Cross-Language Tweet Classification using Bing translator

by

Bhavani Krithivasan

M.S., VIT University, 2012

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Social media affects our daily lives. It is one of the first sources for finding breaking news. In particular, Twitter is one of the popular social media platforms, with around 330 million monthly users. From local events such as Fake Patty's Day to across the world happenings - Twitter gets there first. During a disaster, tweets can be used to post warnings, status of available medical and food supply, emergency personnel, and updates. Users were practically tweeting about the Hurricane Sandy, despite lack of network during the storm. Analysis of these tweets can help monitor the disaster, plan and manage the crisis, and aid in research.

In this research, we use the publicly available tweets posted during several disasters and identify the relevant tweets. As the languages in the datasets are different, Bing translation API has been used to detect and translate the tweets. The translations are then, used as training datasets for supervised machine learning algorithms. Supervised learning is the process of learning from a labeled training dataset. This learned classifier can then be used to predict the correct output for any valid input. When trained to more observations, the algorithm improves its predictive performance.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I wish to express my most profound gratitude to my advisor, Dr. Doina Caragea for the continuous support of my research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me throughout my Masters. I could not have imagined having a better advisor.

Besides my advisor, I would like to thank the rest of my Master's committee: Dr. Torben Amtoft, and Dr. Mitchell L. Neilsen, for their encouragement and advice.

Words are inadequate in offering my thanks to the faculty and the staff of the Computer Science department for their unyielding support, understanding, and cooperation whenever I needed it.

Last, but not the least, I would like to thank my family: my husband, Hariharan Thiagarajan, my parents S.Krithivasan and Ganga Krithivasan, and my sister, Krithika Krithivasan, for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

Chapter 1

Introduction

1.1 Context

Social networks are web platforms that allow users to create an account, establish a profile, and interact with other members. Most social networks permit users to manage their privacy and preferences for sharing content and personal information. Examples of social networks include Facebook, Google+, LinkedIn, MySpace, and Twitter [DHS, 2013].

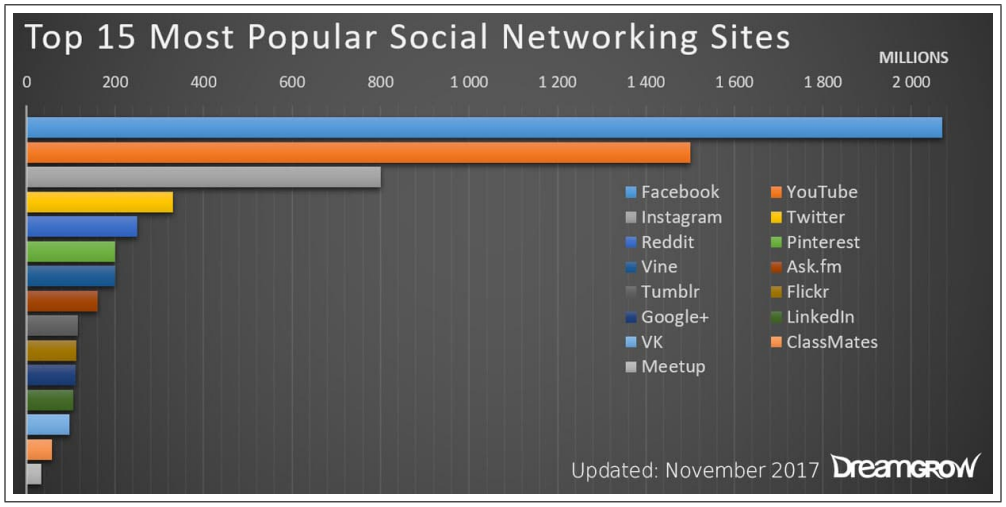

Figure 1.1: 15 Most Popular Social Media Sites [Kallas, 2017]

"When disasters happen, people need to know their loved ones are safe. It's moments like this that being able to connect really matters." - Mark Zuckerberg, CEO Facebook [Adim-

bola, 2017]. As soon as we wake up in the morning, we check our phones for messages, tweets, and trending news. Social media has changed from a tool to post about ourselves to a mechanism where you get the current news along with the vibe of the people. It became a medium through which we get information from the people who are at the center of an event.

