if "happy" has a sentiment score of +4, "very happy" will have a sentiment score of +5 and so on. Negations are also taken into consideration. This is unlike the *TidyText* package in R which simply worked on uni-grams and completely overlooked negations and adverbs. Repeated letters which is the usual writing style in tweets boosts the strength of a sentiment word, example, "nice" has a lower score than "niiiiice". Emoticons and exclamation marks amplify the sentiment score by +2 unless negative. Repeated punctuations also boost sentiment score, example, "good" is +3 and "good!!!!" is +5.

Senti-strength uses an extended list of 7 input files to generate the sentiment scores:

Emotion LookUp Table - a list of emotion words with a strength 1 to 5 or -1 to -5.

Emoticon LookUp Table - a list of emoticons with a strength 1 to 5 or -1 to -5.

English Word List - a list of English words to correct spelling mistakes.

Idiom Lookup Table - consists of idiomatic phrases and sentiment strengths.

Negating Word List - a list of negation words like not, dont, can't.

Slang Lookup Table - slang words and translations in informal text.

Booster Word List - a list of sentiment intensity modifiers like very, much, some, extremely, quite, tad, few.

Senti-Strength works in three different configurations to produce sentiment scores for a given data instance (a single sentence, or a single tweet etc.):

- Generates sentiment scores using *all the sentiment words* in a given data instance.
- Generates sentiment scores using the *average of all sentiment words* in a given data instance.
- Generates sentiment scores using the *strongest of all sentiment words* in a given data instance.

The sentiment scores generated using the three different working configurations produce different results. This work provides the sentiment analyses results using

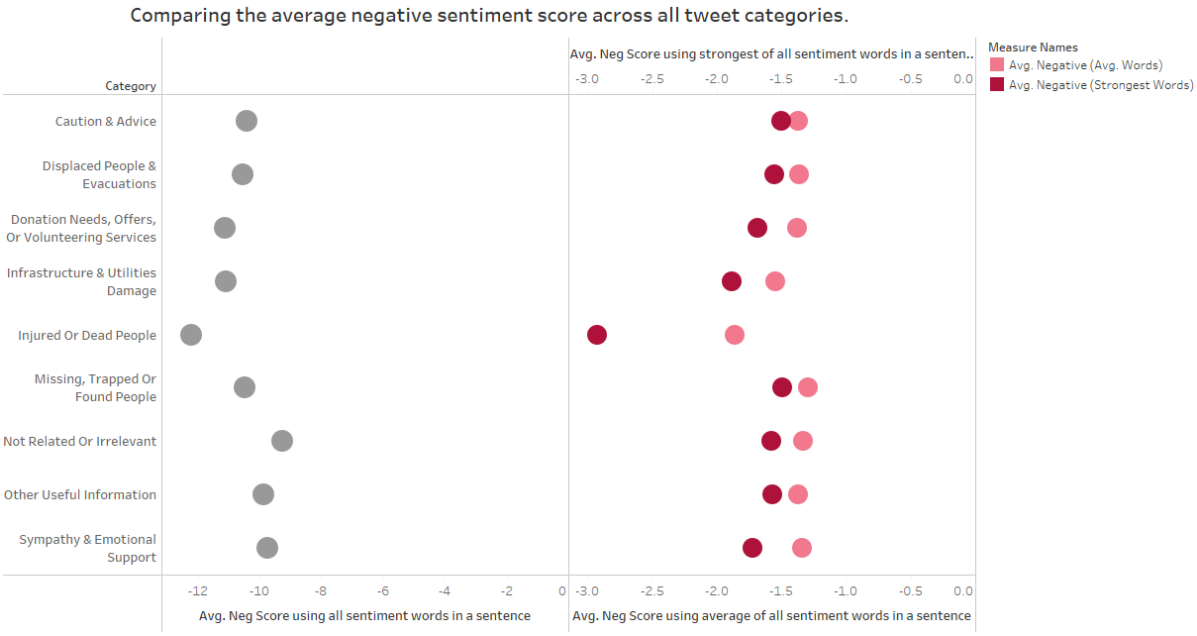


Figure 4.3: Comparing Average Negative Sentiment Score across various tweet categories using all three configurations.

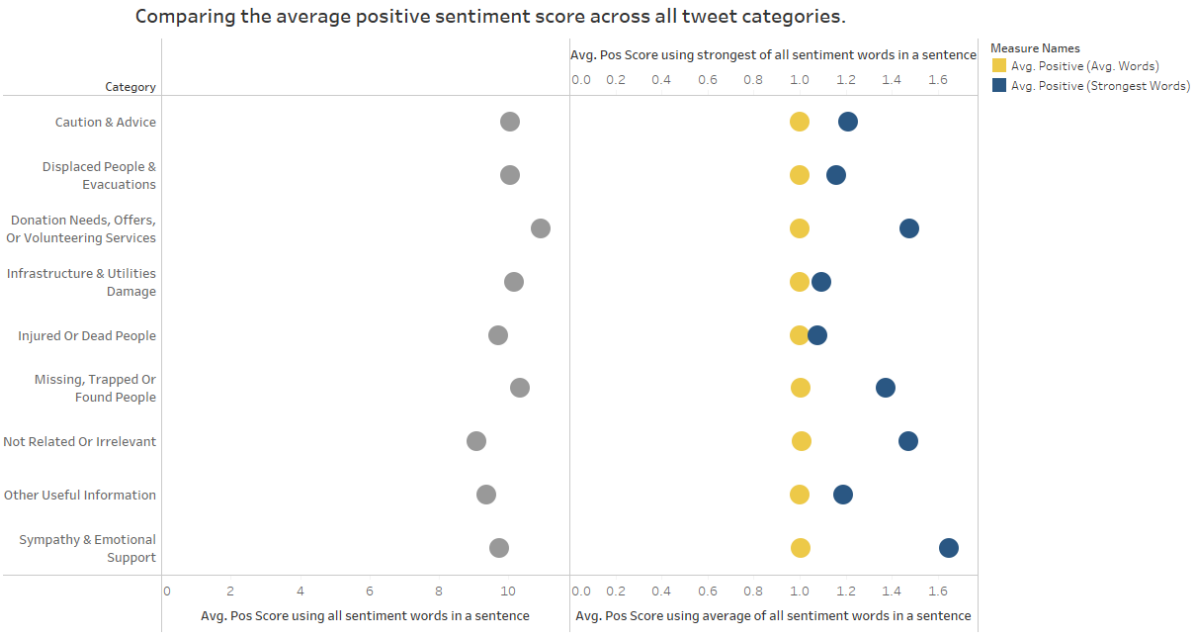


Figure 4.4: Comparing Average Positive Sentiment Score across various tweet categories using all three configurations.

all the three configurations as well as their comparison. SentiStrength provides two sentiment scores per data instance (sentence or tweet): *A negative sentiment score ranging from -1 (not negative) to -5 (extremely negative)* and *a positive sentiment score ranging from +1 (not positive) to +5 (extremely positive).* This is backed up by research in psychology suggesting that humans process emotions in parallel (both positive and negative sentiments) for a single sentence (tweet). A sentence is considered neither entirely positive nor entirely negative, but a combination of both sentiment polarities.

Figures 4.3 and 4.4 depict the average negative and average positive sentiment scores generated using these three configurations and provides their comparison. One can see from figure 4.3 that the average negative sentiment score ranges from -10 to -12 using all sentiment words in a tweet while the average negative sentiment score ranges from roughly -1.5 to -3.0 using both the average and strongest of all sentiment words in a tweet. One can also observe a greater variance in the average negative score generated using the strongest of all sentiment words. Also, from figure 4.4, the average positive sentiment score ranges from +9 to +11 using all sentiment words in a tweet while the average negative sentiment score ranges from roughly +1.0 to +1.7 using both the average and strongest of all sentiment words in a sentence. One can also observe a greater variance in the average positive score generated using the strongest of all sentiment words.

It was concluded that the strongest of all sentiment words in a sentence is a better sentiment score technique than others. Also, interesting to note is the fact that the findings of sentiment analyses using Senti-strength are in alignment with the results obtained using R. Both of them show an overall positive sentiment score for the tweet categories of *Sympathy & emotional support, Donation needs, offers & volunteering services*, and an overall negative score for *Injured & dead people* and *Infrastructure & utilities damage*. As the sentiment scores calculated using the strongest of all sentiment words in a sentence is of the better quality, it was finally used to calculate the average sentiment scores (both positive and negative polarities) as shown in figure 4.5. One can see that the most negative average score is observed for *Injured & Dead People*

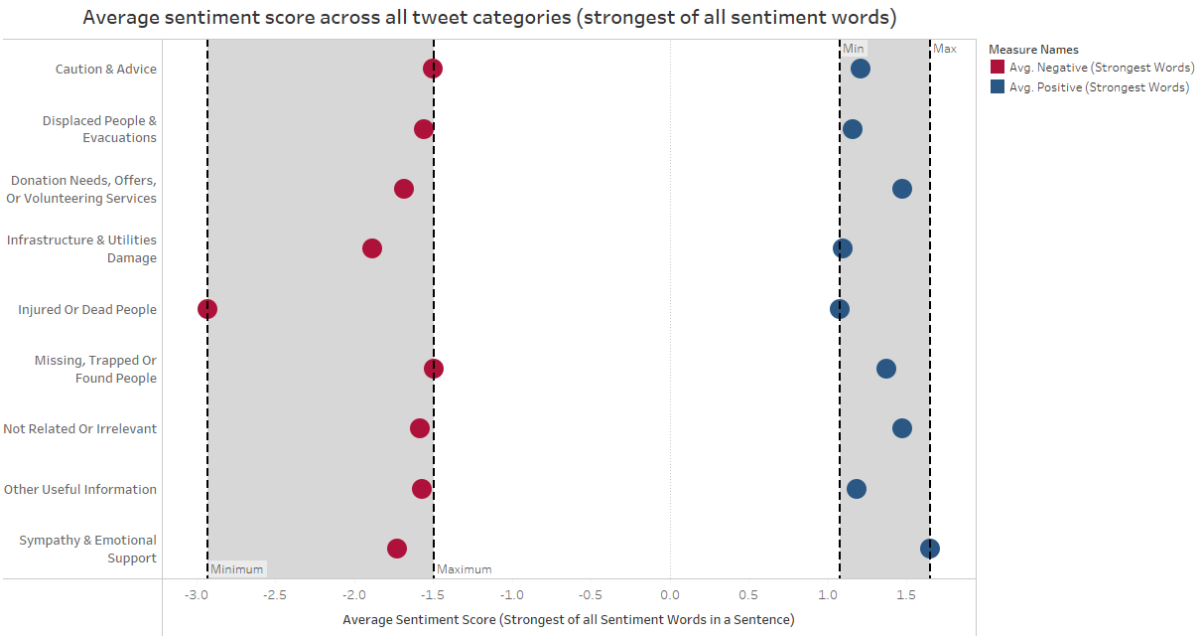


Figure 4.5: Average Sentiment Scores (Positive and Negative) across various tweet categories using the strongest of all sentiment words in a data instance.

followed by *Infrastructure & Utilities Damage*. The rest of the tweet categories have an almost similar average negative score of around -1.5. Similarly, the most positive average score is observed for *Sympathy & Emotional Support* followed by *Donation needs, offers & Volunteering Services, Not Related Information* and *Missing, Trapped or Found People*, while the rest of the tweet categories have an almost similar average positive score of around +1.0.

Figure 4.6 depicts the average positive and negative sentiment scores across the identified 4 most important tweet classes from the standpoint of sentiments - *Injured or Dead People, Infrastructure & Utilities Damage, Sympathy & Emotional Support* and *Donation Needs, Offers & Volunteering Services*, where the blue bars denote the average positive score while the red bars denote the average negative score.

Sentiment Analysis using Rapid Miner

Rapid Miner is a data science software platform that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive

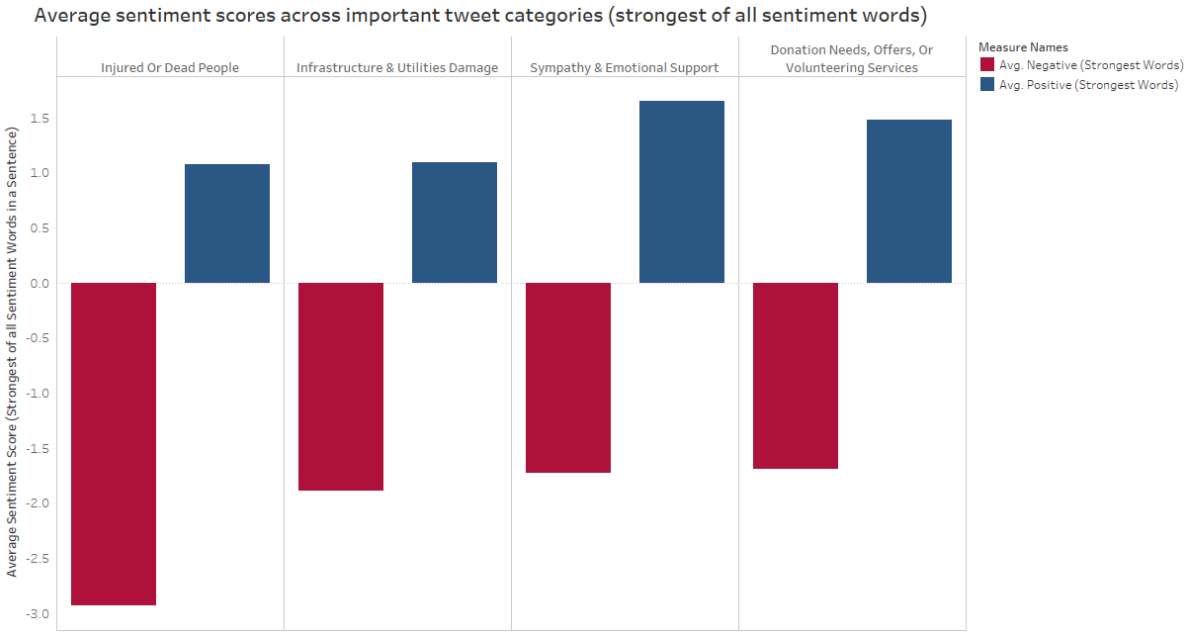


Figure 4.6: Average Sentiment Scores across the most important tweet categories using strongest of all sentiment words in a data instance.

analytics. It is commonly used for business and commercial applications as well as for education and research. Rapid Miner is developed on an open core model and was used for this project. There are several in-built operators that allow for *Data Access, Blending, Cleansing, Modeling, Scoring, Validation and Utilities* that are accessible as functions from inside Rapid Miner studio. Depending on the type of dataset in use, there are several additional *extensions* available that work along with the usual operators to build effective machine learning models. As the current project dealt with text processing and analytics, three specialized extensions were downloaded. These were - *AYLIEN Text Analysis Extension, Rosette Text Analytics* and *Text Processing*. Sentiment analyses was performed on twitter-disaster dataset using both the AYLIEN as well as Rosette extension.

Using AYLIEN Text Analysis Extension - Text Analysis by AYLIEN is an extension made up of different operators that allows us to analyze and make sense of textual data supplied to it. The different operators contained in this extension include: *Sentiment Analysis*, *Entity Extraction*, *Language Detection*, *Hashtag Suggestion* and

Related Phrases. *Sentiment Analysis Operator* from AYLIEN was used to perform the task of Sentiment Analyses on disaster-related tweets. This operator does two things:

1. Classifies the tweets according to polarity as predicted into three classes - Positive, Neutral and Negative.

2. Classifies the tweets according to subjectivity as expressed into two classes - Objective and Subjective.

While understanding the polarity of a tweet only involves observing the strength of sentiment words used, understanding objectivity involves observing the context and content of a tweet. An objective sentence differs from a subjective sentence in terms of the type of information supplied, objective is usually about some factual information while subjective contains specific beliefs and personal opinions, feelings. Classifying a sentence as opinionated or not opinionated is called *subjectivity classification* while classifying a sentence as expressing positive, negative or no sentiments is called *polarity classification*.

In addition to classifying tweets as per their polarity and subjectivity, AYLIEN provides a confidence level measure which is a number between 0 and 1 denoting the confidence with which the task of classification is performed. So, 1 denotes a confidence of 100% and 0.5 denotes a 50% confidence and hence a debatable classification.

Figure 4.7 represents tweets classified into Objective or Subjective as well as into Positive, Negative or Neutral for the entire dataset. Around 70% of the total number of tweets are classified as Neutral, around 21% as negative and the remaining as positive. Also, interesting to observe is the fact that the number of neutral objective and neutral subjective tweets is very similar as well as negative objective and negative subjective. However, the number of positive subjective tweets is almost double the number of positive objective tweets.

Tweet Polarity: From figure 4.8, the major share of neutral tweets is coming from *Other Useful Information* while maximum negative tweets are coming from *Injured & Dead People, Infrastructure & Utilities Damage* and *Missing, Trapped or*

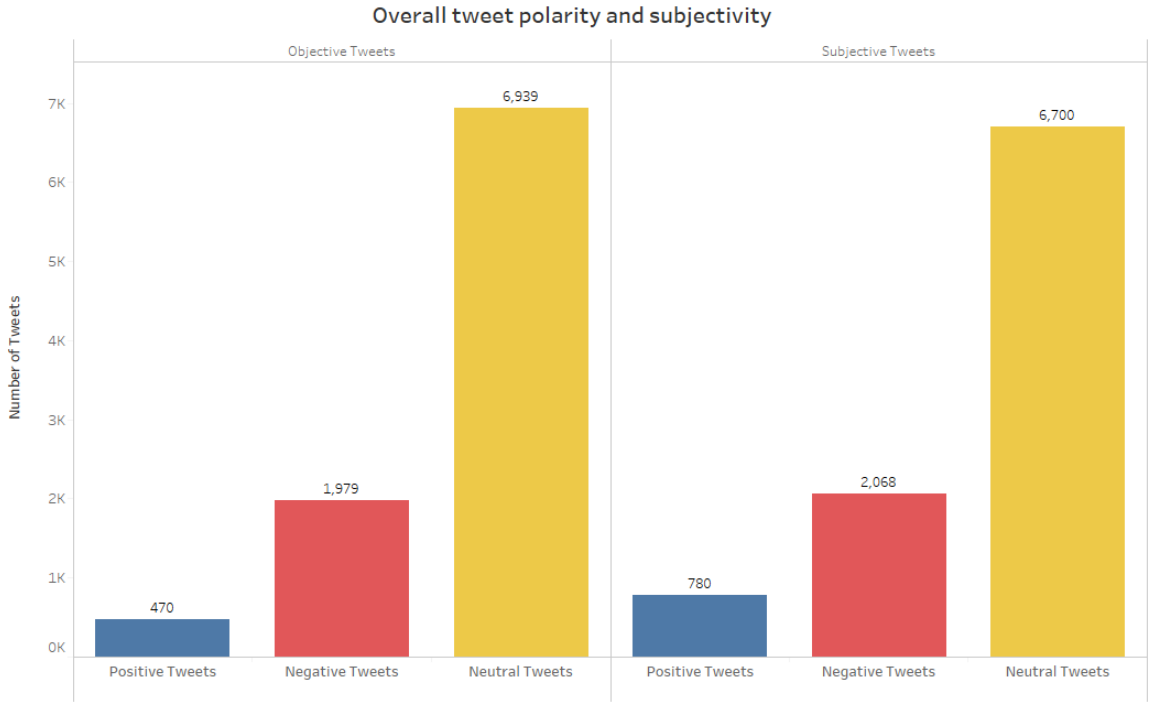


Figure 4.7: Polarity and Subjectivity of tweets.

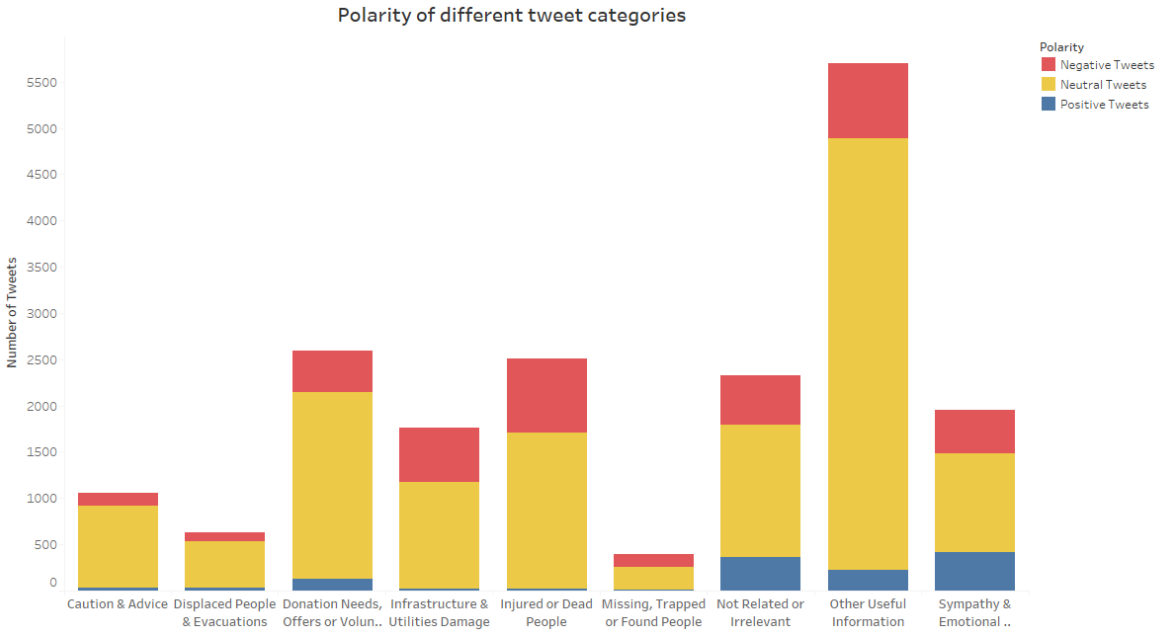


Figure 4.8: Polarity of various tweet categories.