In the current technological world, one can receive updates in real-time, whenever a calamity like a natural disaster or a terrorist attack occurs. Pekar et al. [2016]believes that the capability of emergency services can be improved by building infrastructure that collects data from the affected people through social media, making them better equipped to detect disasters at early stages, monitor their development and tackle their consequences in the recovery operations.

An infographic released by The University of San Francisco's Online Masters of Public Administration program called "Social Media: The New Face of Disaster Response" provides statistics to help us understand the extent to which crisis and usage of social media are linked [LePage, 2013].

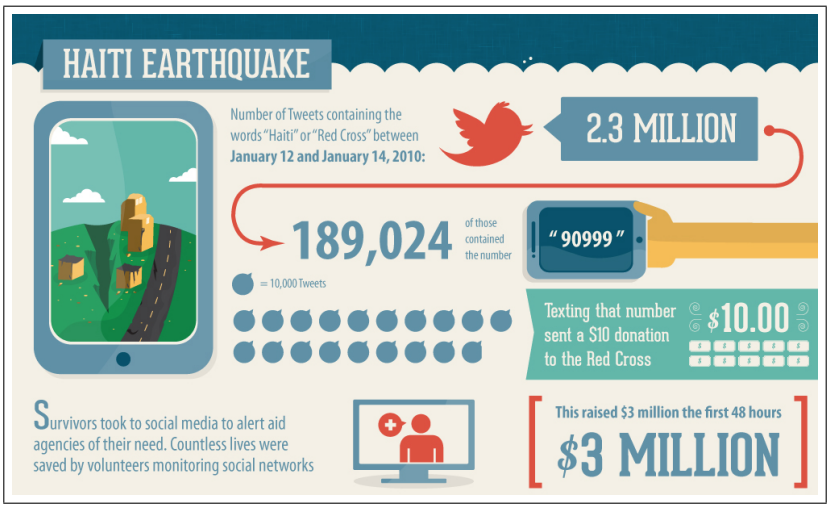

Figure 1.2: Haiti Earthquake [LePage, 2013]

In large-scale events like floods, earthquakes, hurricanes, it is vital to convey information about the situation. According to the U.S. Department of Homeland Security (DHS), "Social media and collaborative technologies have become critical components of emergency

preparedness, response, and recovery" [DHS, 2013].

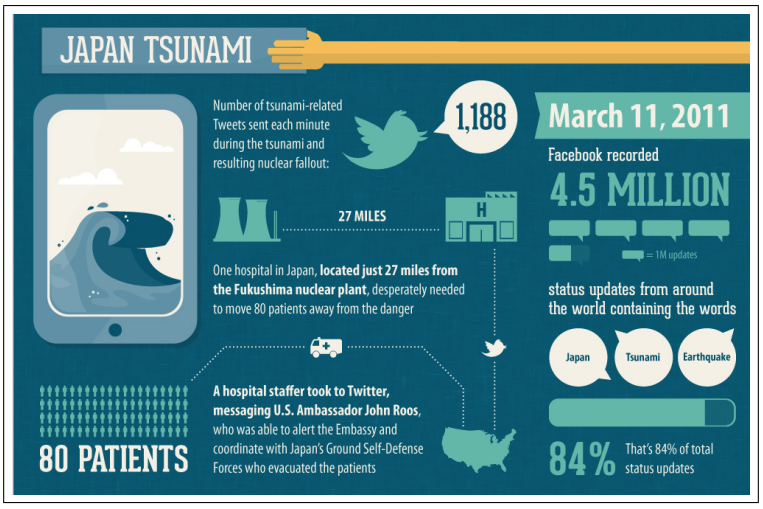

Figure 1.3: Japan Tsunami [LePage, 2013]

It is reported that around ten pictures per second were uploaded during Hurricane Sandy on Instagram [Holtz, 2012]. In fact, almost half the respondents, in a recent survey, said they would use social media in the event of a disaster to let relatives and friends know they were safe [OĎell, 2011].