Found People categories. Tweets belonging to *Sympathy & Emotional Support, Not Related Or Irrelevant, Donation Needs, Offers & Volunteering Services* and *Other*

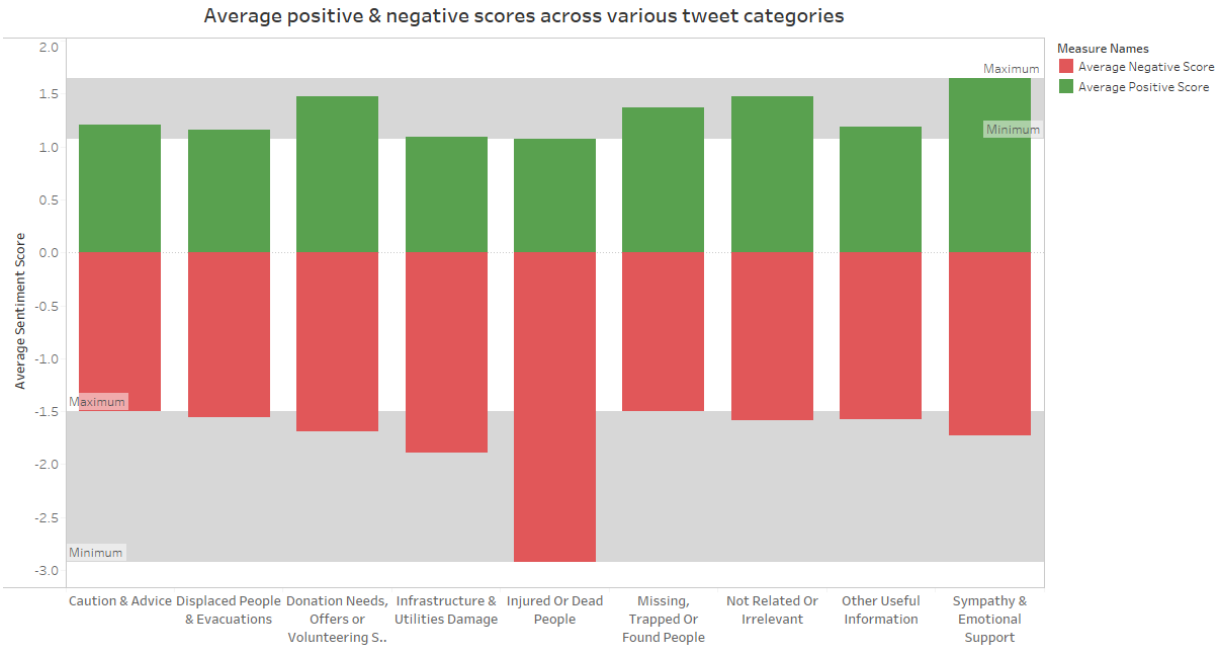


Figure 4.9: Average Sentiment Scores (Positive and Negative) across various tweet categories.

Useful Information categories have most positive tweet content.

Tweet Sentiment Scores: After eliminating the neutral tweets, the sentiment scores for positive and negative tweets was obtained and plotted as represented in figure 4.9. From the figure, it is clear that tweet category 'Injured or Dead People' has the most negative average score of -3.0. Other important classes with significantly negative scores are *Infrastructure & Utilities Damage* and *Donation Needs, Offers & Volunteering Services*. Also, tweet category *Sympathy & Emotional Support* has the most positive average score of +1.7 followed by *Not Related or Irrelevant, Donation Needs, Offers & Volunteering Services* and *Missing, Trapped & Found People*.

It is worth mentioning that the results obtained from AYLIEN are in complete alignment with the results obtained from the previous two techniques - using R and Senti-strength. The only difference is the fact that the tweet category *Donation Needs, Offers & Volunteering Services* is classified fairly equally for both the positive and negative sentiments. This could be due to the fact that the needs expressed by people

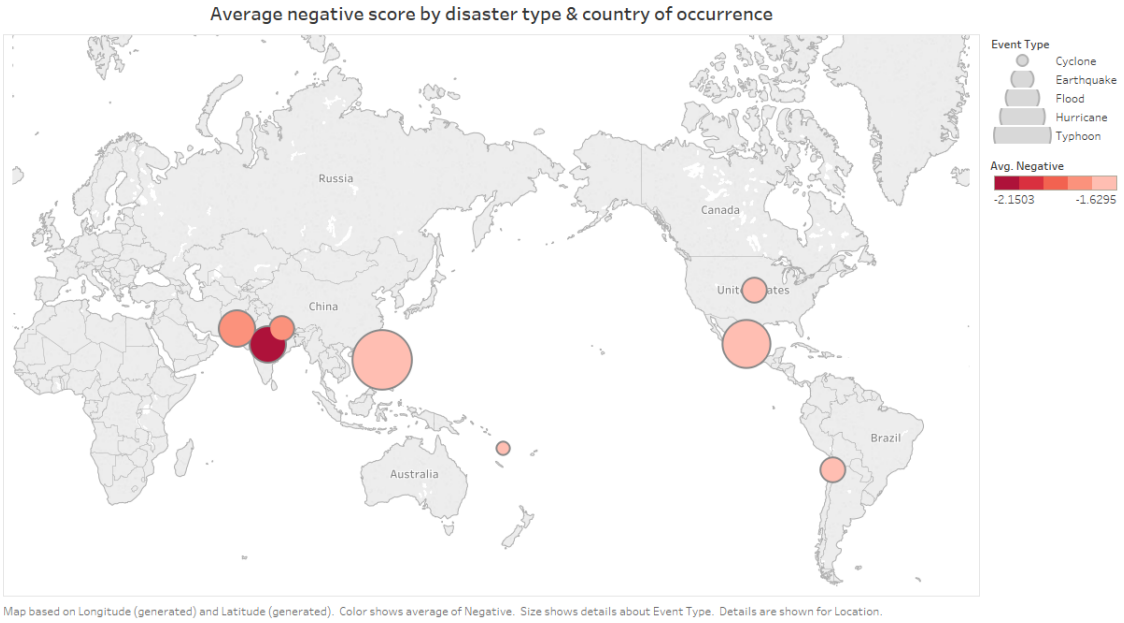


Figure 4.10: Average Negative Sentiment Scores.

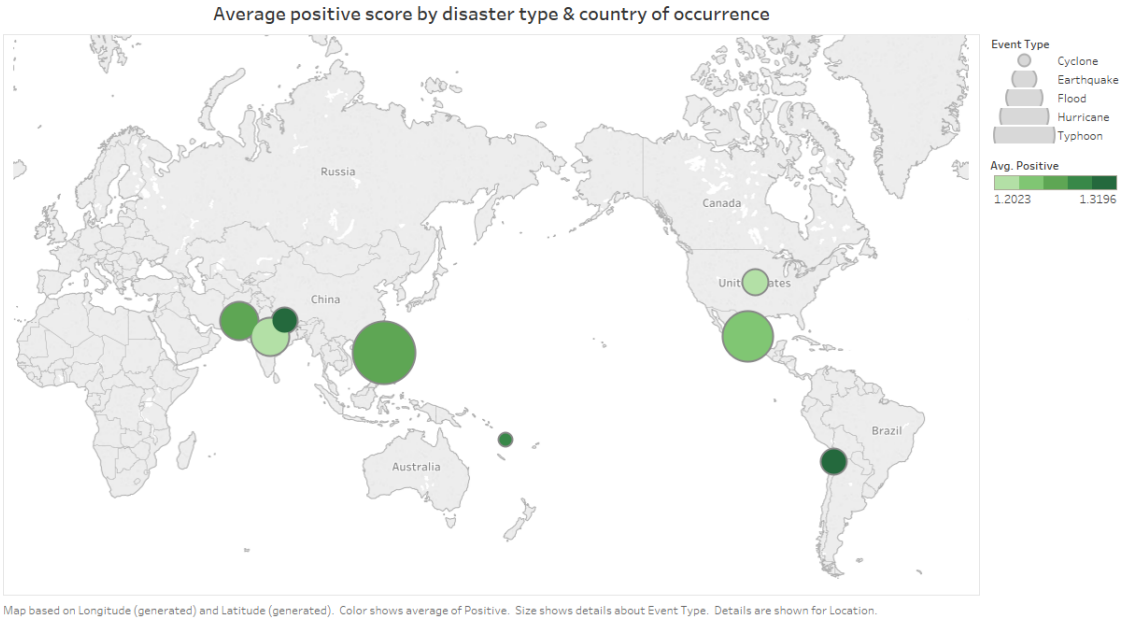


Figure 4.11: Average Positive Sentiment Scores.

can have mentions of panics and concerns classifying them as 'negative', while also expressing fulfilled needs, gratitude and thankfulness towards the services offered making them 'positive'.

Figures 4.10 and 4.11 represent average sentiment scores (Negative and Positive) for

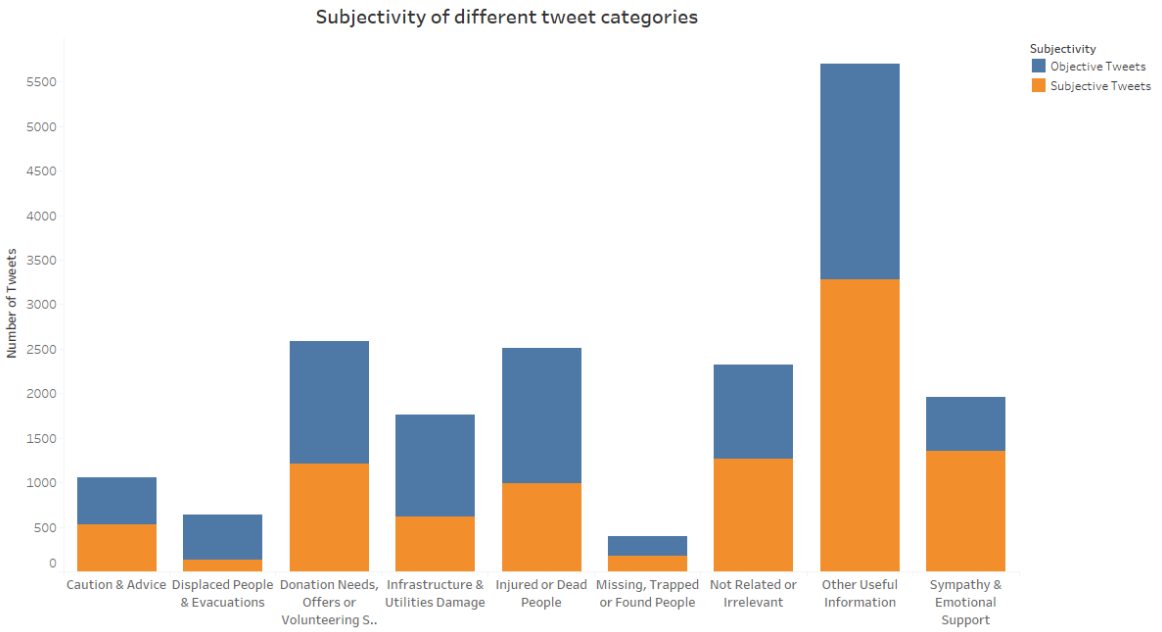


Figure 4.12: Subjectivity of various tweet categories.

all the disaster events geographically. The stepped color gradient shows the intensity (measure) of each score respectively. The different circle sizes correspond to different disaster events to distinguish them visually. An average high negative sentiment score is observed for India, Pakistan and Nepal where the earthquake and flood events took place. On the other hand of the spectrum, a high average positive sentiment score for Chile, Vanuatu, Nepal and a relatively low average positive scores for Pakistan and Philippines is observed.

Tweet Subjectivity: Figure 4.12 shows tweet subjectivity across various tweet categories. There are more objective than subjective tweets in all categories except for *Sympathy & Emotional Support, Other Useful Information* and *Not Related Or Irrelevant*. This makes sense in the real world because these categories are more about personal beliefs and opinions than describing facts. An objective perspective is one that is not influenced by emotions, opinions, or personal feelings - it is a perspective based on fact, on things quantifiable and measurable (example, missing or found people, infrastructure damage, displaced people, donation needs etc.) while a subjective perspective is based on personal feeling, emotion, aesthetics, etc.

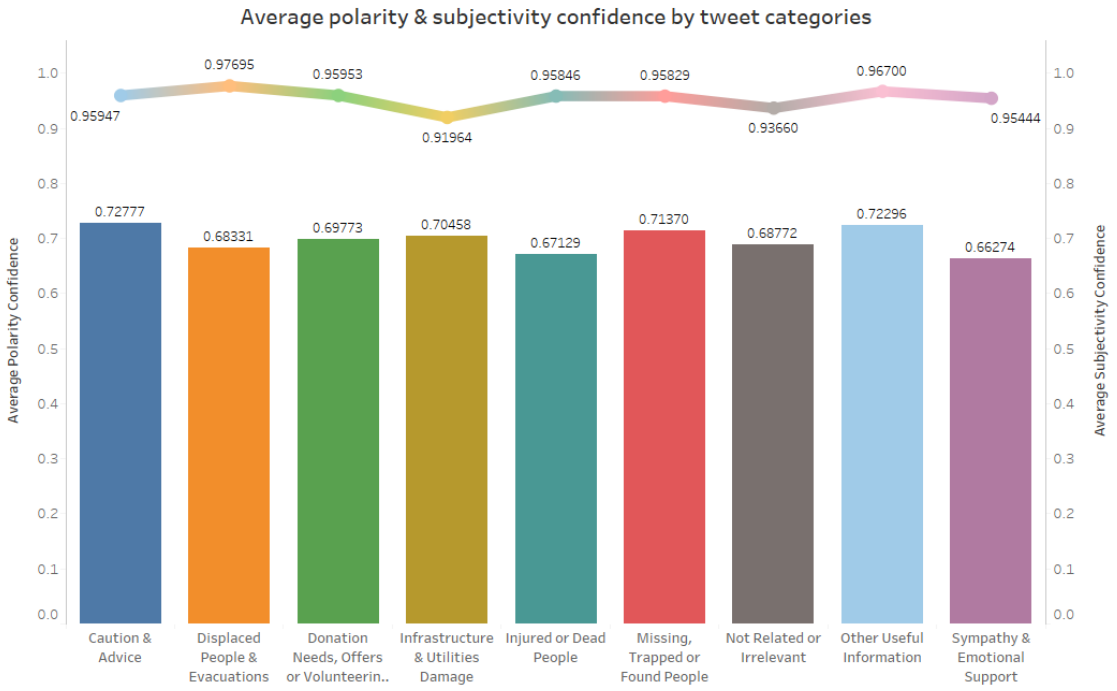


Figure 4.13: Comparing the average confidence levels for tweet polarity and tweet subjectivity.

Lastly, the confidence levels of classifying tweets based on their polarity and subjectivity is provided in this section. This is defined by a number lying between 0 and 1 denoting the accuracy of a tweet being classified as *Positive, Neutral or Negative* and *Objective or Subjective* across the 9 tweet categories. Figure 4.13 is a combination chart representing both the confidence levels. The vertical bars represent the average polarity confidence while the lines represent the average subjectivity confidence. The average tweet polarity confidence lies between a little over 65% to 72% while the average tweet subjectivity confidence ranges from 92% to 97%. This suggests that the accuracy of classifying a tweet into *subjective or objective* is much higher in comparison to classifying a tweet into *positive, negative or neutral*. This pattern holds true for all the 9 tweet categories in the dataset.

Using Rosette Text Analysis Extension- *Rosette Text Analytics Extension* is another popular multi-lingual text analytics solution that is used in conjunction with Rapid Miner to facilitate linguistic analysis, statistical modeling and machine learning

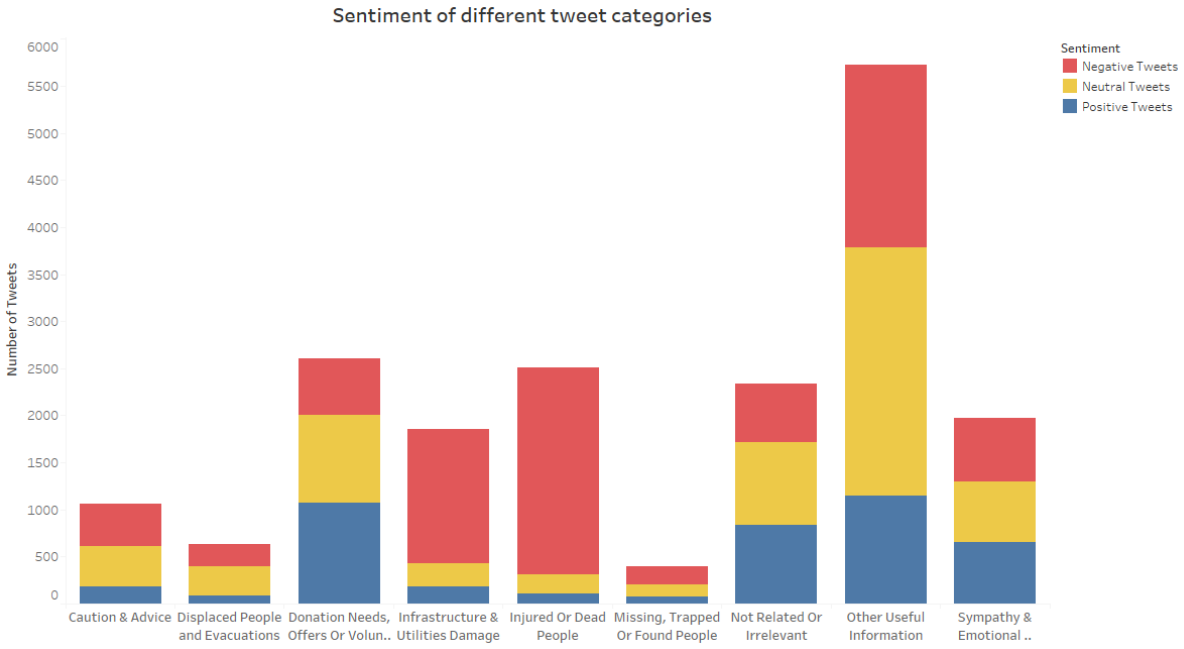


Figure 4.14: Sentiments across various tweet categories.

for generating actionable information and valuable insights from unstructured text. Again, this extension comes along with several operators that perform functions like *extracting and linking entities, analyzing entity sentiments, matching, translating and de-duplicating names, identifying language, analyzing sentiments, morphology, tokenization* and *transliteration*.

In order to perform sentiment analysis on twitter-disaster dataset, *Operator Analyze Sentiments* was used. The result obtained from this is shown in figure 4.14. From the figure 4.14, most negative sentiments are obtained for *Injured & Dead People, Infrastructure & Utilities Damage* and *Other Useful Information* tweet categories. One can clearly observe the difference in sentiment allocation with respect to AYLIEN extension by looking at figure 4.8 and comparing it with figure 4.14. AYLIEN has classified most tweets into a neutral class followed by negative and then positive, Rosette classified most tweets as negative followed by neutral and positive (this difference is elaborated in Appendix). Since, the results obtained from AYLIEN extension are used for tweet classification and not from Rosette, a decision was made not to go further into the differences between the two.

4.1.2 Extracting Named Entities from Disaster-Related Tweets

Even though posted at the time of disaster events, many tweets are simply off-topic and do not contain any relevant information. As outlined in the previous section that the information needs of various humanitarian organizations vary depending upon their coverage and level of operation, it is important to rapidly assess the crises situation in terms of named entities. Named-entities are basically the names of people, organizations and locations extracted from tweets and they provide ways to understand the disaster situation better. Example, it is easier to filter out tweets on the basis of specific locations (named mentions) than without them. Another advantage of extracting named-entities from disaster-related tweets is to establish trustworthiness of the published content (W. Wang & Stewart, 2015; W. Wang, 2014; Alam, Ofli, Imran, & Aupetit, 2018). Finding mentions of trusted government organizations or agencies inside a tweet makes the messages to be taken more seriously than if delivered by an unknown person or source. Similarly, mentions of specific street, park, bridge, highway or river help prepare for rapid disaster response by sending the rescue team directly to the right place.

Rosette Text Analytics Extension's operator *Extract Entities* was used to perform named-entity extraction from tweets. Figure 4.15 presents a subset of identified named-entities from tweets. The identified named entities ranged from Level 0 through Level 6, with Level 0 being at the top with broad entities and Level 6 extracting minute sub-entities. For the sake of simplicity, entities only up-to Level 2 were extracted. The green column highlights the number of occurrences of a given entity or sub-entity. An additional operator *Link Entities* can be used to link the extracted entities to a knowledge base of people, locations, and organizations by returning wiki-data ID to reveal more information about the entitys identity and resolve any ambiguity due to same names or duplicated entities. However, this lies beyond the scope of this work and is not included.

Extracting entities and sub-entities with Rosette (Level 0 through 2)

T1	T0	T2	
mamata banerjee	bengal chief minister	pm	1
manila	pm	jtwc	1
mayor	december	edwin olivarez	1
	mandaluyong	abalos	1
	tuesday	maribel eusebio	1
metro manila	metro manila	december	1
	pagasa	december	1
modi hm rajnath singh	pm	us	1
monday	philippines	december	1
mp	seema malhotra	heathrow	1
mrs	pam kulow ben davis	erdel	1
ms	eu	contributions	1
narendra modi	prime minister	kashmir	2
nawazsharif	pm	kashmir	1
		sher shah	1
nbl usa news	uk	kashmir	1
netanyahu	pm	pm	1
obama	president	prime minister	1
pagasa	metro manila	forecaster	1
pam nierlich	professor	jim gilbert chcdtv	1
philippines	god	f rubyph	1
pm	congress	jammukashmir economic times	1
		kashmir	1
	japan	shinzo abe fm fumio kishida	1
	metro manila	monday	1
	monday	pt tina sciabica	1
	sec	saturday	1
	typhoon rubyph flight bulletin	december	2
president	usa today	vanuatu	1

Figure 4.15: Named-Entity Extraction using Rosette Text Analytics.

4.1.3 Contextual Categorization of Disaster-Related Tweets

While machine learning classifiers can be trained to classify tweets into into one of the 9 humanitarian categories as identified by the Crowd-Flower platform, it can be useful to classify tweets based on the contextual knowledge structures. This is extremely helpful for discovering popular, trending topics from the tweets and tapping the web to obtain more precise and relevant information about those identified topics. As mentioned in (Alam, Ofli, Imran, & Aupetit, 2018), there are often different topics of discussion before, during and after disaster events on Twitter and its convenient to generate topics from large amounts of textual information. By performing contextual categorization, one can understand and summarize the textual content and discover hidden topics of discussions.

Rosette Text Analytics operator *Categorize* was used to obtain the list of Tier 1 categories associated with tweets. The Interactive Advertising Bureau (IAB) provides a list of Tier 1 and Tier 2 contextual categories primarily optimizing digital adver-

Comparing manually annotated tweet categories with Tier-1 contextual categories as per QAG Taxonomy

	Caution & Advice	Displaced People & Evacuations	Donation Needs, Offers or Volunteering Services	Infrastructure & Utilities Damage	Injured or Dead People	Missing, Trapped or Found People	Not Related or Irrelevant	Other Useful Information	Sympathy & Emotional Support
ARTS_AND_ENTERTAINMENT	16	11	30	32	67	14	93	123	32
AUTOMOTIVE	12	4	13	13	12	3	27	43	9
BUSINESS	16	9	66	51	93	10	57	148	23
CAREERS	7	4	22	3	1	1	13	35	2
EDUCATION	9	2	28	7	6	10	29	50	16
FAMILY_AND_PARENTING	8	12	68	3	1	19	36	33	34
FOOD_AND_DRINK	11	3	12	9	2	1	28	38	7
HEALTH_AND_FITNESS	29	32	203	23	64	14	55	148	88
HOBBIES_AND_INTERESTS	11	1	22	20	25	4	36	58	5
HOME_AND_GARDEN	6	9	20	28	9	2	20	42	8
LAW_GOVERNMENT_AND_POLITICS	9	11	17	15	28	1	30	91	13
PERSONAL_FINANCE	8	3	33	15	18	3	17	41	6
PETS	11	2	9	7	3	3	11	31	12
REAL_ESTATE	4	10	14	53	28	2	14	38	2
RELIGION_AND_SPIRITUALITY	8	10	67	24	273	12	70	121	340
SCIENCE	124	39	75	135	150	9	81	451	41
SOCIETY	4	3	17	17	15	8	37	26	27
SPORTS	33	17	139	36	49	12	93	203	56
STYLE_AND_FASHION	7	4	9	16	10	2	29	32	7
TECHNOLOGY_AND_COMPUTING	23	22	116	82	84	10	79	217	32
TRAVEL	67	45	58	151	72	21	73	323	24

Figure 4.16: IAB Contextual categorization of disaster tweets.

tising and marketing campaigns. The *Categorize* operator returns the most likely Tier 1 Category listed under IAS as per quality assurance guidelines in the form of a data table. The taxonomy of contextual categories as defined by IAB can be found in the link: https://www.iab.com/guidelines/iab-quality-assurance-guidelines-qag-taxonomy/iab-quality-assurance-guidelines-qag-taxonomy/.

From figure 4.16, *Categorize* operator returned around 20 contextual categories relevant to the disaster tweets. Tweets were then compared for IAB contextual categories and Crowd-Flower humanitarian categories. Most tweets belonging to *Other Useful Information* are classified under the topics "Science" and "Travel". Tweet categories *Injured & Dead People* and *Sympathy & Emotional Support* fall under the topic "Religion & Spirituality". This information sheds light on key topics of discussion during the disaster events and hence offers valuable insights into the data.

While the multi-dimensional content analyses of tweets in terms of understanding sentiments of people, extracting useful named-entities and finding topics of discussion from twitter feeds does help in generating more situational awareness, the main task is to automatically classify the collected tweets into predefined humanitarian categories. It is important to note that the output generated from sentiment analyses was used

for extending features used for text classification as presented in Section 4.2.

4.2 Classification of Tweet Text Using WEKA

Text Classification is an important area in machine learning which finds many applications in spam filtering, content tagging, sentiment analysis, opinion and intent mining, content enrichment etc. It basically groups and sorts natural language text of the same type into one of the predefined class labels. In the current exercise, text classification of tweets was performed using Waikato Environment for Knowledge Analysis (WEKA) which is an open-source data analyses program shared by machine learning group at the University of Waikato in New Zealand. WEKA has a classic repository of machine learning algorithms.

Working with WEKA

WEKA works both with CSV and ARFF file formats however, ARFF file format is easier to understand and model in machine learning classifiers available in WEKA. ARFF stands for Attribute-Relation File Format. It is an ASCII text file describing a list of instances which share a common set of attributes. This file format is less memory intensive, faster and better for data analyses because it includes metadata about column headers. The tweet text that needs to be classified resides in *Data* section while the variables corresponding to different columns in the dataset reside in *Attribute* section of the ARFF file. As the study aims to identify the impact of adding sentiment features to the original dataset to observe the changes in classification performance, the additional features including text translation, sentiment scores, sentiment polarity, sentiment subjectivity and their confidence levels will form the *Attribute* section in the ARFF file. Data can be represented in boolean, real numbers, single words, phrases, or other types depending on the dataset, to find a configuration for the machine learning classifiers to maximize their performance as their performance is highly dependent on the data representation. The data preparation performed on disaster-related tweets to get them ready for training the machine learning classifiers

is presented in the following section.

4.2.1 Data Preprocessing

First of all, the original dataset in CSV file format was converted to ARFF format. This was done using an operator called *Write ARFF* in *Data Access Package* of Rapid Miner software. As mentioned in the previous chapter, dataset was already cleaned for hash-tags, symbols, numerics, stop-words, slang words and twitter handles etc. using R, there was no need to perform additional pre-processing on the tweet text in WEKA. Also, all the tweets were translated (into appropriate sentence forms) with the help of Senti-Strength. An example of translated tweet is presented here:

Original Tweet text : f know spell pam right idk cyclonnnnnnne

Translated Tweet text: know spell pam right i don't know cyclone

Tweet translation removes duplicated and extended spellings, it also changes common short-hands like IDK and IMO to their full forms 'I don't know' and 'In my opinion' respectively. This eventually leads to better word features to be used in the classifiers.

The ARFF file was imported into WEKA via *WEKA Explorer Window*. All the attributes of the file were presented in the form of a list depicting the type of variables (nominal, numeric, string etc.) along with the translated tweet text and output categories. Tweet text was identified to be of type 'Nominal'. This type can't be used directly in machine classifiers, as a result of which it had to be translated from 'Nominal' to type 'String'. String type can then be manipulated/utilized to extract feature vectors as presented in the next section.

4.2.2 Preparation of Feature Vectors

After initial data preprocessing and data preparation, the next step is to transform the raw data into feature vectors. In text classification, each term, phrase or character can be represented as a feature. A feature is a measurable property about a tweet text in the dataset. The reason to vectorize the data strings i.e. to convert sequences of text

into attributes with number and categorical values is performed in the hope of finding the best feature vector for a learning classifier. The feature vectors are generated from the existing data by transforming the raw data into machine readable units, called, 'feature nuggets' which essentially carry significant information about the data and its characteristics. These feature vectors are able to 'learn' certain aspects about the dataset that will be utilized during machine classification in later stages. There are many different ways to generate feature vectors which include: *Count Vectors, Term-Frequency Inverse-Document-Frequency (TF-IDF), Word Embeddings, Text or Natural Language Processing based* and *Topic Model* based features.

Count Vectors as Features: In this case, the dataset is represented as a matrix where every row is a document from the corpus and every column is a term from the corpus, while every cell of the matrix is a frequency count of a particular term in a particular document.

TF-IDF as Features: As the name suggests, there are two computations involved, one is normalized term frequency and the other is logarithm of the number of documents in the corpus divided by documents with a specific term. This score calculates the relative importance of a term in the document as well as in the overall corpus. TF-IDF feature vectors can be generated at different levels of input token sequence. This usually can be in the form of words, characters/alphabets and n-grams. While character and word level TF-IDF works on alphabets and words (terms) in the corpus respectively, n-grams involve n terms which could be uni-gram, bi-gram, tri-gram or their combinations in the corpus.

Word Embeddings as Features: This is a dense representation scheme where the position of a word within the vector space is learned from the text. This representation is based on the location of the words surrounding a particular word at the time of its usage. Word-embeddings can be generated from the input corpus as well as from pre-trained word embeddings like Glove, Word2Vec, Facebook vectors on 90 languages, Wikipedia Dump, FastText, Skip-Gram, CBOW etc.

Text or Natural Language Processing Based Features: These include simple word, character and average word density of the documents. These can be highly specific to the problem and can include punctuation counts as well as upper and lower case counts in the documents. Frequency distribution of parts-of-speech tags like noun, verb, adjective, adverb, pronoun can also be used depending upon the case.

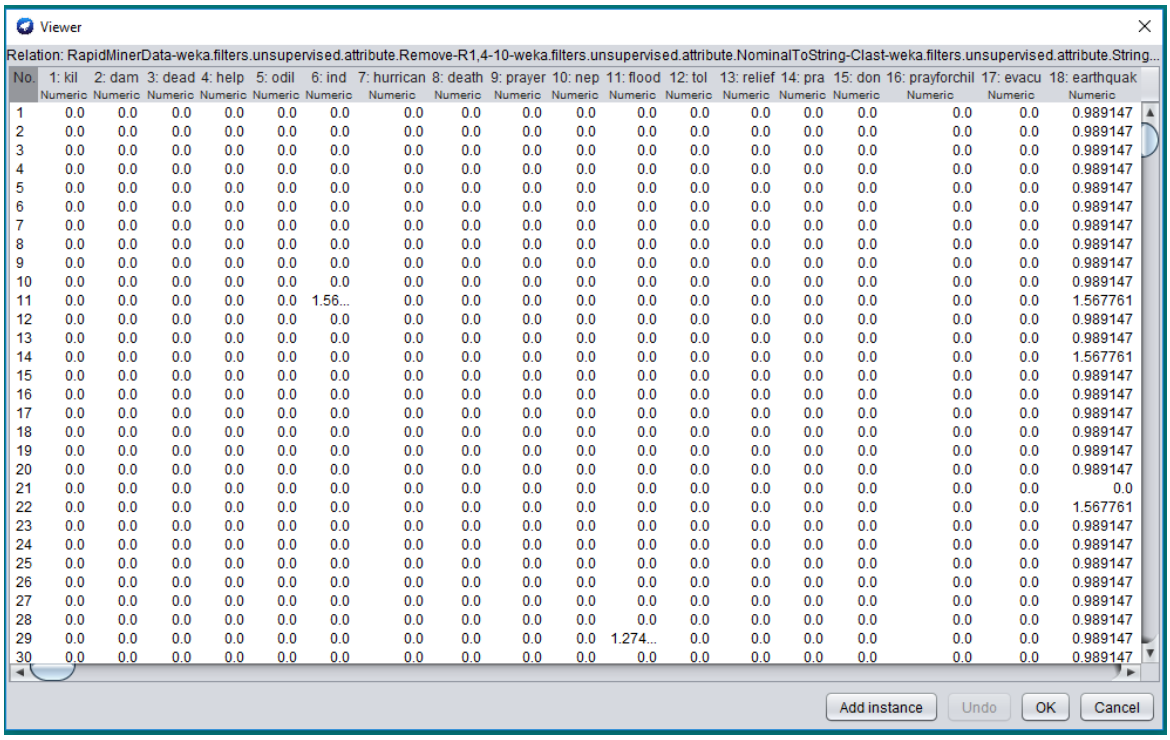
Topic Models as Features: This technique involves topic identification from a given collection of documents. Each topic is represented as a distribution over words while each document is represented as a distribution over topics. It is the probability distribution over words as defined by topics which provides an idea about the themes contained in a document. Latent Dirichlet Allocation (LDA) is the most commonly used topic modeling technique.

In the current work, the variable 'Translation' changed from type nominal to type string during pre-processing was then converted into word feature vectors for training the classifiers. *String to Word Vector* operator in WEKA was used to perform this task. This operator provides ways to choose how to represent a tweet text as a document vector. Lovins Stemmer was used to stem the words and 500 words were kept as word features. High term frequency and low document frequency were enabled using a filter in *unsupervised attribute selection* settings. High term frequency and low document frequency refers to those words that rarely appear in the document collection, but occur frequently in particular documents. The TF-IDF scores for every word feature can be seen in the 'Edit' window as shown in figure 4.17.

The word feature vectors were generated in WEKA using TF-IDF and various tokenization schemes, 8 in particular; alphabetic, word, uni-gram, bi-gram, tri-gram, uni+bi-gram, bi+tr-gram and uni+bi+tri-gram, to see if the performance of classifiers differ based on the token-levels in use.

4.2.3 Feature Engineering

Machine learning algorithms work by making predictions about the class a given data instance should belong to on the basis of various attributes available in the dataset.



No.	1: kil	2: dam	3: dead	4: help	5: odil	6: ind	7: hurrican	8: death	9: prayer	10: nep	11: flood	12: tol	13: relief	14: pra	15: don	16: prayforchil	17: evacu	18: earthquak
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
11	0.0	0.0	0.0	0.0	0.0	1.56...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.567761
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.567761
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.567761
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
29	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.274...	0.0	0.0	0.0	0.0	0.0	0.0	0.989147
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.989147

Figure 4.17: TF-IDF Scores of Word Vector Features.

Some of these attributes are extremely relevant to making predictions while others simply increase model complexity but do not offer any valuable information, such features/attributes should be eliminated before training the classifiers. The process of selecting valuable features and eliminating the non-essentials before being fed into a model is called feature engineering. Specifically, feature engineering is not limited to feature selection but might also include things like data transformation, dimensionality reduction, principal component analyses etc. Feature selection makes it easier to train and apply a classifier by decreasing the vocabulary size of the word features in use. Also, it enhances classification accuracy by removing noisy features.

There are several ways to perform and improve feature selection for text classification (Dasgupta et al., 2007) which predominantly include: document frequency (DF), information gain (IG), chi squared, mutual information and sampling (subspace sampling, weight-based sampling, uniform sampling). Performed on three different datasets of *TechTC-100, 20-Newsgroups* and *Reuters-RCV2*, it was consistently observed that Information Gain method produced the maximum accuracy levels for

different classifiers followed by document frequency while other techniques were comparatively lower. Also, as presented in (Jovic et al., 2015), even though there is no silver bullet method, filters based on information theory and wrappers based on greedy stepwise approaches seem to offer best results. The same article mentions that while combination of different feature selection techniques can be used to improve classifier accuracy, it comes at the expense of higher feature correlation, which in itself, wastes the purpose of feature selection and dimensionality reduction problem in text analytics. There is plethora of ongoing research in this area which focuses on optimizing the efficiency and accuracy of feature subset search strategy by combining best filter and wrapper approaches. However, most research tends to focus on small number of datasets; larger comparative studies should be pursued in order to have more reliable results.

Feature selection in WEKA can be performed using various techniques accessible via *WEKA Explorer* as shown in figure 4.18.

The 'Select Attribute' tab in WEKA Explorer provides access to different feature selection methods. The task of feature selection in WEKA is divided into two parts: *Attribute Evaluator* and *Search Method* and each corresponds to multiple techniques from which to choose. The *attribute evaluator* method evaluates each attribute in the dataset in the context of output variable while the *search method* chooses a list of features from various combinations of attributes which maximize model performance. The three main feature selection methods in WEKA are provided here:

Correlation Based Feature Selection

This technique selects the most relevant features in the dataset by generating Pearson's Correlation Scores for every attribute. The correlation is computed between each attribute and the output class variable; this technique picks up only those features which have relatively higher correlation values and neglects the rest. This technique is not relevant in the context of disaster-twitter dataset, for the experiment. This is so because the textual data (tweets) bear very low to no correlation with the output class, this feature selection method is not appropriate and hence was discarded.

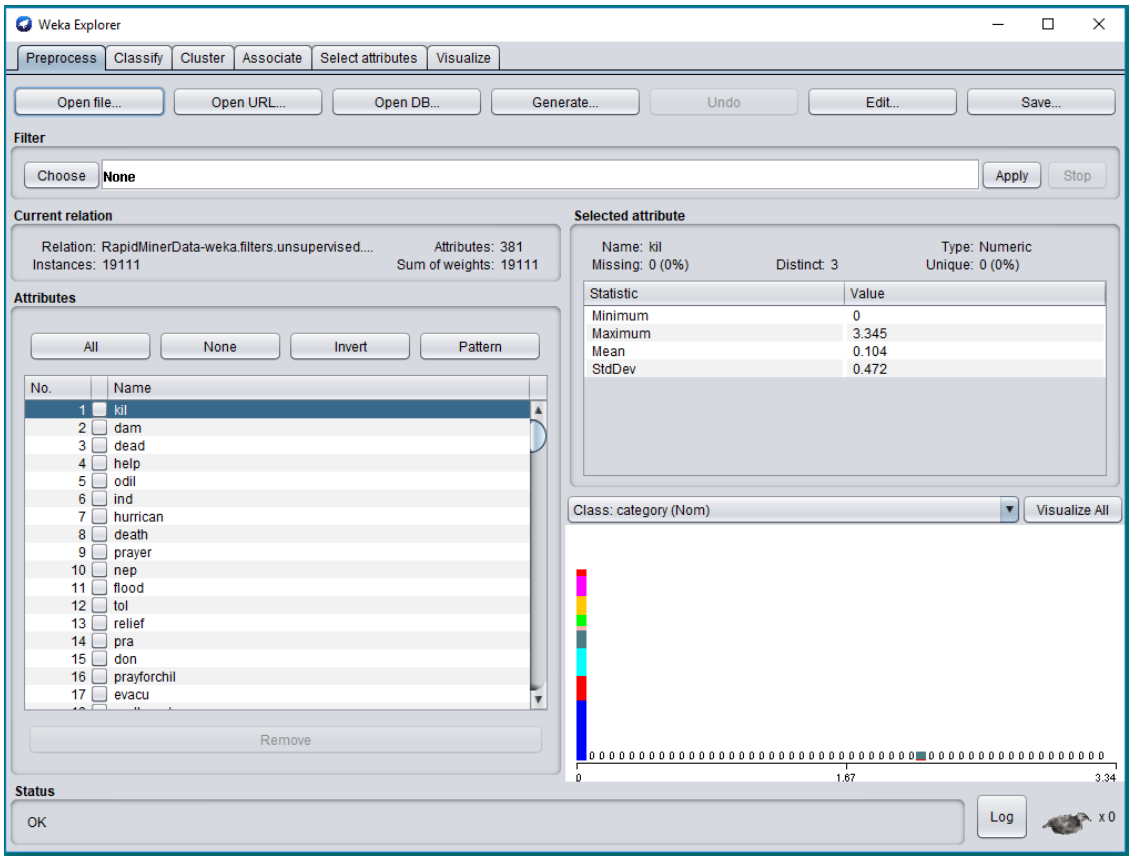


Figure 4.18: Feature Selection in WEKA

Learner Based Feature Selection

This technique works on the principle that a machine learning classifier performs differently with different subsets of selected attributes. The final selection is based on the subset that yields best classification performance. A decision tree method is the most preferable model for selecting the best subset since it provides rules on which to divide the data. WEKA uses Wrapper Subset Evaluation and Best First Search Method to perform this task.