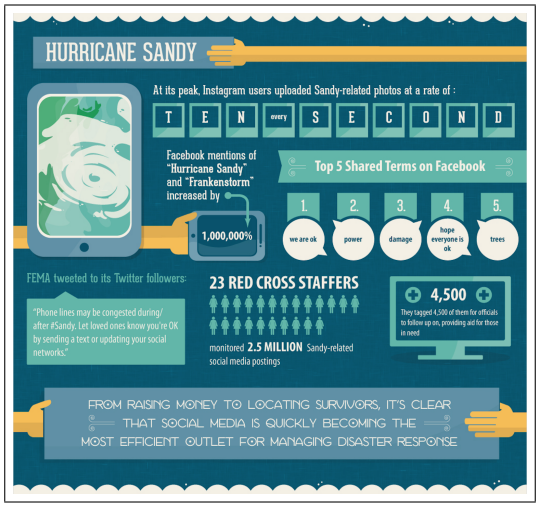

Figure 1.4: Hurricane Sandy [LePage, 2013]

1.2 Background

1.2.1 Twitter

Twitter allows its users to express in short text messages, called tweets, of up to 280 characters. These tweets are used to broadcast relevant information and report news of emergency situations. A total of 500 million tweets are sent per day [Internet Live Stats]. Huge amount of data are available to consume due to the increase in the growth of social media. Evaluating and assimilating the ocean of information is not possible by humans, it requires capabilities that only a machine can handle.

Not all of these datasets contain disaster-specific information. Based on the analysis of Gupta and Kumaraguru [2012], only 17% of the total tweets posted about the event held situational awareness information that was credible. There is still needed to develop the technologies for filtering and retrieving the informational or reliable tweets automatically during disasters. There are three problems involved in developing a system that can classify tweets as originating in social media applications into specific information categories. The first issue is to deal with the massive amount of tweets arriving per minute. The second is effective features extraction for noisy and not curated short text messages [Kumar, 2015]. Tweets are highly varied regarding subject and content, and the influx of tweets particularly in the event of a disaster may be overwhelming. It is impractical to classify these various tweets to extract needed information automatically. Tweets classification is, therefore, the third challenge.

1.3 Problem Description

Labeled data is required to classify tweets through text classification to extract information during a disaster. Data is usually labeled manually by humans using the crowdsourcing model, which is time-consuming. Also, there are instances where the crowdsourced labels might not be accurate. Another concern is that a classifier trained for a specific disaster might not function on a different disaster as the individual words obtained from each disaster can

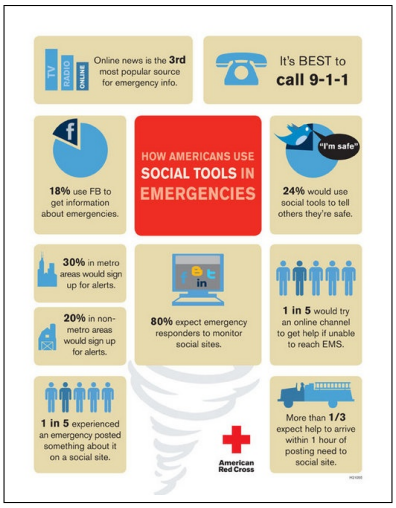

Figure 1.5: Usage of Social Tools in Emergencies (2011) [Fox, 2011]

vary as could the primary language of the tweets from a disaster.

Thus, the questions that need to be answered are

- Is it possible to use a classifier trained in English to label translated tweets?
- How to learn from a different language (Spanish / French / Chinese, etc.) and how does it compare against a classifier learned from English?

Chapter 2

Related Work

In this chapter, we discuss some of the previous work that has been done in the field of domain adaptation and review some of the relevant research papers.

The information detection and extraction system for microblog posts were described by [Imran et al., 2013]. In their work, Naive Bayesian classifiers were used to classify a tweet into one of the types such as Caution and Advice, Informative source, Donation, and Causalities & damage. Gupta et al. [2014] also provided an SVM-rank based system, TweetCred to assign a credibility score to tweets in a user's timeline. According to the literatures, supervised machine learning algorithms have been applied by most of the researchers to detect and classify the content in Online Social Media. Naive Bayes (NB) and Support Vector Machine (SVM) are used for tweets classification in [Parilla-Ferrer et al., 2014].

Also, the idea of learning from multiple sources is researched by Wu et al. [2016] in the area of sentiment classification. [Wu et al., 2016] propose a new domain adaptation approach which can exploit sentiment knowledge from multiple source domains. They first extract both global and domain-specific sentiment knowledge from the data of multiple source domains using multi-task learning. Then, they transfer the knowledge from source domains to target domain with the help of words sentiment polarity relations extracted from the unlabeled target domain data. The authors state that experimental results show the effectiveness of the approach in improving cross-domain sentiment classification performance. However,

their approach is not entirely transferable to other problems. The reason is that it might be difficult to apply their method to other datasets because we would first need to build a sentiment word graph, on which the technique heavily relies, and this is not scalable and not trivial.

There has been some research done in the area of disaster management using tweets, by Li et al. [2015]and by Imran et al. [2016], among others.

Li et al. [2015] study the effectiveness of labeled data from a prior source domain, together with unlabeled data from the current target domain to learn domain adaptation classifiers for the target. Results indicate that, the source is sufficient to classify target data. However, to classify a particular disaster, domain adaptation techniques that use unlabeled data from the target in addition to labeled data from the source are the best.

Imran et al. [2016] analyze the performance of the classifiers trained using different combinations of training sets obtained from past disasters. Their experiments show that the annotations are useful when the source and the target are of the same crisis (For example Hurricanes). Performance of cross-language domain adaptation decreases when different languages are used in the experiments.

Chapter 3

Experimental Design

In the following sections, I present the dataset used in the experiment, then discuss the translation API employed to identify and translate the tweets, followed by the data extraction and cleaning process. I conclude the chapter by describing the various steps involved in data analysis.

3.1 Methods

3.1.1 Weka

A software called Weka was used to preprocess and analyze the Twitter data. Weka is an acronym for *Waikato Environment for Knowledge Analysis* and is developed by The University of Waikato in New Zealand. The software is developed in Java and distributed under the terms of the GNU General Public License. It can be run on multiple platforms like Linux, Windows, and Macintosh operating systems.

Weka contains several machine learning algorithms that can be used to preprocess or filter the data, apply one or more algorithms to classify or cluster and analyze the results using any one of the options available - GUI, Command Line Interface or integrate with one of the several supported programming languages. [Frank et al., 2016]

I have used the three options at various stages of the project, based on the need and

comfortability. Most of the preprocessing work has been completed using the command line, while the classifier has been implemented in Java.

3.1.2 Naive Bayes Classifier

Naive Bayes algorithm is a confirmation that sometimes the simplest solution is the most powerful solution. [Stecanella, 2017]

Naive Bayes is a probabilistic classifier that makes use of probability theory and Bayes Theorem to predict the class of a sample. Naive Bayes classifier assumes that the features are independent of each other.

A Naive Bayes Classifier predicts the class value when a set of set of attributes is provided. For each known class value, [Thornton, 2017]

- Calculate probabilities for each attribute, conditional on the class value.
- Use the product rule to obtain a joint conditional probability for the attributes.
- Use Bayes rule to derive conditional probabilities for the class variable.

The class with the highest probability is deemed as the most likely class. According to Bayes theorem, the probability is calculated as shown below

$$P(Y = y_i|X = x_k) = \frac{P(X = x_k|Y = y_i)P(Y = y_i)}{\sum_j P(X = x_k|Y = y_j)P(Y = y_j)} \tag{3.1}$$

where Y is the set of classifications $y_1, y_2,...y_m$ and X is the set of features $x_1, x_2,...x_n$.

3.2 Dataset

For this project, CrisisLexT26 [Olteanu et al., 2015] data has been used. The dataset contained around 25K tweets from as many as 26 crises between the year 2012 and 2013, with roughly 1500 tweets per crisis.

Crowdsource workers labeled the tweets according to informativeness (informative or not informative), information types (e.g., caution and advice, infrastructure damage), and information sources (e.g., governments, NGOs). Labels were decided based on the majority voting among at least three crowdsourcing workers.

Tweets collected for a particular crisis are in a specific CSV file. Thus, each file contained one tweet per line with the following comma-separated fields: Tweet ID, Tweet Text, Information Source, Information Type, Informativeness. The file contained labels provided by crowdsource workers, indicating if the tweet is:

- Informativeness : Related and informative, Related - but not informative, Not related, Not applicable
- Information source : Eyewitness, Government, NGOs, Business, Media, Outsiders, Not applicable
- Information type : Affected individuals, Infrastructure, and utilities, Donations, and volunteering, Caution and advice, Sympathy and support, Other Useful Information, Not applicable

When the decision makers know whether the tweets are relevant or not, eyewitness based or not, useful information or not; they will be better enabled to make a collective decision on handling the situation. In this project, I concentrated on the source of information, i.e., whether the tweet is from an individual who has been affected or was present at the time of the disaster. These sources are categorized into two types:

Primary Sources Eyewitness accounts

Secondary/Tertiary Sources Mainstream media, Government officials, NGOs, etc.