Information Gain Based Feature Selection

This is another popular technique for selecting features based on the level of informativeness each feature has. 'Information Gain' is a metric used to calculate the entropy of each attribute for the output class. In this case, 0 shows no information

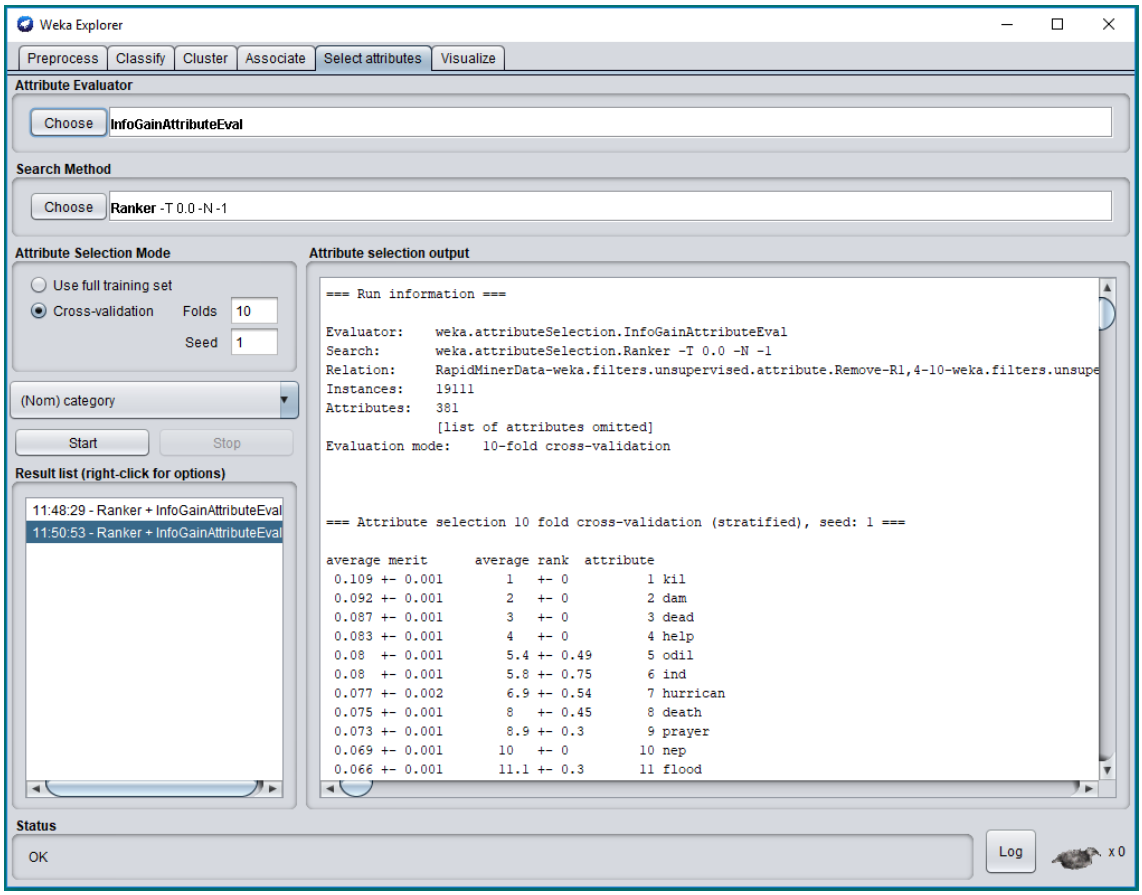


Figure 4.19: Ranking word features based on their informativeness (Information Gain)

while 1 shows maximum information. The attributes contributing more information (hence with a higher value of information gain) are selected while the remaining are rejected. WEKA performs a ranking of variables on the basis of their information gain to select the most informative features. This technique was employed in the current experiment. The arbitrary cut-off value of 0.05 was used as the lower threshold and the word vectors were ranked in decreasing order of their informativeness as shown in figure 4.19.

It is important to understand that each method can result in slightly different but mostly overlapping features. While it is not straightforward to determine correctly about which features to use in the predictive models, it is a good idea to try different approaches and compare their performance. Different subsets can be used to train a new model that can be compared against a baseline to evaluate the performance.

4.2.4 Initial Modeling

The dataset was split into training and testing sets respectively, keeping the ratio of 60 to 40. As mentioned in the previous chapters, there is a varying number of tweets in each of the 9 humanitarian categories, ranging from several hundreds to several thousands, it was important to correctly segment the data in a way that 60 to 40 ratio per category was preserved, this was done by manually selecting the required number of tweets from each category leaving out the rest. The selection of tweets was random and not stratified. Out of a total 19,000 tweets, 11,000 were used as training set and remaining 8,000 were used for testing. There were two problems that were addressed before applying the classification algorithms on the dataset. These were:

Cost-Effectiveness Learning: This defines how costly it is for a classifier to misclassify a classification task; specifically telling the classifier to be n times (n is user-defined) more careful with false positives (including something that shouldn't have been included) than false negatives (forgetting to include something which should have been included).

Sampling: This is done to adjust for class imbalance. Since the twitter-disaster dataset had class imbalance, a sampling technique was used. This was done by over-sampling the minority class and under sampling the majority class in WEKA.

Tweet Classification Results Using Different Tokenizer Schemes

The different machine learning classifiers were initially modelled using different token-sequences (lengths) to observe any difference in classification performance. The performance of machine learning classifiers is measured in terms of weighted average precision, recall, F-score as well % accuracy of correctly classified text.

Different tokenization schemes used were:

- Alphabetic Tokenizer: This tokenization algorithm generates tokens only from contiguous alphabetic sequences, in the text strings.
- Word Tokenizer: This tokenization algorithm generates tokens based on words and a set of delimiter characters (example carriage-return, line-feed, tab etc.)

- Uni-gram Tokenizer: This tokenization algorithm splits a string into a uni-gram or a single word in the corpus.
- Bi-gram Tokenizer: This tokenization algorithm splits a string into a bi-gram or a word-pair in the corpus.
- Tri-gram Tokenizer: This tokenization algorithm splits a string into a tri-gram or a word-triplet in the corpus.
- Uni+Bi-gram Tokenizer: This tokenization algorithm performs string split based on a combination of a uni-gram and a bi-gram.
- Bi+Tri-gram Tokenizer: This tokenization algorithm performs string split based on a combination of a bi-gram and a tri-gram.
- Uni+Bi+Tri-gram Tokenizer: This tokenization algorithm performs string split based on a combination of a uni-gram, a bi-gram and a tri-gram.

While bi-grams and their combinations with other n-grams are preferred over single words when used as features in a Naive Bayes Classifier for next word predictions and text categorization on certain types of corpora including *Yahoo-Science* and *Reuters-21578* as mentioned in (Tan, Wang, & Lee, 2002), there is no significant evidence that it definitely improves the quality of text classification. In a paper by (Bekkerman & Allan, 2003), it was concluded that bigrams do not produce dramatically accurate results when used in the context of unrestricted text categorization. They could however prove to be superior to uni-grams in domains with very limited availability of lexicons. Similarly, (S. Wang & Manning, 2010) suggest that the benefits from using higher variants of n-grams (bi-gram and above) depend on the task and are limited in case of topical text classification but they produce significantly better results for sentiment classification.

Table 4.1 provides an entire list of classification performance using different token-levels. The ones shown in bold have highest accuracy of correctly classified text and the ones shown in blue (and bold) have an accuracy of 70% or more. The different token-levels are separated by means of a horizontal line in the full table. From the

table, the performance of alphabetic, word and uni-gram tokenizers is the highest and is very similar to each other. The performance however varies from classifier to classifier but a consistently high value is observed for these token levels than others. Bi-grams have a lower classification accuracy whereas tri-grams offer worst performance. A combination of uni and bi-grams is better in accuracy than bi and tri-grams. Also, there is much better classification accuracy by combining uni, bi and tri grams than handling bigrams and trigrams singularly. It is interesting to note that the precision values for bi-grams and tri-grams are higher but there is a sharp decline in recall and F-score. This boils down to the fact that as the length of the n-grams increase, the number of times such a sequence will be seen in the corpus actually decreases. This problem is referred to as *Data Sparsity Problem*. This causes poor generalizability and hence low recall in the absence of sufficient training data for such events. Thus, with a large token-size occurring much less number of times in the dataset, lower-order n-gram models provide better results as observed in our case.

Table 4.1: Classification Performance using Different Tokens

Using Alphabetic Token Sequence					
S.No.	Classifier Name	Precision	Recall	F-Score	Accuracy
1	Naive Bayes	0.581	0.540	0.537	53.9742 %
2	Naive Bayes Multinomial	0.626	0.603	0.599	60.2742 %
3	Bayes Net	0.648	0.645	0.643	64.4864 %
4	Multinomial LR	0.695	0.682	0.688	67.3531 %
5	Multi Layer Perceptron	0.661	0.659	0.659	66.5943 %
6	Deep CNN	0.681	0.672	0.676	68.1142 %
7	Simple LR	0.717	0.715	0.711	**71.4516 %**
8	Seq. Minimal Optimization	0.723	0.718	0.714	**71.7859 %**
9	Filtered Classifier	0.778	0.777	0.774	**77.7406 %**
10	Decision Table	0.697	0.655	0.639	65.4806 %
11	J-48	0.698	0.700	0.695	**70.0225 %**
12	REP-Tree	0.685	0.678	0.669	67.8405 %

Classification Performance using Different Tokens - Continuation of Table 4.1					
S.No.	Classifier Name	Precision	Recall	F-Measure	Accuracy
13	Random Tree	0.600	0.600	0.599	60.0126 %
14	Random Forest	0.696	0.698	0.693	69.7661 %
15	Rotation Forest	0.808	0.805	0.802	**80.4824 %**
Using Word Token Sequence					
1	Naive Bayes	0.582	0.542	0.539	54.1782 %
2	Naive Bayes Multinomial	0.625	0.602	0.599	60.2114 %
3	Bayes Net	0.648	0.645	0.643	64.4655 %
4	Multinomial LR	0.713	0.719	0.713	**70.4899 %**
5	Multi Layer Perceptron	0.685	0.687	0.672	67.6317 %
6	Deep CNN	0.631	0.614	0.599	63.3429 %
7	Simple LR	0.718	0.698	0.712	**70.7413 %**
8	Seq. Minimal Optimization	0.721	0.719	0.712	**71.3316 %**
9	Filtered Classifier	0.729	0.735	0.721	**73.6783 %**
10	Decision Table	0.703	0.696	0.688	69.5987 %
11	J-48	0.699	0.701	0.696	**70.0905 %**
12	REP-Tree	0.684	0.679	0.670	67.8981 %
13	Random Tree	0.602	0.602	0.601	60.2219 %
14	Random Forest	0.694	0.696	0.691	69.5934 %
15	Rotation Forest	0.756	0.762	0.772	**76.3314 %**
Using Uni-gram Token Sequence					
1	Naive Bayes	0.590	0.547	0.532	54.8766 %
2	Naive Bayes Multinomial	0.629	0.612	0.578	60.1232 %
3	Bayes Net	0.649	0.647	0.641	64.4899 %
4	Multinomial LR	0.717	0.713	0.710	**70.7867 %**
5	Multi Layer Perceptron	0.645	0.637	0.611	61.5644 %
6	Deep CNN	0.647	0.632	0.621	64.7656 %
7	Simple LR	0.765	0.732	0.721	**73.5654 %**
8	Seq. Minimal Optimization	0.726	0.721	0.716	**71.6543 %**

Classification Performance using Different Tokens - Continuation of Table 4.1					
S.No.	Classifier Name	Precision	Recall	F-Measure	Accuracy
9	Filtered Classifier	0.779	0.765	0.754	**76.1432 %**
10	Decision Table	0.713	0.701	0.698	**70.5659 %**
11	J-48	0.712	0.711	0.708	**71.1238 %**
12	REP-Tree	0.687	0.677	0.674	67.5438 %
13	Random Tree	0.605	0.603	0.600	60.1227 %
14	Random Forest	0.695	0.697	0.692	69.6562 %
15	Rotation Forest	0.752	0.741	0.762	**75.1143 %**
Using Bi-gram Token Sequence					
1	Naive Bayes	0.512	0.399	0.317	39.9299 %
2	Naive Bayes Multinomial	0.486	0.445	0.437	44.4665 %
3	Bayes Net	0.570	0.482	0.444	48.1974 %
4	Multinomial LR	0.519	0.511	0.508	51.0171 %
5	Multi Layer Perceptron	0.491	0.487	0.417	47.1187 %
6	Deep CNN	0.519	0.481	0.476	48.9120 %
7	Simple LR	0.562	0.551	0.461	51.2189 %
8	Seq. Minimal Optimization	0.569	0.518	0.498	52.8797 %
9	Filtered Classifier	0.619	0.601	0.412	**54.1176 %**
10	Decision Table	0.576	0.541	0.512	52.7652 %
11	J-48	0.531	0.529	0.501	**53.1251 %**
12	REP-Tree	0.685	0.591	0.576	**57.8756 %**
13	Random Tree	0.600	0.600	0.599	**60.0126 %**
14	Random Forest	0.584	0.493	0.457	49.3224 %
15	Rotation Forest	0.543	0.524	0.516	52.8762 %
Using Tri-gram Token Sequence					
1	Naive Bayes	0.722	0.350	0.298	34.9642 %
2	Naive Bayes Multinomial	0.525	0.395	0.331	39.4746 %
3	Bayes Net	0.660	0.404	0.321	40.3851 %
4	Multinomial LR	0.609	0.510	0.482	**51.3231 %**

Classification Performance using Different Tokens - Continuation of Table 4.1					
S.No.	Classifier Name	Precision	Recall	F-Measure	Accuracy
5	Multi Layer Perceptron	0.601	0.495	0.501	**49.5909 %**
6	Deep CNN	0.612	0.509	0.510	**51.0192 %**
7	Simple LR	0.771	0.372	0.278	39.4516 %
8	Seq. Minimal Optimization	0.612	0.413	0.401	41.7859 %
9	Filtered Classifier	0.718	0.492	0.371	43.2271 %
10	Decision Table	0.697	0.501	0.239	40.4806 %
11	J-48	0.698	0.500	0.369	44.0225 %
12	REP-Tree	0.660	0.410	0.329	41.0183 %
13	Random Tree	0.631	0.415	0.432	42.0346 %
14	Random Forest	0.659	0.416	0.338	41.5729 %
15	Rotation Forest	0.612	0.513	0.491	**52.2217 %**
Using Uni+Bi-gram Token Sequence					
1	Naive Bayes	0.581	0.537	0.539	53.723 %
2	Naive Bayes Multinomial	0.611	0.577	0.576	57.7259 %
3	Bayes Net	0.632	0.617	0.617	61.6556 %
4	Multinomial LR	0.529	0.519	0.504	52.8718 %
5	Multi Layer Perceptron	0.518	0.510	0.471	51.7611 %
6	Deep CNN	0.513	0.521	0.510	53.1781 %
7	Simple LR	0.616	0.610	0.572	**60.0916 %**
8	Seq. Minimal Optimization	0.652	0.691	0.682	**67.7662 %**
9	Filtered Classifier	0.623	0.576	0.532	**59.1212 %**
10	Decision Table	0.592	0.589	0.572	55.1675 %
11	J-48	0.591	0.611	0.595	59.8877 %
12	REP-Tree	0.691	0.561	0.519	57.1178 %
13	Random Tree	0.570	0.569	0.568	56.9149 %
14	Random Forest	0.678	0.680	0.674	**69.7661 %**
15	Rotation Forest	0.598	0.601	0.542	**59.8891 %**

Classification Performance using Different Tokens - Continuation of Table 4.1					
S.No.	Classifier Name	Precision	Recall	F-Measure	Accuracy
Using Bi+Tri-gram Token Sequence					
1	Naive Bayes	0.513	0.394	0.308	39.3909 %
2	Naive Bayes Multinomial	0.470	0.420	0.413	42.0012 %
3	Bayes Net	0.564	0.428	0.439	42.8344 %
4	Multinomial LR	0.509	0.510	0.512	**51.3231 %**
5	Multi Layer Perceptron	0.501	0.459	0.496	**49.5909 %**
6	Deep CNN	0.512	0.509	0.510	**51.0192 %**
7	Simple LR	0.601	0.418	0.491	**51.4221 %**
8	Seq. Minimal Optimization	0.523	0.418	0.471	43.1819 %
9	Filtered Classifier	0.601	0.493	0.477	47.6416 %
10	Decision Table	0.514	0.411	0.416	43.1872 %
11	J-48	0.569	0.501	0.495	45.0211 %
12	REP-Tree	0.498	0.478	0.449	45.8405 %
13	Random Tree	0.556	0.470	0.425	46.9572 %
14	Random Forest	0.564	0.475	0.432	47.5381 %
15	Rotation Forest	0.605	0.495	0.478	**48.1145 %**
Using Uni+Bi+Tri-gram Token Sequence					
1	Naive Bayes	0.579	0.537	0.539	53.6602 %
2	Naive Bayes Multinomial	0.607	0.570	0.569	57.0143 %
3	Bayes Net	0.627	0.608	0.607	60.7765 %
4	Multinomial LR	0.609	0.610	0.612	61.3231 %
5	Multi Layer Perceptron	0.601	0.559	0.598	60.1217 %
6	Deep CNN	0.622	0.617	0.592	61.0017 %
7	Simple LR	0.691	0.672	0.681	**68.1342 %**
8	Seq. Minimal Optimization	0.672	0.682	0.682	**68.7859 %**
9	Filtered Classifier	0.687	0.690	0.685	**68.9864 %**
10	Decision Table	0.697	0.655	0.639	65.4806 %
11	J-48	0.684	0.687	0.682	**68.7457 %**

Classification Performance using Different Tokens - Continuation of Table 4.1					
S.No.	Classifier Name	Precision	Recall	F-Measure	Accuracy
12	REP-Tree	0.676	0.668	0.660	66.8411 %
13	Random Tree	0.595	0.594	0.593	59.4161 %
14	Random Forest	0.683	0.685	0.680	**68.5469 %**
15	Rotation Forest	0.663	0.691	0.659	**67.1528 %**

Selecting the best performing tokenizer scheme

Table 4.2 summarizes the results of the long table 4.1 by averaging out the overall performance using all the classifiers. From table 4.2, it is seen that percentage of correctly classified instances is the highest in case of alphabetic (A), word (W) and uni-gram (U) tokens followed by the combination uni+bi+tri-gram (U+B+T) and uni+bi-gram (U+B). Performance drops significantly for bi-gram (B), bi+tri-gram (B+T) and tri-gram (T) sequences. This possibly happens because higher order n-grams do not occur as frequently in the dataset as the lower order. High precision, recall and F-score is observed for alphabetic, word and uni-gram tokenizers. Lowest recall and F-score is observed for tri and bi-grams.

Measure	A	W	U	B	T	U+B	B+T	U+B+T
Accuracy	67.7%	67.3%	67.5%	50.9%	43.5%	58.4%	46.4%	63.7%
Precision	0.686	0.681	0.684	0.554	0.604	0.596	0.536	0.671
Recall	0.676	0.673	0.721	0.513	0.443	0.580	0.456	0.654
F-Score	0.663	0.659	0.665	0.471	0.376	0.561	0.451	0.629

Table 4.2: Comparing Classification Performance of Different Tokens

Since the best performing tokenization sequence had to be selected in this work in order to finally perform tweet classification on the enhanced dataset with additional sentiment features, the alphabetic tokenizer scheme was used. It is to be noted that a total of 381 features were used in case of alphabetic tokenizer.

4.2.5 Final Classification Of Tweets Using Supervised Machine Learning

As the objective of the current research is to present a reliable model of tweet classification for disaster response using various machine learning classifiers and to compare their performance against each other, with and without using the enhanced dataset, we trained a total of 15 classifiers in each case. As the alphabetic tokenizer produced highest accuracy, all the machine learning models made use of that token sequence. The classifiers were trained in two modes as mentioned: *Mode 1 - Using only tweet text as features*, and *Mode 2: Using tweet text along with sentiment scores, polarity, subjectivity and confidence levels as features* for text classification. 15 different classification algorithms were used in each mode of tweet classification and were finally compared for performance.

Machine learning for text classification can be performed using supervised, unsupervised and semi-supervised techniques. Since our dataset was pre-labeled, we used supervised machine learning for tweet text classification. Supervised classification include parametric classifiers like Logistic Regression and Naive Bayes as well as non-parametric classifiers like SVM, decision tree, rule induction, KNN and neural networks etc. It is important to note that all the classifiers were trained using 10-fold cross validation. There are many different choices of machine learning models but the ones we chose were consistently shown to provide better text classification capabilities in this domain (based on literature review). The following different classifiers were implemented for this purpose:

Probabilistic Classifier - Naive Bayes, Naive Bayes Multinomial, Bayes Net, ***Linear Classifier*** - Simple Logistic Regression, Multinomial Logistic Regression, ***Support Vector Machine Classifier*** - Sequential Minimal Optimization, ***Simple Neural Network*** - Multi Layer Perceptron, ***Deep Neural Network*** - Convolutional Neural Network, ***Rule Based Classifier*** - Decision Table, ***Tree Based Classifier*** - J-48, Random Tree, REP-Tree, Random Forest, Rotation Forest and ***Filtered Classifier***.