3.3 Data Translation

As the tweets were collected from different crisis across the world, they were in different languages. Microsoft Translator Text API, Microsoft machine translation services, was used

Source:	
Eyewitness	citizen reporters, members of the community [29]; eyewitnesses [6, 14, 27, 34]; local, peripheral, personally connected [45]; local individuals [43, 50]; local perspective, on the ground reports [46]; direct experience (personal narrative and eyewitness reports) [40]; direct observation, direct impact, relayed observation [48];
Government	(news organizations and) authorities [29]; government/administration [34]; police and fire services [22]; police [13]; government [6]; public institutions [46]; public service agencies, flood specific agencies [45];
NGOs	non-profit organizations [12, 46]; non-governmental organization [34]; faith-based organizations [45];
Business	commercial organizations [12]; enterprises [46]; for-profit corporation [34];
Media	news organizations (and authorities), blogs [29]; journalists, media, and bloggers [12, 14]; news organization [34]; professional news reports [28]; media [6]; traditional media (print, television, radio), alternative media, freelance journalist [46]; blogs, news-crawler bots, local, national and alternative media [45]; media sharing (news media updates, multimedia) [40];
Outsiders	sympathizers [27]; distant witness [9]; remote crowd [43]; non-locals [45, 46].

Figure 3.1: Types of Information Sources

to identify the languages and determine their frequencies. Microsoft Translator Text API uses Statistical Machine Translation (SMT) to detect the correspondences between source and target language and find the best translation. The top 8 languages are as follows:

Language	Count
English	20689
Spanish	3240
Italian	1632
Portuguese	635
Filipino	594
French	517
Russian	261
Japanese	108
Indonesian	79
Bangla	33

Table 3.1: Top 10 languages in CrisisLexT26

3.4 Extraction of relevant datasets

Four subsets of the CrisisLexT26 dataset mentioned in 3.3 were created. They are

- D1 : All the earthquake tweets in Spanish except Costa Rica

- D2 : All the floods tweets in English except Colorado
- D3 : All the Spanish tweets
- D4 : All the English tweets

The label, Information Source, was converted to binary values, as Eyewitness and NotEyewitness (Government, NGOs, Business, Media, Outsiders).

Dataset No.	Language	Eyewitness	Not Eyewitness	Total
D1	Spanish (Earthquakes)	14	499	513
D2	English (Floods)	638	2092	2730
D3	Spanish	117	3123	3240
D4	English	1783	18906	20689

Table 3.2: Subsets of CrisisLexT26

3.5 Data Preprocessing

To ensure that the dataset can be interpreted by the machine learning algorithms for analysis, preprocessing is performed to provide an organized structure. Preprocessing is done as follows [Li et al., 2015]:

- Remove non-printable, ASCII characters
- Convert printable HTML entities to their corresponding ASCII equivalents
- Replaced URLs, email addresses, and usernames with a URL / email/username placeholder for each type of entity
- Remove retweets, as they might not be informative for the classification task
- Remove duplicate tweets and empty tweets (no characters after cleaning)

The total number of tweets for each dataset, after preprocessing is shown in the table 3.3.

Dataset No.	Language	Eyewitness	Not Eyewitness	Total
DS1	Spanish (Earthquakes)	14	499	513
DS2	English (Floods)	638	2092	2730
DS3	Spanish	117	2803	2920
DS4	English	1652	17110	18762

Table 3.3: Subsets of CrisisLexT26, after preprocessing

3.6 ARFF Datasets Creation

3.6.1 Conversion to ARFF

As mentioned earlier in the section 3.1.1, Weka has been used as a medium to implement the classifier. Before starting the classification, the CSV files needed to be converted to a format that Weka understands: Attribute-Relation File Format (ARFF). ARFF is an ASCII file that contains a list of values (instances) along with their attributes. ARFF files have two sections - Header section and a Data section.