The description of each of the algorithms used is provided in this section followed by the results obtained.

Probabilistic Classifier: These classifiers provide a probability distribution of a piece of text over a number of classes instead of a single likely class. Three probabilistic classifiers were used in our case: *Naive Bayes, Naive Bayes Multinomial* and *Bayes Net*. Naive Bayes is a classification technique which assumes independence among predictor variables. Thus, the presence of a particular feature in a class is independent of any other features and each of these features contribute independently to the probability of a data instance to belong to a certain class. This algorithm has long been utilized for text categorization, spam filtering, medical diagnosis and many other application areas and is known to work very well. Multinomial Naive Bayes is another variant of Naive Bayes Classifier which uses multinomial distribution of each of the features to generate the likelihood of a data instance to belong to a certain class. Multinomial Naive Bayes is particularly suited for word counts in documents which naturally assumes multinomial distribution of features for prediction. Bayes Net or Bayes Network is another probabilistic graphical model representing features with their conditional dependencies with the help of a directed graph. Naive Bayes, Multinomial Naive Bayes and Bayes Net were used in their default configuration settings in WEKA. Of all the three probabilistic classifiers, Bayes Net performed the best in terms of highest % of accurately classified instances, precision, recall, F-score and kappa statistic. Lowest mean absolute error and root mean squared error was observed for Bayes Net.

Linear Classifier: A linear classifier performs classification on the basis of linear combination of the features. Input feature values are fed into the classifier in the form of feature vectors. This classifier is known to work well for problems with multitudes of features, for example text classification, as it includes multiple input features. Logistic regression is one type of linear classifiers which measures the relationship between the target categorical dependent variable and one or more independent variables. This is done by calculating the probabilities using a sigmoid logistic function. Both simple logistic and multinomial logistic regression were used in the current experiment, again,

in their default configurations. Multinomial logistic regression works well with multi-class problems resulting in more than two possible outcomes. Multinomial logistic regression performed better than simple logistic regression for tweet text classification.

Support Vector Machine Classifier: Support Vector Machine (SVM) is a supervised machine learning technique used for classification and regression problems. This is found to be extremely useful for text categorization because of its minimal need of labeled training examples. An SVM model represents mapping of data points in space such that there exists a clear gap between different categories. This model works by identifying a single hyperplane out of many, which provides the largest separation between two classes. This hyperplane maximizes the distance from it to the nearest data point on each of its side. Sequential Minimal Optimization (SMO) is an iterative optimization algorithm used for training SVMs and particularly suited for sparse datasets. This algorithm is much faster as it breaks down the learning problem into a subset of small tasks for optimizing learning time. SMO was used in its default settings and it produced very high quality classification results.

Simple Neural Network: A simple neural network is a mathematical formalism which is based on a collection of units called neurons which are designed to emulate the behavior of nerve cells or neurons in biological systems. These networks model complex relationships and patterns existing in a dataset by means of finding a mathematical function that establishes a relationship between different features. Simple neural networks comprise of three layers - input, hidden and output layers which receive, map and output this relationship in the form of a mathematical function. A multilayer perceptron (MLP) is a special type of feed-forward neural network consisting of the same three layers and uses back-propagation algorithm for model training. Each neuron (perceptron) in an MLP classifier uses a non-linear activation function to distinguish non-linearly separable data. Again, MLP was used in its default settings.

Deep Neural Network: Deep Neural Networks are an extension of simple neural networks with a differing number of hidden layers. The word 'deep' refers to the high number of hidden layers used in this model which actually perform complex computations than simple sigmoid or relu activations. There are different types of deep learning

models that can be applied to the problem of text classification. WEKA does not have a deep neural network model readily available in its Explorer Window. An external deep learning package for WEKA workbench which provides access to deep learning in WEKA was downloaded. This package known by the name 'WekaDeeplearning4j' was then used to train a deep Convolutional Neural Network in WEKA in its default settings.

Rule Based Classifier: Rule-based classifier is based on a set of rules to perform classification. The rules are simple IF-THEN statements that can be extracted from the dataset. Rules are written for each class so that a given class can be correctly identified and separated from others. Decision Table is one such rule induction classifier which works like a simple lookup table. This algorithm has two components - a schema and a body. Schema holds the list of attributes while the body holds sets of labeled instances. For a given unlabeled instance, this classifier searches for an exact match using the attributes in schema and returns a majority class of the decision table if no instances are found. Otherwise, the majority class of all matching instances are returned if a match is found. Decision Tables can also be run in conjunction with other classifiers such as Naive Bayes etc. to generate a hybrid classifier. The hybrid classifier evaluates the merit of splitting the attributes into two disjoint subsets: one for decision table, and the other for the other classifier used in conjunction with decision table. A stand-alone Decision Table classifier in its default settings was used. This classifier produced very good results with high percentage of accurately classified instances and higher values for precision, recall, F-measure and Kappa statistic.

Tree Based Classifier: These classifiers use decision trees for predictive modeling of the dataset into a set of discrete class values represented as leaf nodes of the tree. The individual leaves represent the respective class labels while the branches of the tree represent the pathway of rules that led to the leaves. Five different tree-based models were run on our dataset including J-48, Random Tree, REP-Tree, Random Forest and Rotation Forest. J-48 classifier builds uni-variate pruned and un-pruned decision trees in WEKA using C4.5 algorithm. Random Tree classifier constructs a decision tree based on k-randomly chosen attributes at each node. It does not perform any

pruning. It also provides an estimation of class probabilities based on a hold-out set. A REP-Tree is another variant of decision tree which is built using information gain or variance. It performs pruning using reduced-error approach and is a fast tree learning algorithm. Random Forest is a bagging ensemble model used for constructing a forest of random trees. This model averages multiple decision trees trained on subset of feature vectors with an aim of reducing the overall variance. Lastly, a rotation forest is another ensemble technique which thrives on splitting feature vectors into subsets and then performs a principal component analysis to each subset. This usually generates much accurate results than boosting and bagging ensemble models. All tree based classifiers were run on their default settings. Rotation forest gave the best results followed by J-48 and Random Forest. Other tree based methods didn't perform very well on our dataset.

Filtered Classifier: Manipulation of attributes is sometimes necessary before they are fed into a classifier. This can be done by removing, adding, transforming, randomizing or normalizing them. WEKA allows for a number of filters that can be applied on the text before running a classifier. Filtered classifier can be any arbitrary classifier performed on a dataset passed through an arbitrary filter for attribute manipulation. The structure of this filter depends entirely on the training data. In case of unequal instance weights or other problems with the dataset, the attributes are re-sampled with replacements before being fed into the classifier. This classifier was again used in its original configuration and produced results with high classification accuracy.

Text classification models can be improved further by text cleaning (by cleaning noise present in text), feature stacking (by combining different subsets of feature vectors), tuning hyper parameters in modeling (changing model parameters, e.g. changing tree depth, network parameters, learning rate, activations etc.) and by using ensemble models (combining different models and blending their outputs). Improvement in text classification however could not be accommodated in this work due to sheer lack of time.

The classification results obtained both by using original dataset and enhanced dataset with additional sentiment features are presented in the upcoming sections.

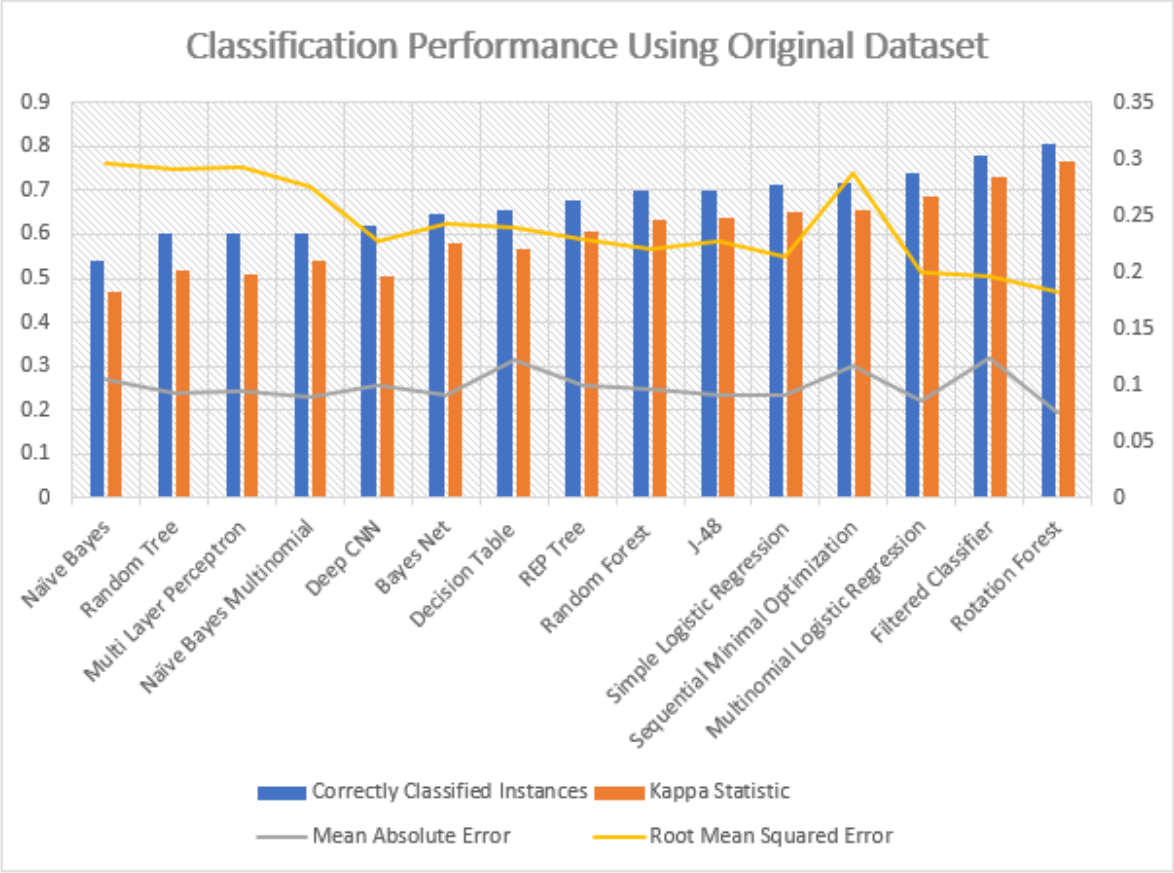


Figure 4.20: Performance of machine learning classifiers using original dataset

Using Original Dataset

Figure 4.20 represents the performance of all the classifiers run on original tweet dataset. This is a dual axis chart where the left side scale represents values of correctly classified instances and kappa statistic while the right side scale represents the values of mean absolute and root mean squared error for the classifiers.

From the figure, it is observed that probabilistic Naive Bayes classifier produced lowest classification accuracy, simple neural and deep neural networks also performed low in comparison to tree based methods of Random Forest and J-48. Highest classification performance is observed for Logistic Regression, Sequential Minimal Optimization, Filtered Classifier and Rotation Forest ensemble models.

In addition, Table 4.3 is presented which provides a numerical value for % of correctly classified instances, Kappa Cofficient Statistic, Mean Absolute Error (MAE)

and Root Mean Squared Error (RMSE) for each algorithm.

S.No.	Classifier Name	% Accuracy	Kappa-Coeff.	MAE	RMSE
1	Naive Bayes	53.97	0.47	0.105	0.296
2	Random Tree	60.01	0.52	0.093	0.290
3	Multi Layer Perceptron	60.06	0.51	0.095	0.292
4	Naive Bayes Multinomial	60.27	0.54	0.090	0.275
5	Deep CNN	61.77	0.51	0.099	0.227
6	Bayes Net	64.49	0.58	0.090	0.243
7	Decision Table	65.48	0.57	0.122	0.240
8	REP-Tree	67.84	0.61	0.101	0.229
9	Random Forest	69.77	0.63	0.096	0.220
10	J-48	70.02	0.64	0.091	0.229
11	Simple LR	71.45	0.65	0.091	0.213
12	Seq. Minimal Optimization	71.78	0.66	0.116	0.287
13	Multinomial LR	74.13	0.68	0.088	0.200
14	Filtered Classifier	77.74	0.73	0.123	0.197
15	Rotation Forest	80.48	0.76	0.075	0.182

Table 4.3: Accuracy of Classification Performance using Original Dataset

The class-wise performance of these classifiers on the original dataset is provided in Evaluation and Analysis (Chapter 5) where a fully detailed account of True Positive (TP) Rate, False Positive (FP) Rate, Precision, Recall, F-Measure, Receiver Operating Characteristic (ROC) Area and Precision Recall Curve (PRC) Area are plotted for each of the 9 humanitarian classes that the data was classified into. Discussions and observations are then presented towards the end.

Using Enhanced Dataset

The original dataset that was loaded into WEKA in the form of ARFF file was converted into 381 word feature vectors as discussed in the previous sections. The 6

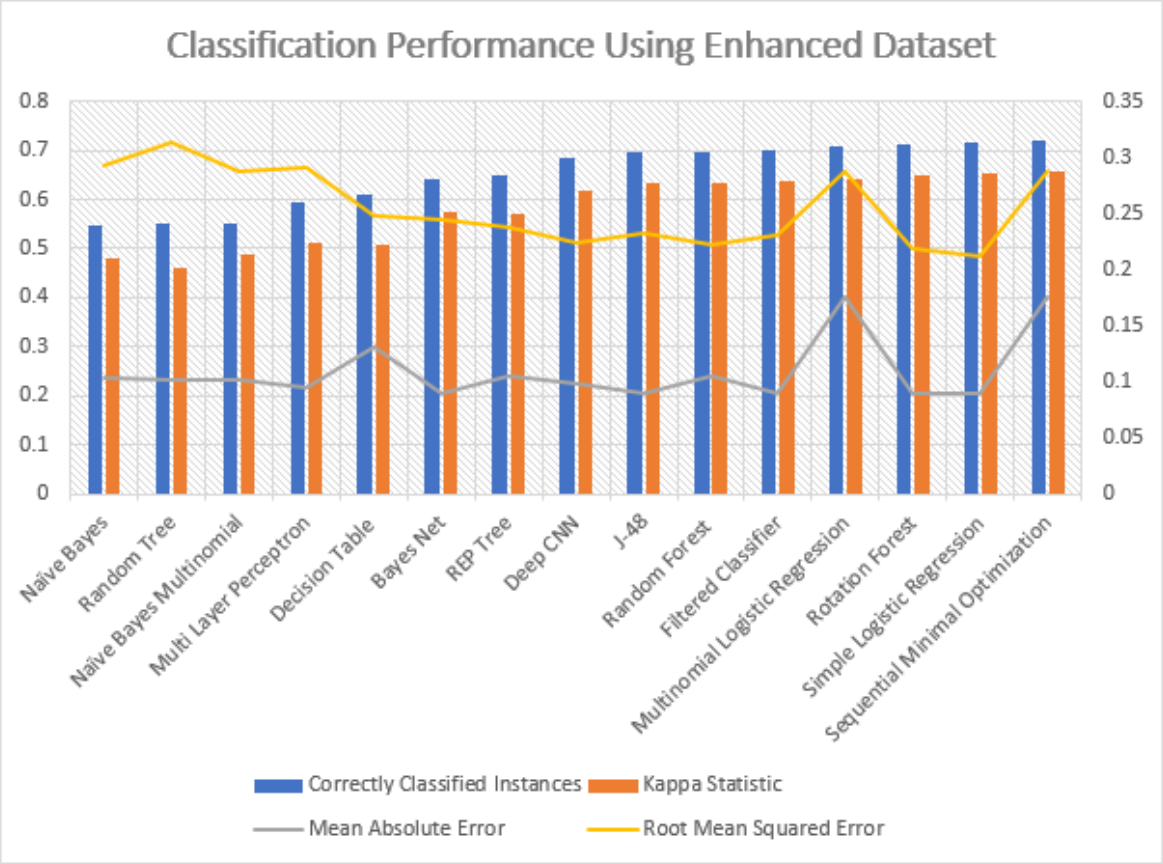


Figure 4.21: Performance of machine learning classifiers using enhanced dataset

additional sentiment features: *positive sentiment score, negative sentiment score, sentiment polarity, sentiment subjectivity, polarity confidence and subjectivity confidence* could not be appended directly to the 381 word feature vectors. The 381 word vectors extracted from the original dataset were represented as type 'numeric' while 4 out of 6 additional sentiment features were of type 'numeric'. The remaining two features, sentiment polarity and sentiment subjectivity were of type 'nominal'. The merging of additional 6 features into 381 word features was done with the help of a small program written in Java as shown in the Appendix of this report. The new ARFF file containing 381+6 = 387 features, also called 'Enhanced Dataset' was then used to perform text classification. All the 15 classifiers that were used on the Original Dataset were then re-used on the Enhanced Dataset to identify any differences in performance of text classification.

S.No.	Classifier Name	% Accuracy	Kappa-Coeff.	MAE	RMSE
1	Naive Bayes	54.90	0.48	0.103	0.294
2	Random Tree	54.95	0.46	0.101	0.314
3	Naive Bayes Multinomial	55.13	0.49	0.102	0.287
4	Multi Layer Perceptron	59.42	0.51	0.094	0.292
5	Decision Table	61.00	0.51	0.130	0.249
6	Bayes Net	64.21	0.57	0.090	0.244
7	REP-Tree	65.10	0.57	0.106	0.237
8	Deep CNN	68.55	0.62	0.098	0.223
9	J-48	69.73	0.63	0.089	0.233
10	Random Forest	69.80	0.63	0.105	0.222
11	Filtered Classifier	69.96	0.64	0.090	0.230
12	Multinomial LR	70.73	0.64	0.177	0.288
13	Rotation Forest	71.07	0.65	0.090	0.218
14	Simple LR	71.69	0.65	0.090	0.212
15	Seq. Minimal Optimization	71.84	0.66	0.176	0.287

Table 4.4: Accuracy of Classification Performance using Enhanced Dataset

Figure 4.21 represents the performance of all the classifiers that were run on enhanced tweet dataset. This is again a dual axis chart as was before in case of original dataset. From the figure, probabilistic Naive Bayes classifier produced lowest classification accuracy, simple neural and deep neural networks produced intermediate quality results while still lower in comparison to tree based methods of Random Forest and J-48. Highest classification performance is observed for Filtered Classifier, Logistic Regression, Rotation Forest ensemble models and lastly Sequential Minimal Optimization.

Again, Table 4.4 is presented which provides a numerical value for % of correctly classified instances, Kappa Coefficient Statistic, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for each algorithm that was run on enhanced dataset.

Chapter 5

Evaluation and Analysis

This chapter evaluates and discusses the results obtained from chapter 4. Inferences from results are generated and an overview of strengths and weaknesses of the conducted experiment are also provided. The evaluation of classification results are then discussed in terms of statistical testing so that quantitative conclusions about the formulated hypothesis can be drawn.

5.1 Evaluating Results of Sentiment Analyses

Tweet sentiment analyses was performed using a variety of methods including programming with R, Senti-Strength Tool and Rapid Miner Data Analysis Platform using AYLIEN and Rosette Text Analytics package. There was no sentiment labeling available for the dataset and thus there was no way to directly ascertain or evaluate the quality of sentiment analyses tasks, than to use the tools with their predefined accuracy levels (derived from literature review). Senti-strength is specifically dedicated to perform sentiment analyses on twitter dataset with an accuracy of around 70% (Abbasi et al., 2018), while the commercially available tools of Rosette and AYLIEN also produce an equivalent accuracy of sentiment analyses. The 'TidyText' package of R is also quite extensively used for sentiment analyses and word processing. Again, in the absence of a labeled sentiment dataset, using three different approaches to compare and assess how they relate or differ from one another provides a sound basis

of comparison and evaluation. From the results of multidimensional textual content analyses of tweets, the average positive and negative sentiment scores did vary across different techniques because of the underlying differences in weighting mechanisms of sentiments words, the polarity of tweets however remained fairly the same across each tool. Also, the distribution of sentiment polarity (Negative and Positive) across the 9 humanitarian categories remained surprisingly same using all the three different tools. The consensus among the results obtained from different tools validates the generated output.

5.2 Evaluating Performance of Text Classification

To address the problem of randomness in machine learning, a 10-fold cross validation was used so as to generate a population of performance measures from a certain classifier instead of getting a single result from it. In addition, the random repeats/restarts in machine training of classifiers was performed to generate many different results from the same classifier in an attempt to reduce uncertainty. This resulted in providing a summary statistics of the performance measures for weighted average score for precision, recall, F-measure, True Positive and False Positive Rates, Receiver Operating Characteristic (ROC) and Precision Recall Curves (PRC) rather than a single score per classifier.

Mean and standard deviation of performance were also provided including the highest and the lowest performance observed for a classifier. This gives a realistic picture of the classifier performance and minimizes the possibility of any bias. Furthermore, statistical significance tests using Paired-T Tester in WEKA were used to determine if the difference between one population of results is significantly different from another population performed on the two datasets - original and enhanced. A significance level of 0.05 was used. This gave evidence as to whether the null hypothesis should be accepted or rejected.

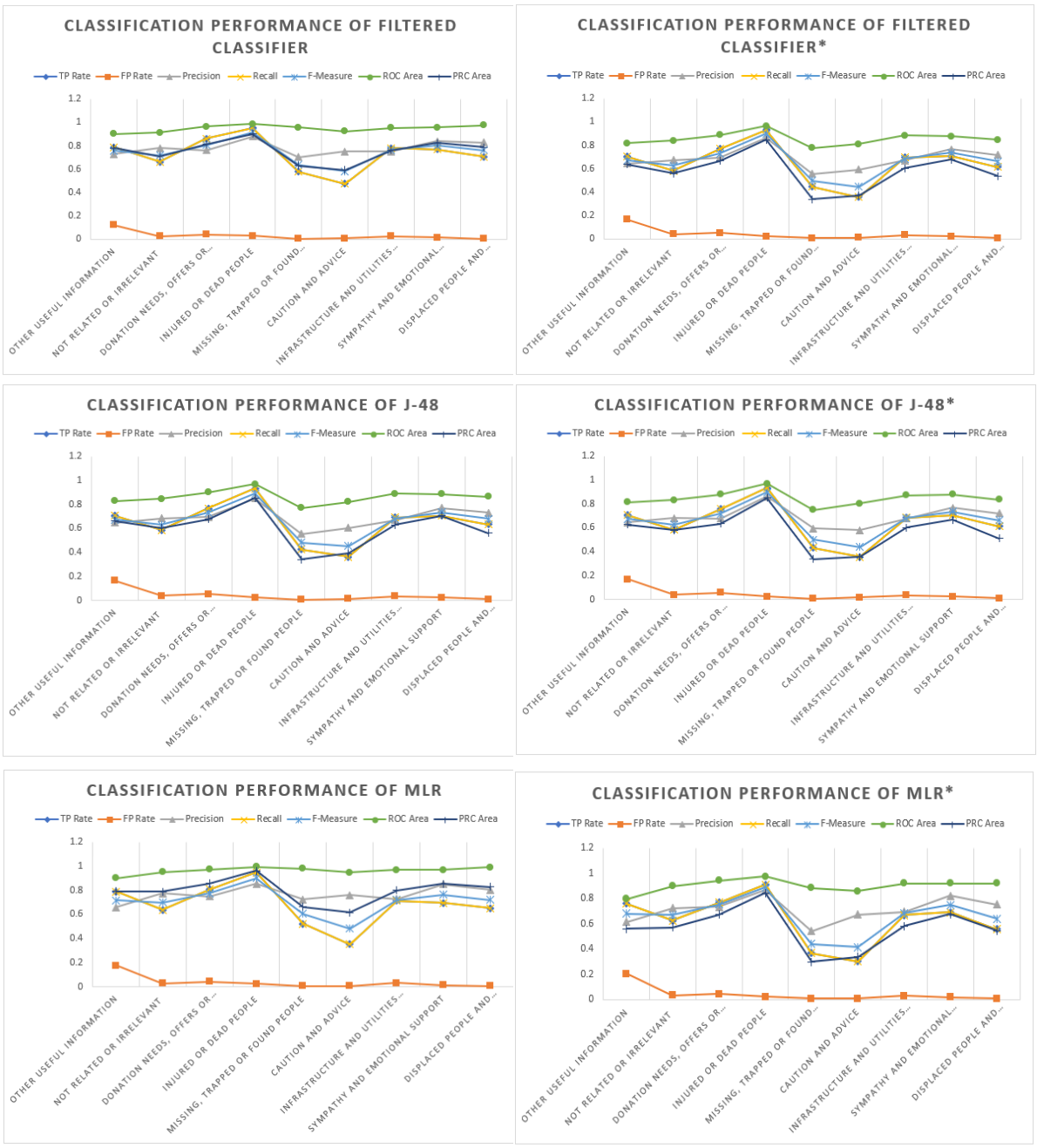


Figure 5.1: Performance of Filtered Classifier, J-48 and Multinomial Logistic Regression using both the datasets.

Note: The figures on the left represent the performance on original dataset while the figures on the right with an asterisk (*) represent the performance on enhanced dataset.

5.2.1 Analyzing Classification Results of Original & Enhanced Datasets

Fifteen different machine learning classifiers were trained on the Original as well as Enhanced Dataset in their default configuration settings. The different classifiers used have already been discussed in the previous chapter describing their working principle and performance levels in terms of percentage of correctly classified instances, Kappa statistic, mean absolute error and root mean squared error. From the results presented in Chapter 4, it is observed that the Tree-based methods (J-48, Random Forest, Rotation Forest), filtered classifier, Support Vector Machine optimization method, also called Sequential Minimal Optimization (SMO), and Linear Classifiers (Logistic Regression) perform much better than Neural Networks, Decision Tables and Probabilistic Classifiers (Naive Bayes, Naive Bayes Multinomial and Bayes Net). The results obtained are in close alignment with the previous works in this domain as detailed in (Imran et al., 2015, 2018).

The classification accuracies for this multi-class problem obtained from the best performing classifiers ranged from as low as around 60% in case of Naive Bayes Multinomial and Bayes Net Classifiers to as high as around 80% in case of Rotation Forest. The percentage accuracy of correctly classified instances in case of Logistic Regression and Sequential Minimal Optimization was found out to be roughly around 70%. Filtered classifier performed fairly well due to the use of suitable filters applied to the dataset. The resulting text classification accuracy in case of filtered classifier was observed to be consistently over 75%. Similarly, the classification accuracy of tree-based models (J-48, Random Forest etc.) is seen to be consistently high reaching up-to 80% in case of Rotation Forest (Ensemble Model).

The precision and recall values for these classifiers lie in the range of 0.70 to 0.80 while the F-score is a bit low (0.65 to 0.70 to be specific). Precision, Recall, F1-Score and % of correctly classified instances do provide a means to compare the classification performance , using additional statistics and error percentage is also helpful. The value of Kappa Statistic for the best performing classifiers ranged from 0.63 to 0.76 while the

mean absolute error and root mean squared error ranged from 0.07 to 0.12 and 0.18 to 0.29 respectively. The value of Kappa Statistic for the least performing algorithms ranged from 0.47 to 0.58 while mean absolute and root mean squared errors ranged from 0.09 to 0.10 and 0.22 to 0.29 respectively. These values were observed in case of the original dataset.

The same statistics for enhanced dataset were computed and were found to differ slightly with respect to the original dataset. The value of Kappa Statistic for the best performing classifiers ranged from 0.63 to 0.66 while the mean absolute error and root mean squared error ranged from 0.09 to 0.18 and 0.21 to 0.29 respectively. These results indicate that the performance of classifiers do not differ much at all in the two datasets. An analysis of confusion/error matrices for these classifiers however showed a different classification behavior, not evident on the basis of observing the performance metrics of Precision, Recall and F-Score. The results of confusion matrices are covered in sub-section 5.2.2 of the current Section.

As mentioned before, six out of the total of fifteen classifiers which gave the best classification results, are used for illustration purposes, for the sake of simplicity and convenience. Figures 5.1 and 5.2 present the findings of the six best performing classifiers on both the datasets. The images on the left refer to the classifiers trained on Original Dataset while the images on the right with an asterisk (*) refer to the classifiers trained on Enhanced Dataset.

From these figures, it is clear that the overall classification performance is lowest for the tweet category of *Caution and Advice* followed by *Missing, Trapped or Found People* and *Displaced People and Evacuations* while highest classification performance is observed for the tweet categories of *Injured or Dead People* followed by *Donation Needs, Offers or Volunteering Services* and *Sympathy and Emotional Support*. This pattern is the same for all the 6 best performing algorithms trained on both the datasets (original and enhanced). This behavior can be partly explained in terms of class imbalance. The dataset, to begin with, had lower number of instances for the tweet categories of *Caution and Advice*, *Missing, Trapped or Found People* and *Displaced People and Evacuations* and the results reveal a lower classification perfor-

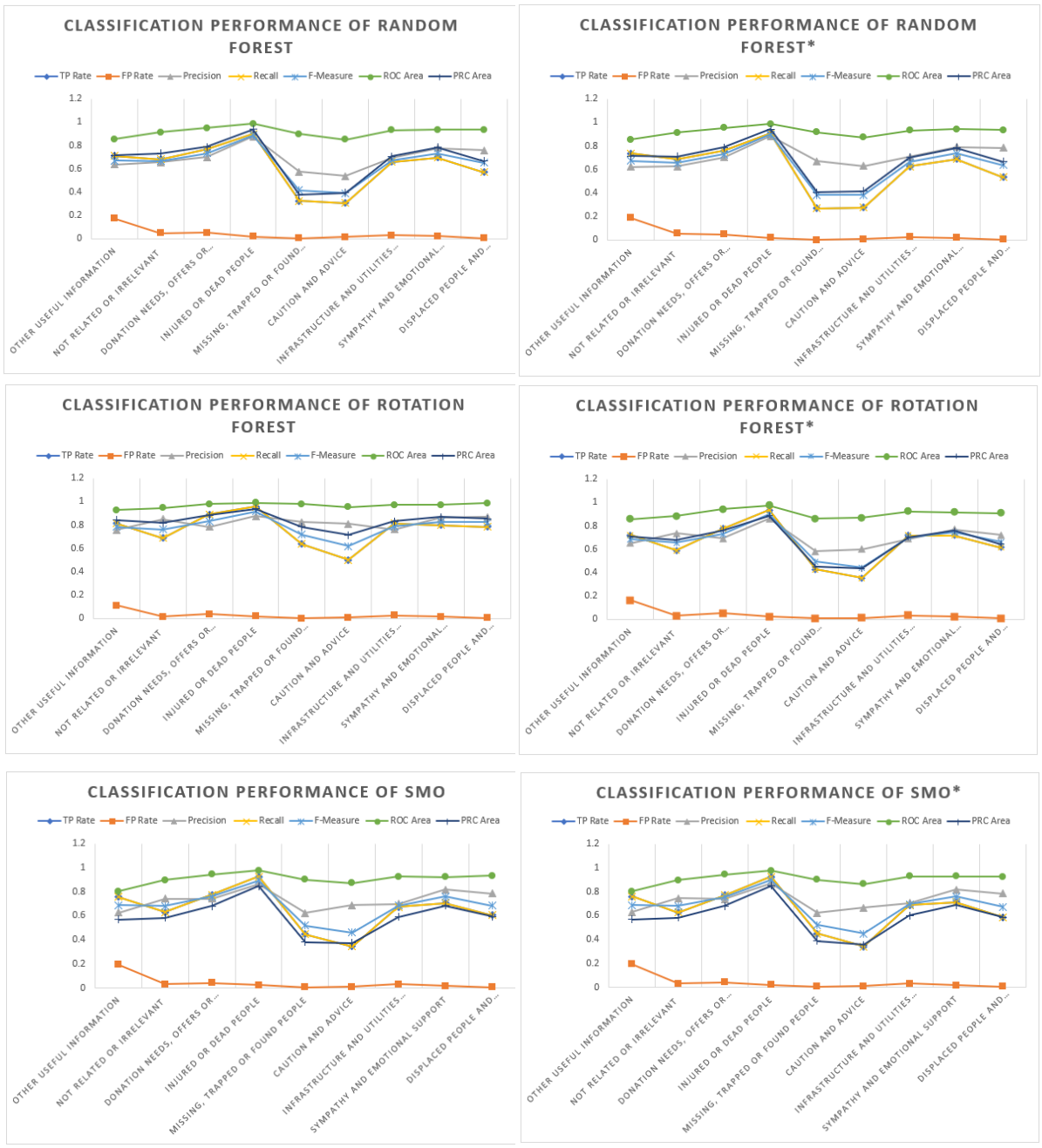


Figure 5.2: Performance of Random Forest, Rotation Forest and SMO using both the datasets.

Note: The figures on the left represent the performance on original dataset while the figures on the right with an asterisk (*) represent the performance on enhanced dataset.

mance in those classes, possibly due to lower number of data instances. There could be several other confounding factors responsible for this which can't be evaluated entirely

from these plots however.

5.2.2 Analyzing Confusion Matrices of Best Performing Classifiers

Of the 15 machine learning classifiers that were run on the dataset, 6 produced the best results in terms of overall weighted average precision, recall, F-score and percentage of correctly classified instances. Not much difference in performance of the classifiers was seen using both the datasets (Original and Enhanced). A closer look at the classification confusion matrices, however, suggest that including sentiment based features during text classification changes the classification behavior (without really changing the classification accuracy). This is an interesting insight obtained from the current exercise and is discussed here.

A confusion matrix, also known as error matrix provides a tabular visualization of statistical classification per class. Each row of the matrix represents predicted value of a class while each column represents actual value of a class that a data instance belongs to. The diagonal entries are correctly classified instances while the remaining entries are mis-classified.

	a	b	c	d	e	f	g	h	i
a	**4335**	252	353	104	34	120	400	91	40
b	557	**1475**	78	87	4	11	42	87	3
c	394	60	**2022**	36	24	4	11	44	15
d	110	22	15	**2320**	8	1	11	15	8
e	131	9	33	19	**179**	1	12	9	9
f	538	31	26	18	2	**367**	24	42	15
g	445	31	36	45	12	7	**1248**	16	11
h	274	104	115	36	7	20	19	**1390**	4
i	129	8	44	17	18	4	24	6	**383**

Table 5.1: Confusion Matrix for SMO on Original Dataset

	a	b	c	d	e	f	g	h	i
a	**4345**	251	345	102	34	125	396	96	35
b	573	**1465**	79	85	4	14	31	91	2
c	404	56	**2015**	34	18	9	17	42	15
d	105	22	21	**2320**	9	1	10	14	8
e	125	11	40	20	**181**	1	6	7	11
f	555	25	27	15	2	**362**	26	36	15
g	425	29	37	43	11	7	**1271**	18	10
h	261	103	116	31	12	21	20	**1399**	6
i	126	8	50	21	20	5	24	7	3**72**

Table 5.2: Confusion Matrix for SMO on Enhanced Dataset

The confusion matrices for two of the best performing classifiers is provided here - Sequential Minimal Optimization and Rotation Forest (for the sake of convenience). These are presented in the form of tables 5.1 and 5.2. In the confusion matrix tables, the class labels are represented by means of alphabets for simplicity, as follows:

a is *Other Useful Information*, *b* is *Not Related or Irrelevant*, *c* is *Donation Needs or Offers or Volunteering Services*, *d* is *Injured or Dead People*, *e* is *Missing, Trapped or Found People*, *f* is *Caution and Advice*, *g* is *Infrastructure and Utilities Damage*, *h* is *Sympathy and Emotional Support* and *i* is *Displaced People and Evacuations* respectively.

The analysis of confusion matrices of the best performing classifiers yielded a common pattern as presented: The number of correctly classified tweets belonging to categories *Infrastructure and Utilities Damage, Injured or Dead People, Sympathy and Emotional Support* and *Other Useful Information* usually increased when using Enhanced Dataset. The number of correctly classified tweets belonging to the category *Not Related or Irrelevant* remained almost the same while a decrease in number of correctly classified tweets is observed for the category *Donation Needs, Offers or Volunteering Services* when using Enhanced Dataset. Interesting is to observe that those

categories (mentioned above) having maximum sentiment words (strongest sentiment scores, both positive and negative) are generally seen to be more accurately classified than those with low or minimum amount of sentiment words present. This is an important insight from the findings of the classification results using the enhanced dataset as it helps to devise better classifiers yielding more accuracy in correctly classifying certain humanitarian classes over the others.

	a	b	c	d	e	f	g	h	i
a	**4059**	424	401	80	26	190	382	125	42
b	425	**1596**	91	77	6	20	30	93	6
c	360	103	**1982**	34	20	16	18	63	14
d	107	37	28	**2264**	6	8	25	26	9
e	126	16	57	25	**130**	3	10	14	21
f	544	54	34	12	2	**328**	28	48	13
g	428	62	52	41	3	20	**1223**	15	7
h	251	126	144	32	5	21	25	**1363**	2
i	119	22	50	27	20	12	18	10	**355**

Table 5.3: Confusion Matrix for Rotation Forest on Original Dataset

5.2.3 Statistical Treatment of Experimental Results

WEKA provides an experiment analyzer window which is used to analyze the results of the experiment. In order to compare the performance of the best selected classifiers on the two datasets - Original and Enhanced Dataset, the statistical Paired T-test available from *WEKA Experimenter* window was used. The Paired T-Tester was, in turn, run for each of the 6 classifiers to confirm if there was any statistically significant difference in the performance of the text classification results using the two datasets. The comparison field on which the statistical test was based was *Percent Correct* Attribute kept at a significance level of 0.05.

In the WEKA Experimenter, the text classification schemes used in the experiment

	a	b	c	d	e	f	g	h	i
a	**4230**	436	377	74	17	130	322	114	29
b	442	**1599**	95	83	1	8	29	85	2
c	409	109	**1974**	29	13	9	15	43	9
d	118	38	29	**2266**	5	1	22	26	5
e	134	32	66	20	**105**	1	6	20	18
f	593	56	29	9	1	**284**	23	57	11
g	510	57	57	37	1	12	**1151**	20	6
h	233	165	134	24	3	15	21	**1373**	1
i	145	19	53	22	10	12	21	9	**342**

Table 5.4: Confusion Matrix for Rotation Forest on Enhanced Dataset

are shown in columns while the dataset used is shown in row by default. However, this setting was swapped to include data in columns and classification scheme in row. This was done because the experiment to be conducted had to be tested on the different datasets and not different classifiers. A corrected paired T-test was then performed using each of the 6 best performing classification algorithms using the two datasets - Original and Enhanced.

This statistical test on each of the following algorithms:*Logistic Regression, Sequential Minimal Optimization, J-48, Random Forest, Rotation Forest* and *Filtered Classifier* were conducted using 10-fold cross validation with a specified number of repetitions (10 for each dataset). It was observed that the results of statistical paired T-test were same using all the 6 best performing algorithms. Here, the results of statistical analysis using random forest and filtered classifier algorithms are presented for simplicity, in the form of Figures 5.3 and 5.4 respectively.

It is important to note that the percentage correct for each of the classifier scheme using both the datasets is shown in columns. The annotation v or * below the percentage correct indicates that a specific result is statistically better (v) or worse (*) than the baseline scheme (in this case, Original Dataset) at the significance level of

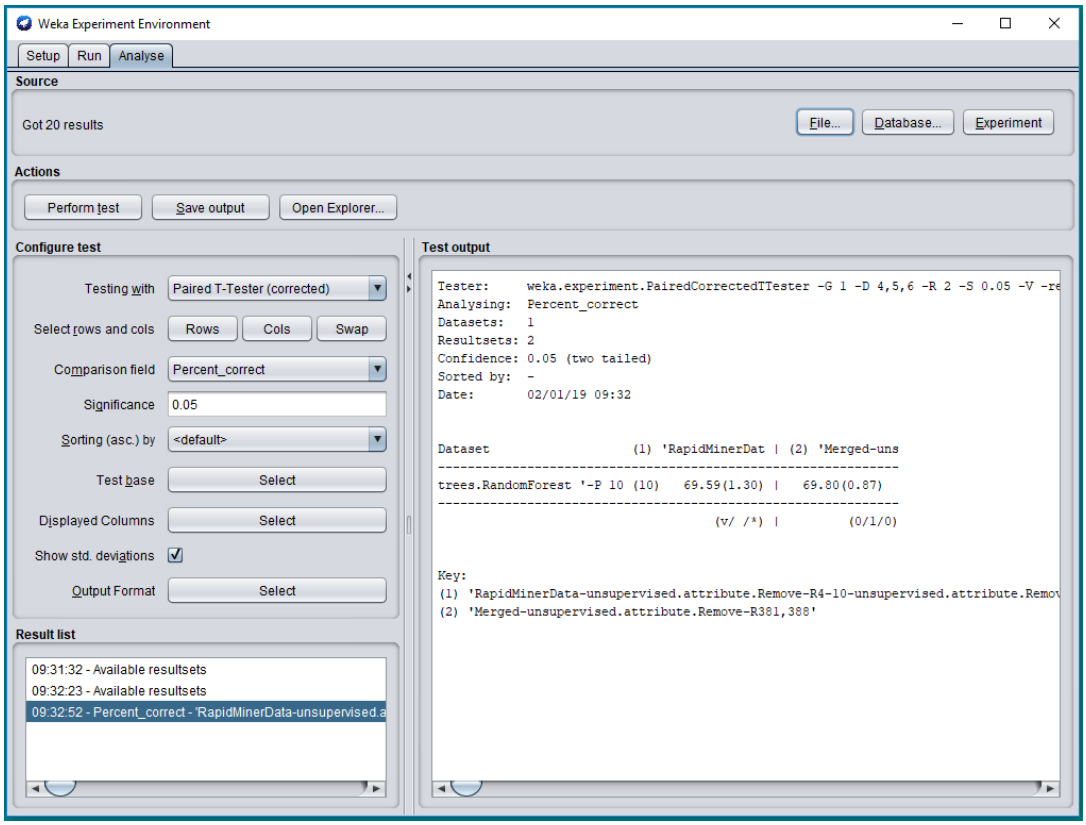


Figure 5.3: Statistical Paired T-Testing of Random Forest Classifier

0.05. The results of both the datasets for a single classifier are observed for statistical comparison with the baseline. The baseline in this case refers to the percentage of correctly classified tweets using the Original Dataset.

As one can see from the figure 5.3, the average accuracy using a Random Forest Classifier is marginally better in case of enhanced dataset than the original, the result (0/1/0) indicates that at the 0.05 confidence level, there is no significant difference between the performance of Random Forest on the baseline dataset (Original) vis-a-vis the Enhanced Dataset. Again, from figure 5.4, the average accuracy using a Filtered Classifier is slightly less in case of enhanced dataset than the original, the result (0/1/0) indicates that at the 0.05 confidence level, there is no significant difference between the performance of Filtered Classifier on the baseline dataset (Original) vis-a-vis the Enhanced Dataset. The same results were observed for all the classifiers, this helped in arriving at the conclusion that the slight variation in performance of a classifier using

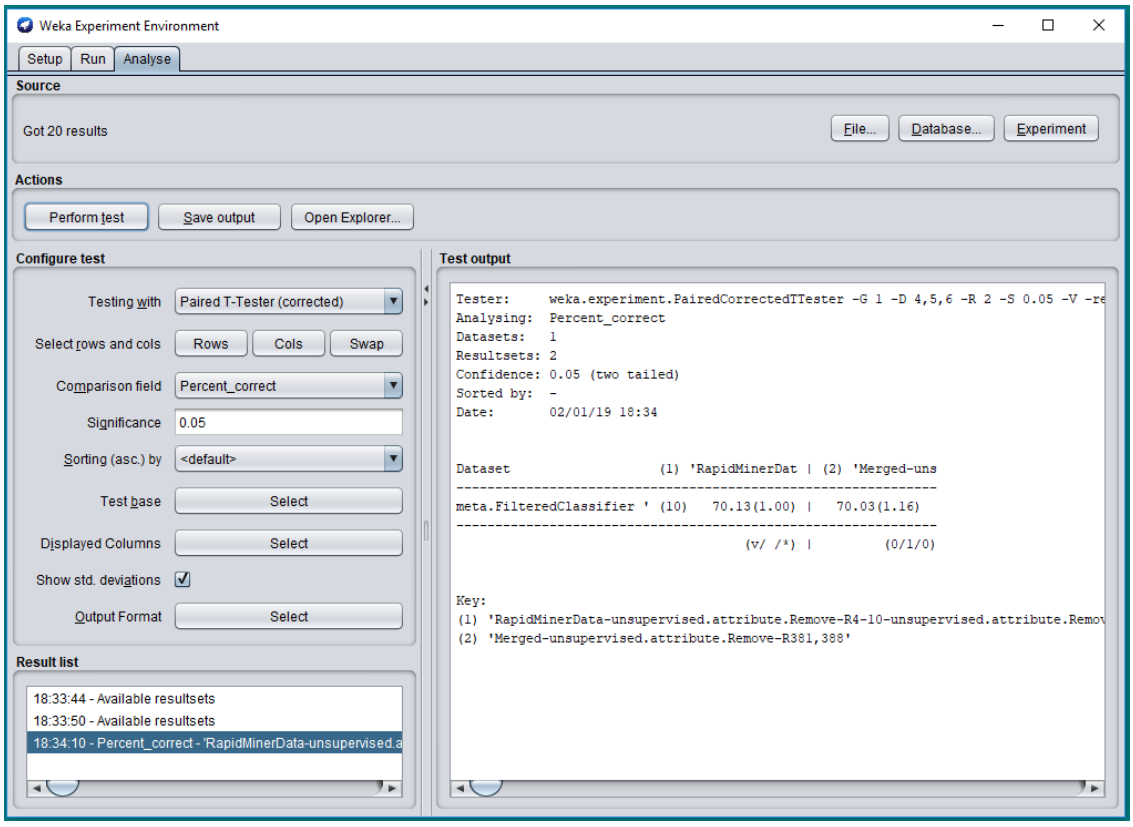


Figure 5.4: Statistical Paired T-Testing of Filtered Classifier

original and enhanced datasets is not statistically significant, thereby not allowing the null hypothesis to be accepted. It was inferred that the classification accuracy of a machine learning classifier does not improve by including sentiment based features in addition to usual word vector features.

5.3 Strength of Findings

The main strength of the results lie in the fact that no studies (to the author's awareness) have taken to combine the sentiment analyses with tweet text classification in this domain, thus making the research question authentic and original. Another strength is that the results of text classification as obtained from other works are in alignment with the work presented here. The current study mirrors the findings of previous work and manages to outline the application of text classification using sentiment based features in a disaster scenario. Sentiments are important indicators of situational awareness

when coupled with additional analyses, their use as features for text classification has a solid basis. In terms of classification performance, Logistic Regression, SMO, Random and Rotation Forest demonstrated highest capabilities. From inspecting the confusion matrices of classification results using both the datasets, there is an improvement in the detection of tweets belonging to the categories *Infrastructure and Utilities Damage*, *Injured or Dead People*, *Other Useful Information*, and *Sympathy and Emotional Support*. There was a drop in performance of detecting tweets belonging to *Donation Needs, Offers or Volunteering Services* while no change in detecting tweets related to *Not Related or Irrelevant* is observed using the enhanced dataset. The results are extremely useful to humanitarian organizations working at various levels (local, regional etc.) as their specific information needs can be addressed directly by including sentiment features for text classification. Even though inclusion of sentiment features for tweet text classification resulted in no overall and significant improvement in performance, it opens up new avenues for using entity-level or aspect-based sentiment features to see their impact on classification.

5.4 Limitation of Findings

The main limitation of the results is that the text classification algorithms were run in their default settings and thus, parameter tuning was not possible. Some algorithms would have worked better if the model parameters were fine-tuned. Again, hybrid classifiers or ensemble models could have given better results, this could not be explored. Lastly, feature stacking and further cleaning of tweets could have resulted in better performance. As suggested by Imran (Alam, Ofli, Imran, & Aupetit, 2018), tweets from one type of disaster event can't be used to effectively classify tweets coming from another type of disaster event, thus the validity and suitability of applying machine learning models for disaster response would improve further with the richness of dataset available (larger number of tweets coming from diverse disaster events). Another major limitation is that the models cannot determine or assess the quality of the textual content. This can be something that future studies can focus upon.

Chapter 6

Conclusion

This chapter provides conclusions inferred from this body of work. It briefly touches on the research overview, problem definition, experiment design, evaluation and results as discussed in the previous chapters. Towards the end, it discusses the contributions and impact of the experiment conducted in this work while also pointing towards any future work and recommendations for further studies in this domain.

6.1 Research Overview

This research was conducted in two parts - analyzing the textual content of tweets posted during crises events and performing text classification on those tweets. Content analysis of tweets was performed using sentiment analysis (various approaches), extracting important named-entities from tweets and lastly contextual categorization of tweets into various topics. The results from sentiment analyses of tweets yielded important features like sentiment scores, polarity, subjectivity and confidence levels that were utilized for text classification with a view to enhance the classification performance. The performance of tweet text classification was characterized and evaluated against original dataset with no sentiment features for training 15 different classification algorithms. The classification performance of each algorithm was compared in terms of calculated precision, recall, F-score and % of correctly classified instances. Finally, to conclude, a statistical analysis of results obtained from different classifiers

was used to ascertain whether the classification performance using additional sentiment features was statistically significant so that the formulated hypothesis of the research could be accepted or rejected. The relationship between sentiment features and classification performance was then summarized.

6.2 Problem Definition

The research problem was defined by the question: *'Does the performance of tweet-text classification improve (change) by including sentiment based features in addition to word vector features?'* And four sub-questions:

What type of multi-dimensional textual content analyses can be performed on disaster-related tweets that can be used as features for text classification?

Does classification performance differ based on token(izer) sequence used?

Does the inclusion of additional sentiment features along with default word features improve the accuracy of text classification?

Which classifier gives the best performance in terms of weighted average precision, recall and F1 score?

The primary purpose of the research was to establish the validity of the following hypotheses:

Null Hypothesis- Using sentiment based features in addition to word vector features does not affect the performance accuracy of tweet-text classification.

Alternative Hypothesis- Using sentiment based features in addition to word vector features improves the performance accuracy of tweet-text classification.

The research focused on exploring the application of 15 different classification algorithms with a goal of classifying disaster-related tweets into one of the 9 predefined humanitarian categories to facilitate disaster response during mass emergencies. The results from different classifiers were compared against each other (using original and enhanced dataset with additional sentiment features).

6.3 Experimentation, Evaluation & Results

The design of experiment was sound and well grounded as it included an in-depth textual content analyses of tweets before performing text classification. The dataset was adequate in size (around 20,000 tweets) for different disaster events (hurricane, earthquake, typhoon, cyclone and flood). All the disaster-related tweets were pre-labeled by Crowd-Flower crowd-sourcing platform into one of the 9 predefined humanitarian categories, the dataset was quite balanced in terms of number of tweets per humanitarian category thereby providing a firm statistical ground for performing the experimentation.

The approach to perform tweet text content analyses was comprehensive as it not only included sentiment analyses but also named-entity extraction and contextual tweet categorization. This made sure that all relevant aspects to generate situational awareness during disaster events were thoroughly covered. The tools chosen to perform multidimensional textual content analyses of tweets were based on exhaustive literature review as they were known to produce best results for related tasks. In addition, the same task of sentiment analyses was performed using three different methods - using programming with R, using Senti-Strength and using Rapid Miner tool with text analytics extensions. This provided comparative assessments of the performance of sentiment analyses. Same results were produced from each tool suggesting a valid output. This was extremely useful as there was no other way to assess the accuracy of performance (the dataset was not labeled for sentiments). This also added complexity to the project in terms of data preparation, feature engineering, model execution, and also in communicating and presenting the findings by means of graphical plots.

Similarly, the task of tweet text classification was performed on the original and enhanced dataset using 15 different classifiers to compare the performance. The choice of classifiers was again based on extensive literature review. Also, with a larger number of classifiers, one can be more confident in implying that the results produced are not randomly by chance but are an observed pattern and hence statistically significant. Six out of the 15 classifiers produced the best results for both the datasets (original

and enhanced). The machine learning classifiers that produced the best classification results in terms of highest percentage of accurately classified features, highest precision, recall, F-score and Kappa Statistic were found to be *Logistic Regression, Filtered Classifier, J-48, Random Forest, Rotation Forest* and *Sequential Minimal Optimization*. Analyzing the changes in classifier performance by changing model configuration settings was beyond the scope of this work and hence was not undertaken. This is one area where future work can be done.

From the results obtained and their analyses, it was concluded that there is no significant difference (change or improvement) in classification performance using sentiment based features in addition to word vector features for text classification. This was inferred because the same behavior was observed using all the classifiers. There was a very low to marginal improvement in classifier performance using additional sentiment features in some cases. The slight change in classifier performance can't be entirely attributed to the inclusion of sentiment based features because it is not statistically significant as suggested by ***Paired T-Tester*** in WEKA. The performance of each of the 6 best performing classifiers was compared for original and enhanced datasets using Paired T-tester (statistical significance of 0.05) and no significant difference between the performance was observed on the baseline dataset (original dataset) vis-a-vis the enhanced dataset (with additional features) although there were differences in terms of the classification behaviour evident in the confusion matrices. This result however, provides enough evidence to accept the null hypothesis that using sentiment based features in addition to word vector features does not affect the performance accuracy of tweet-text classification.

6.4 Contributions and Impact

In the current work, a thorough analyses and processing of the data collected from Twitter during natural disasters was performed. The richness of useful information obtained from Twitter feeds during disaster events has been demonstrated in this work. Although the focus of the current work was limited only to textual data obtained from

Twitter, it has the potential to be supplemented with additional information like images, multimedia and other content published on Twitter. Some of this work is already beginning to take place in the Qatar Computing Research Institute which involves the integration of images with text for better situational awareness.

The innovation of this work is that it brings together sentiment analyses and text classification which are much discussed as separate topics but haven't been amalgamated as one. Sentiments have a capability to distinguish one type of tweet from another and it was the driving force for trying to make use of sentiment scores, polarity and subjectivity to classify the tweets. The quality of textual content analyses impacts the tweet classification as it includes sentiment based features. In the current project, there was no better way to evaluate the accuracy of the sentiment analysis than to use a variety of methods and do a comparative analysis. Lastly, several state-of-the-art machine learning classifiers were employed for analyzing and classifying tweets useful for crises management and emergency response. While quantitative statistical analyses of tweet classification results using original and enhanced dataset (with additional sentiment features) do not show any significant improvement in classification performance, certain behavioral differences in text classification are evident.

6.5 Future Work & Recommendations

Extraction of spatio-temporal information to identify different patterns in the data using story-graph (Shrestha, Miller, Zhu, & Zhao, 2013) can be coupled with Geographic Information Systems (GIS) in real time and location, helping disaster management and planning considerably. Future work could look into expanding the capabilities of crisis analytics tools like AIDR in real-time scenario and not post disaster. The ongoing work is only limited to classifying tweets based on information content (text or image), spatial and temporal characteristics of events in real space and time are completely ignored, which are vital for any disaster response planning, and thus should be looked into. Among other issues is the scalability problem that needs addressal. Also, labeling tweets into categories performed by human annotator inherently slows down the

process, it is therefore worthwhile to look for approaches that do not need any human intervention at all.

Another area for future work is to use feature stacking, tuning model parameters and using ensemble models for improving accuracy of text classification. This work shows a clear behavioral difference in text classification when using sentiment based features in addition to the usual word vector features. This gives rise to several avenues of possible continuation specific to this project such as using aspect-based or entity-based sentiments rather than tweet sentiments (sentence level) and so on. Another future recommendation would be to use word embeddings and topic models on top of tweet sentiments as features for text classification.

References

Abbasi, A., Hassan, A., & Dhar, M. (2018). *Benchmarking Twitter Sentiment Analysis Tools.* U.S. National Science Foundation, University of Virginia, Charlottesville, Virginia, USA. (Unpublished Paper)

Alam, F., Joty, S., & Imran, M. (2018, May). Graph Based Semi-supervised Learning with Convolution Neural Networks to Classify Crisis Related Tweets. In *Proceedings of the twelfth international aaai conference on web and social media (icwsm 2018).* Retrieved 2018-10-07, from `http://arxiv.org/abs/1805.06289` (arXiv: 1805.06289)

Alam, F., Ofli, F., & Imran, M. (2018a, May). CrisisMMD: Multimodal Twitter Datasets from Natural Disasters. In *Proceedings of the twelfth international aaai conference on web and social media (icwsm 2018).* Retrieved 2018-10-04, from `http://arxiv.org/abs/1805.00713` (arXiv: 1805.00713)

Alam, F., Ofli, F., & Imran, M. (2018b, April). Processing Social Media Images by Combining Human and Machine Computing during Crises. *International Journal of HumanComputer Interaction*, *34*(4), 311–327. Retrieved 2018-12-01, from `https://www.tandfonline.com/doi/full/10.1080/10447318.2018.1427831` doi: 10.1080/10447318.2018.1427831

Alam, F., Ofli, F., Imran, M., & Aupetit, M. (2018, May). A Twitter Tale of Three Hurricanes: Harvey, Irma, and Maria. In *Proceedings of 15th international conference on information systems for crisis response and management (iscram),*

2018. Retrieved 2018-11-09, from http://arxiv.org/abs/1805.05144 (arXiv: 1805.05144)

Alfarrarjeh, A., Agrawal, S., Kim, S. H., & Shahabi, C. (2017, October). Geo-Spatial Multimedia Sentiment Analysis in Disasters. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 193–202). Tokyo, Japan: IEEE. Retrieved 2018-11-04, from http://ieeexplore.ieee.org/document/8259778/ doi: 10.1109/DSAA.2017.77

Asghar, M. Z., Kundi, F. M., Ahmad, S., Khan, A., & Khan, F. (2018, February). T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme. *Expert Systems, Wiley Online Library*, *35*(1), e12233. Retrieved 2018-11-06, from https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12233 doi: 10.1111/exsy.12233

Beigi, G., Hu, X., Maciejewski, R., & Liu, H. (2016). An Overview of Sentiment Analysis in Social Media and Its Applications in Disaster Relief. In W. Pedrycz & S.-M. Chen (Eds.), *Sentiment Analysis and Ontology Engineering* (Vol. 639, pp. 313–340). Cham: Springer International Publishing. Retrieved 2018-10-27, from http://link.springer.com/10.1007/978-3-319-30319-2_13 doi: 10.1007/978-3-319-30319-2_13

Bekkerman, R., & Allan, J. (2003, December). *Using Bigrams in Text Categorization.* Department of Computer Science, University of Massachusetts, Amherst, 01003 USA. (Unpublished Paper)

Chandrashekar, G., & Sahin, F. (2014, January). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. Retrieved 2018-12-06, from https://linkinghub.elsevier.com/retrieve/pii/S0045790613003066 doi: 10.1016/j.compeleceng.2013.11.024

Choudhary, R., & Sain, D. (2016, December). An Improved Approach for Twitter Data Analysis using Clustering and J48 Classification. *International Journal of Computer Applications*, *156*(14), 35–41. Retrieved

2018-12-01, from http://www.ijcaonline.org/archives/volume156/number14/choudhary-2016-ijca-912562.pdf doi: 10.5120/ijca2016912562

Chowdhury, S. R., Asghar, M. R., Amer-yahia, S., Comes, T., Fiedrich, F., Fortier, S., ... Castillo, C. (2013, May). 1 Tweet4act: Using Incident-Specific Profiles for Classifying Crisis-Related Messages. In *Proceedings of the 10th international iscram conference* (pp. 834–839).

D'Andrea, A., Ferri, F., Grifoni, P., & Guzzo, T. (2015, September). Approaches, Tools and Applications for Sentiment Analysis Implementation. *International Journal of Computer Applications*, *125*(3), 26–33. Retrieved 2018-11-25, from http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf doi: 10.5120/ijca2015905866

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. W. (2007). Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07* (p. 230). San Jose, California, USA: ACM Press. Retrieved 2018-12-07, from http://portal.acm.org/citation.cfm?doid=1281192.1281220 doi: 10.1145/1281192.1281220

da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014, October). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170–179. Retrieved 2018-12-01, from https://linkinghub.elsevier.com/retrieve/pii/S0167923614001997 doi: 10.1016/j.dss.2014.07.003

Dattu, B. S., & Gore, D. V. (2015). A Survey on Sentiment Analysis on Twitter Data Using Different Techniques. In *International journal of computer science and information technologies* (Vol. 6, pp. 5358–5362).

Garnes, O. (2009). *Feature Selection for Text Categorization* (Unpublished master's thesis). Faculty of the Graduate School of The Norwegian University of Science and Technology.

Gharavi, E., & Bijari, K. (2017). Short text classification using deep representation: A case study of Spanish tweets in Coset Shared Task. In *Proceedings of the second workshop on evaluation of human language technologies for iberian languages (ibereval 2017)* (pp. 28–35).

Go, A., Bhayani, R., & Huang, L. (2010). *Twitter Sentiment Classification using Distant Supervision.* Stanford University, Stanford, CA 94305. (Unpublished Paper)

Hammer, H., Yazidi, A., Bai, A., & Engelstad, P. (2015). Building Domain Specific Sentiment Lexicons Combining Information from Many Sentiment Lexicons and a Domain Specific Corpus. In A. Amine, L. Bellatreche, Z. Elberrichi, E. J. Neuhold, & R. Wrembel (Eds.), *Computer Science and Its Applications* (Vol. 456, pp. 205–216). Cham: Springer International Publishing. Retrieved 2018-12-04, from `http://link.springer.com/10.1007/978-3-319-19578-0_17` doi: 10.1007/978-3-319-19578-0_17

Imran, M., Alam, F., Ofli, F., & Aupetit, M. (2017, December). Enabling Rapid Disaster Response Using Artificial Intelligence and Social Media. *International Roundtable on the Impact of Extreme Natural Events:Science and Technology for Mitigation-2017, IRENE 2017*, 12.

Imran, M., & Castillo, C. (2015). Towards a Data-driven Approach to Identify Crisis-Related Topics in Social Media Streams. In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion* (pp. 1205–1210). Florence, Italy: ACM Press. Retrieved 2018-12-01, from `http://dl.acm.org/citation.cfm?doid=2740908.2741729` doi: 10.1145/2740908.2741729

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015, July). Processing Social Media Messages in Mass Emergency: A Survey. *ACM Computing Surveys (CSUR)*, *47*, 1-37. Retrieved 2018-11-09, from `http://arxiv.org/abs/1407.7071`

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2018). Processing Social Media Messages in Mass Emergency: Survey Summary. In *Companion of the The Web*

Conference 2018 - WWW '18 (pp. 507–511). Lyon, France: ACM Press. Retrieved 2018-11-09, from http://dl.acm.org/citation.cfm?doid=3184558.3186242 doi: 10.1145/3184558.3186242

Imran, M., Castillo, C., Lucas, J., Meier, P., & Vieweg, S. (2014). AIDR: artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion* (pp. 159–162). Seoul, Korea: ACM Press. Retrieved 2018-12-01, from http://dl.acm.org/citation.cfm?doid=2567948.2577034 doi: 10.1145/2567948.2577034

Imran, M., Elbassuoni, S., Castillo, C., & Diaz, F. (2013). Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion* (pp. 1021–1024). Rio de Janeiro, Brazil: ACM Press. Retrieved 2018-10-09, from http://dl.acm.org/citation.cfm?doid=2487788.2488109 doi: 10.1145/2487788.2488109

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013, May). Extracting Information Nuggets from Disaster- Related Messages in Social Media. In *Proceedings of the 10th international iscram conference* (p. 10). Baden, Germany.

Imran, M., Mitra, P., & Castillo, C. (2016, May). Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. In *10th language resources and evaluation conference (lrec), 2016* (p. 6). Retrieved 2018-11-21, from http://arxiv.org/abs/1605.05894 (arXiv: 1605.05894)

Jaderberg, D. (2016). *Sentiment and topic classification of messages on Twitter* (Doctoral dissertation, Uppsala Universitet, Sweden). Retrieved 2018-11-04, from http://uu.diva-portal.org/smash/get/diva2:929542/FULLTEXT01.pdf

Jovic, A., Brkic, K., & Bogunovic, N. (2015, May). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1200–1205). Opatija, Croatia: IEEE. Retrieved 2018-12-07, from http://ieeexplore.ieee.org/document/7160458/ doi: 10.1109/MIPRO.2015.7160458

Kreutz, T., & Daelemans, W. (2018, August). Enhancing General Sentiment Lexicons for Domain-Specific Use. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1056–1064). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved 2018-11-04, from `http://www.aclweb.org/anthology/C18-1090`

Kumar, A., & Sebastian, T. M. (2012, July). Sentiment Analysis on Twitter. *International Journal of Computer Science Issues, IJCSI*, *9*(4), 7.

Labille, K., Gauch, S., & Alfarhood, S. (2017, August). Creating Domain-Specific Sentiment Lexicons via Text Mining. In *Proceedings of workshop on issues of sentiment discovery and opinion mining, (WISDOM17)* (p. 8). Halifax, Canada.

Li, P., He, J.-Q., & Ma, C.-L. (2017, March). Short Text Classification Based on Latent Topic Modeling and Word Embedding. *DEStech Transactions on Computer Science and Engineering*(aice-ncs). Retrieved 2018-12-07, from `http://dpi-proceedings.com/index.php/dtcse/article/view/5647` doi: 10.12783/dtcse/aice-ncs2016/5647

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015, July). Support vector machines and Word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)* (pp. 136–140). Beijing, China: IEEE. Retrieved 2018-12-07, from `http://ieeexplore.ieee.org/document/7259377/` doi: 10.1109/ICCI-CC.2015.7259377

MacEachren, A. M., Jaiswal, A., Robinson, A. C., Pezanowski, S., Savelyev, A., Mitra, P., ... Blanford, J. (2011, October). SensePlace2: GeoTwitter analytics support for situational awareness. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 181–190). Providence, RI, USA: IEEE. Retrieved 2018-12-22, from `http://ieeexplore.ieee.org/document/6102456/` doi: 10.1109/VAST.2011.6102456

Minocha, A. (2012, September). Generating domain specific sentiment lexicons using the Web Directory. *Advanced Computing: An International Journal*, *3*(5), 45–

51. Retrieved 2018-11-04, from http://www.airccse.org/journal/acij/papers/3512acij05.pdf doi: 10.5121/acij.2012.3505

Nguyen, D. T., Alam, F., Ofli, F., & Imran, M. (2017). Automatic Image Filtering on Social Networks Using Deep Learning and Perceptual Hashing During Crises. In *Proceedings of the 14th iscram conference, core paper: Social media studies* (p. 13). Retrieved 2018-12-01, from http://arxiv.org/abs/1704.02602 (arXiv: 1704.02602)

Nguyen, D. T., Joty, S., Imran, M., Sajjad, H., & Mitra, P. (2016, October). Applications of Online Deep Learning for Crisis Response Using Social Media Information. In *4th international workshop on social web for disaster management (swdm), co-located with the 25th conference of information and knowledge management (cikm)* (p. 6).

Nguyen, D. T., Mannai, K. A. A., Joty, S., Sajjad, H., Imran, M., & Mitra, P. (2017). Robust Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks. In *Association for the advancement of artificial intelligence, AAAI, 2017* (p. 4).

Nguyen, D. T., Ofli, F., Imran, M., & Mitra, P. (2017). Damage Assessment from Social Media Imagery Data During Disasters. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM '17* (pp. 569–576). Sydney, Australia: ACM Press. Retrieved 2018-10-09, from http://dl.acm.org/citation.cfm?doid=3110025.3110109 doi: 10.1145/3110025.3110109

Ofli, F., Meier, P., Imran, M., Castillo, C., Tuia, D., Rey, N., ... Joost, S. (2016, March). Combining Human Computing and Machine Learning to Make Sense of Big (Aerial) Data for Disaster Response. *Big Data*, *4*(1), 47–59. Retrieved 2018-12-01, from http://www.liebertpub.com/doi/10.1089/big.2014.0064 doi: 10.1089/big.2014.0064

Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014). CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In *Proceedings of the eighth international aaai conference on weblogs and social media* (p. 10).

Palen, L., & Liu, S. B. (2007, May). Citizen Communications in Crisis: Anticipating a Future of ICT-Supported Public Participation. In *Conference on human factors in computing systems, chi 2007 proceedings: Emergency action* (pp. 727–736). San Jose, California, USA.

Pawar, K. K., Shrishrimal, P. P., & Deshmukh, R. R. (2015). Twitter Sentiment Analysis: A Review. *International Journal of Scientific Engineering Research*, *6*(4), 8.

Psomakelis, E., Tserpes, K., Anagnostopoulos, D., & Varvarigou, T. (2014). Comparing Methods for Twitter Sentiment Analysis:. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (pp. 225–232). Rome, Italy: SCITEPRESS - Science and and Technology Publications. Retrieved 2018-11-05, from `http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005075302250232` doi: 10.5220/0005075302250232

Rogati, M., & Yang, Y. (2002, November). High-Performing Feature Selection for Text Classification. In *Acm conference on information and knowledge management (cikm)* (pp. 659–661). McLean, VA.

Sahoo, S. K. (2017, September). *Message Classification for Twitter Data* (Short Project). Bihar, India: IIT PATNA, Patna, Bihar, India. Retrieved 2018-12-01, from `http://www.iitp.ac.in/~arijit/dokuwiki/lib/exe/fetch.php?media=courses:2017:cs551:08_report.pdf`

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Acm international world wide web conference, www, 2010* (pp. 851–860). Raleigh, North Carolina, USA.

Scott, S., & Matwin, S. (2001). *Feature Engineering for Text Classification.* Canada National Research Council, Interactive Information Group. (Unpublished Paper)

Shrestha, A., Miller, B., Zhu, Y., & Zhao, Y. (2013). Storygraph: extracting patterns from spatio-temporal data. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics - IDEA '13* (pp. 95–103). Chicago, Illinois: ACM Press. Retrieved 2018-11-04, from `http://dl.acm.org/citation.cfm?doid=2501511.2501525` doi: 10.1145/2501511.2501525

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2014, July). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.* Stanford University, Stanford, CA 94305, USA. (Unpublished Paper)

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10* (p. 841). Geneva, Switzerland: ACM Press. Retrieved 2018-12-01, from `http://portal.acm.org/citation.cfm?doid=1835449.1835643` doi: 10.1145/1835449.1835643

Stowe, K., Paul, M. J., Palmer, M., Palen, L., & Anderson, K. (2016, November). Identifying and Categorizing Disaster-Related Tweets. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media* (pp. 1–6). Austin, TX, USA: Association for Computational Linguistics. Retrieved 2018-10-04, from `http://aclweb.org/anthology/W16-6201`

Tan, C.-M., Wang, Y.-F., & Lee, C.-D. (2002, July). The use of bigrams to enhance text categorization. *Information Processing & Management*, *38*(4), 529–546. Retrieved 2018-12-18, from `http://linkinghub.elsevier.com/retrieve/pii/S0306457301000450` doi: 10.1016/S0306-4573(01)00045-0

Temnikova, I., & Castillo, C. (2015). EMTerms 1.0: A Terminological Resource for Crisis Tweets. In *Proceedings of the iscram 2015 conference* (p. 13). Kristiansand, Norway.

Thangaraj, M., & Sivakami, M. (2018). Text Classification Techniques: A Literature Review. *Interdisciplinary Journal of Information, Knowledge, and Management*, *13*, 117–135. Retrieved 2018-12-09, from `https://www.informingscience.org/Publications/4066` doi: 10.28945/4066

Truong, B., Caragea, C., Squicciarini, A., & Tapia, A. H. (2014). Identifying Valuable Information from Twitter During Natural Disasters. *Proceedings of the American Society for Information Science and Technology*, *51*(1), 1–4. Retrieved 2018-10-04, from `http://doi.wiley.com/10.1002/meet.2014.14505101162` doi: 10.1002/meet.2014.14505101162

Vieweg, S., Castillo, C., & Imran, M. (2014). Integrating Social Media Communications into the Rapid Assessment of Sudden Onset Disasters. In L. M. Aiello & D. McFarland (Eds.), *Social Informatics* (Vol. 8851, pp. 444–461). Cham: Springer International Publishing. Retrieved 2018-12-01, from `http://link.springer.com/10.1007/978-3-319-13734-6_32` doi: 10.1007/978-3-319-13734-6_32

Wakade, S., Shekar, C., Liszka, K. J., & Chan, C.-C. (2012, March). *Text Mining for Sentiment Analysis of Twitter Data.* The University of Akron, Department of Computer Science, Ohio, United States. (Unpublished Paper)

Wang, S., & Manning, C. D. (2010). *Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.* Department of Computer Science, Stanford University, Stanford, CA 94305. (Unpublished Paper)

Wang, W. (2014). *Automated spatiotemporal and semantic information extraction for hazards* (Unpublished doctoral dissertation). Faculty of the Graduate College of The University of Iowa.

Wang, W., & Stewart, K. (2015, March). Spatiotemporal and semantic information extraction from Web news reports about natural hazards. *Computers, Environment and Urban Systems*, *50*, 30–40. Retrieved 2018-10-03, from https://linkinghub.elsevier.com/retrieve/pii/S0198971514001252 doi: 10.1016/j.compenvurbsys.2014.11.001

Whipple, A. L. (2017). *Comparison of Algorithms in Twitter Sentiment Analysis* (Master's thesis, Faculty of the Graduate School of The University of Texas at Austin). Retrieved 2018-11-05, from https://repositories.lib.utexas.edu/bitstream/handle/2152/60372/WHIPPLE-MASTERSREPORT-2017.pdf?sequence=1&isAllowed=y

Appendix A

Additional content

This section presents code, figures, tables and other work that was conducted as a part of the study but hasn't been included in the chapters of this report.

A.1 Java code to include additionally generated features in the twitter dataset ARFF file

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

public class MergeFiles
{
  static ArrayList<String> termScoreData = new ArrayList
  <String>();
  static ArrayList<String> columnData = new ArrayList
  <String>();
```

```
  public static void main(String... args) {

// Input file names (headers are removed)

  String termScoreFile = "Final_No_Header.csv";
  String columnFile = "Full_Data_No_Header.csv";

// Important to continue the sequence of attribute numbers

   int attributeIndex = 381;

   merge(termScoreFile, columnFile, attributeIndex);

// Output file name

   String outputFile = "HeaderLessMerged.arff";
   writeToFile(outputFile);
   }

    public static void merge(String f1, String f2, int idx){

        BufferedReader br = null;
        String line = "";
        int count = 0;
        try{
            br = new BufferedReader(new FileReader(f1));

            while ((line = br.readLine()) != null)
            {
```

```
                int temp = line.indexOf('}');
                termScoreData.add(count, line.substring
                (0, temp) + ",");
                count++;
            }
            br = new BufferedReader(new FileReader(f2));

            count = 0;
            while ((line = br.readLine()) != null)
            {
                                int index = idx;
                int temp = line.indexOf(',');
                line = line.substring(temp + 1, line.length());

// Used a Regular Expression because 1 line with comma in text
was causing an issue

                String[] tokens = line.split(",(?=(?:[^\']*\'
                [^\']*\')*[^\']*$)", -1);

                StringBuilder builder = new StringBuilder();

// Don't include category label and duplicated tweet text

// Remember to exclude 'translation' attribute from header


                for(int i = 2; i < tokens.length; i++){
                  builder.append(index + " " + tokens[i] + ",");
                  index++;
```

```
                }
// Important Experimenter needs category label tokens[0] in the
last position

// Remember to adjust header to take this change into account!!

                builder.append(index + " " + tokens[0]);
                builder.append("}");

                columnData.add(count, builder.toString());
                count++;
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally{
            try{
                br.close();
            }
            catch(IOException ie){
                ie.printStackTrace();
            }
        }
    }

    public static void writeToFile(String f){

        PrintWriter out = null;
```

```
        try{
            out = new PrintWriter(f);

// Just joining the two lines into one

            for(int j = 0; j < termScoreData.size(); j++){
              out.println(termScoreData.get(j) +
              columnData.get(j));
            }
        }

        catch(Exception e){

            System.out.println(e.toString());
        }
        finally{
            try{
                out.flush();
                out.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

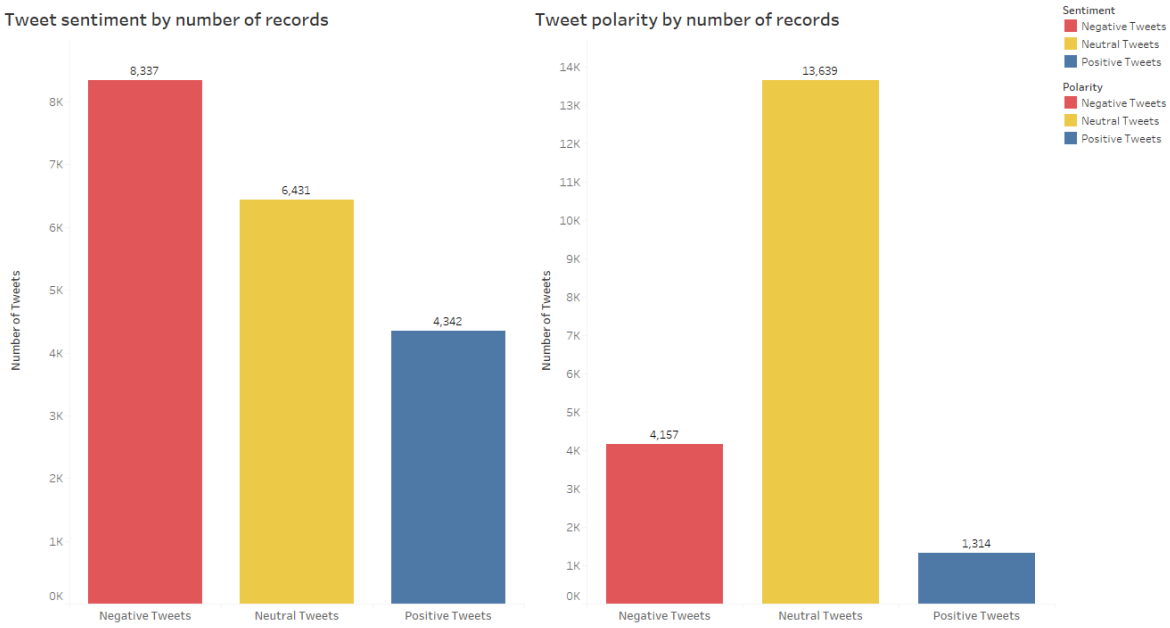


Figure A.1: Analyzing sentiments using Rosette and AYLIEN: Left and right sides represent output from Rosette and AYLIEN respectively.

A.2 Comparing the performance of AYLIEN and Rosette Text Analysis Extension for Sentiment Analysis

Figure A.1 compares the sentiment polarity of tweets developed using two different extensions of Rapid Miner Data Analysis Tool. The left side of the figure shows the use of Rosette Text Analytics package to perform the sentiment polarity while the right side of the figure shows the use of AYLIEN Text Analysis package. It can be seen that Rosette has classified most tweets into Negative followed by Neutral and Positive while AYLIEN has classified most tweets into Neutral followed by some Negative and extremely few Positive Tweets. This is due to the weight each tool gives to the vocabulary of sentiment words.

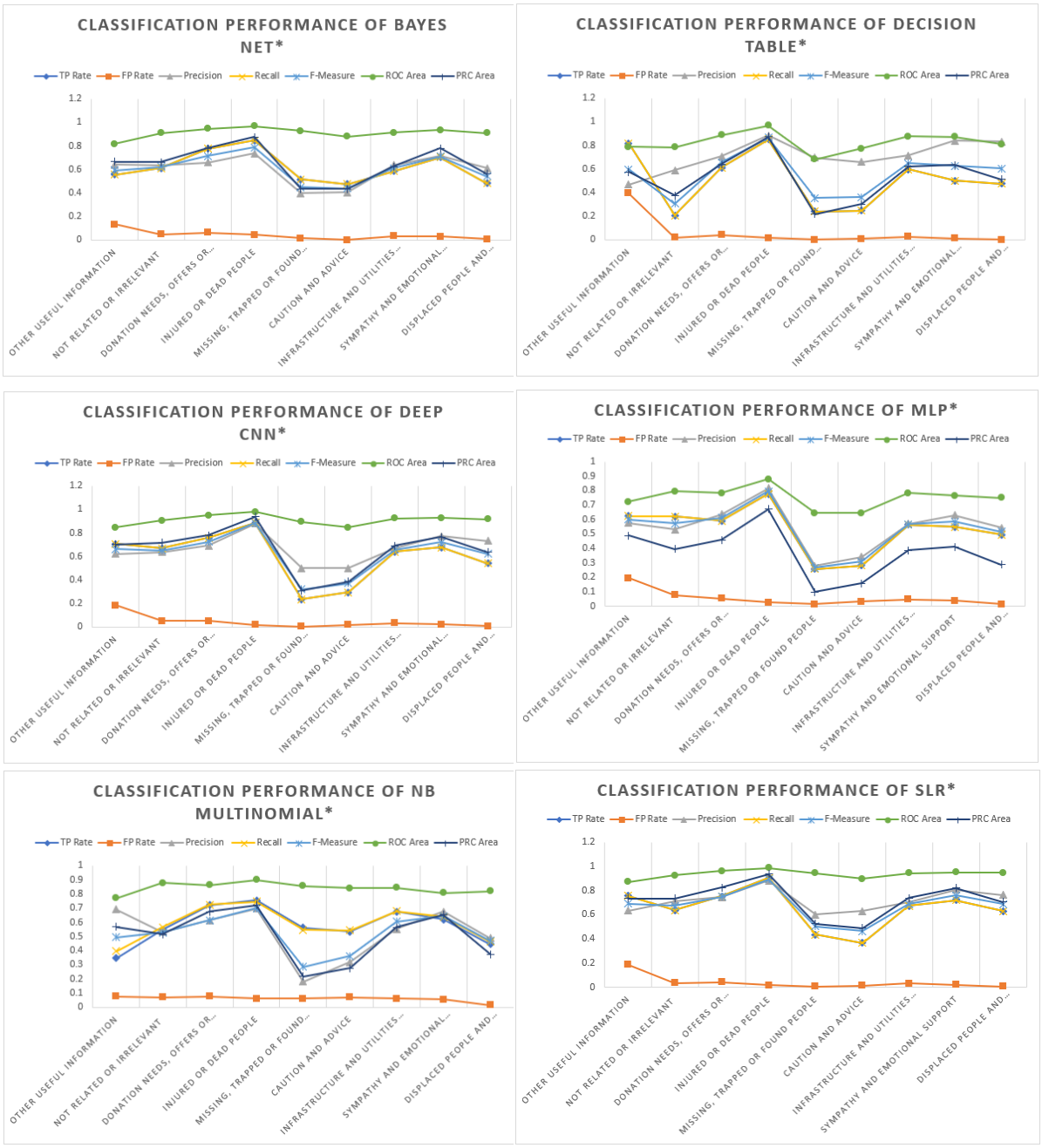


Figure A.2: Classification Performance of Different Classifiers using Enhanced Dataset

A.3 Classification Results using Enhanced Dataset

Figure A.2 provides the performance of text classification on Enhanced Dataset using different machine learning classifiers. Certain classes like *Missing, Trapped or Found People* & *Caution and Advice* have the lowest performance for all classifiers.