The Header contains the following details

- relation name,
- list of features representing the instances (attributes) along with their datatype

The Data section of the file contains the instances in the order of the attributes listed, and the class is the last attribute. Each line indicates an instance or tweets in this case. Attribute values for each instance are separated by commas [Frank et al., 2008]. The preprocessed files can be converted to ARFF using the following command [Frank et al., 2016].

java weka.core.converters.TextDirectoryLoader -dir text_example >text_example.arff

```
@relation weather.symbolic


@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
```

```
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
```

Listing 3.1: Sample ARFF file

3.6.2 Creation of Training and Test Datasets

As mentioned in section 3.5, there were four sets of data to begin with (D1, D2, D3 and D4). Additionally, two more datasets D1a and D3a, were created, by translating D1 and D3, using the API discussed in section 3.3

Dataset No.	Language	Eyewitness	Not Eyewitness	Total
DS1	Spanish (Earthquakes)	14	499	513
DS1a	English (Earthquakes)	14	499	513
DS2	English (Floods)	638	2092	2730
DS3	Spanish	117	2803	2920
DS3a	English	117	2803	2920
DS4	English	1652	17110	18762

Table 3.4: Additional datasets - D1a and D3a

5 - fold cross validation has been performed in this project, that means, each dataset is split into five equally or almost equally sized folds. Data are stratified before breaking into five folds. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole [Refaeilzadeh et al., 2009]. For example, in a dataset where 80% of the target values are "No" and 20% are "Yes" each fold would have roughly 80%

"No" responses and 20% "Yes" ones. Stratified cross-validation is frequently recommended when the target variable is imbalanced.

The performance of each learning algorithm on each fold can be tracked using some predetermined performance metric like accuracy. Upon completion, five samples of the performance metric will be available for each algorithm.

Weka's StratifiedRemoveFolds *weka.filters.supervised.instance.StratifiedRemoveFolds* class has been used to create the required folds. The filter takes as options the number of folds to be created, the selected fold, and the seed for randomizing. All these have default values [Frank, 2017].

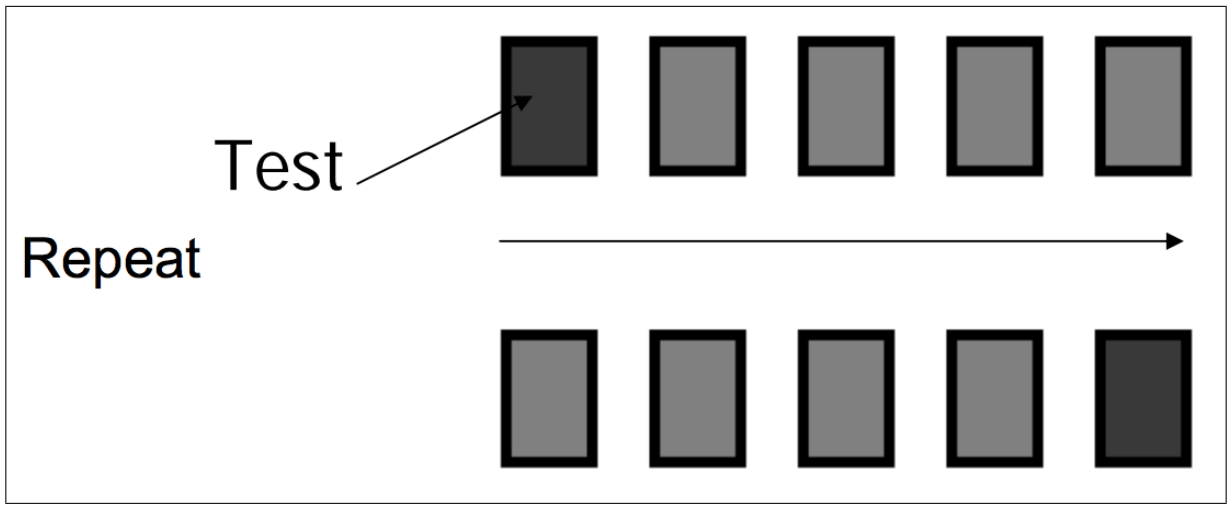

Figure 3.2: Cross-Validation

Four folds are combined to create the training set, while the remaining fold becomes the test set, as shown below.

Set	**Training Set**	**Test Set**
A	2, 3, 4, 5	1
B	1, 3, 4, 5	2
C	1, 2, 4, 5	3
D	1, 2, 3, 5	4
E	1, 2, 4, 5	5

Table 3.5: Cross-Validation datasets

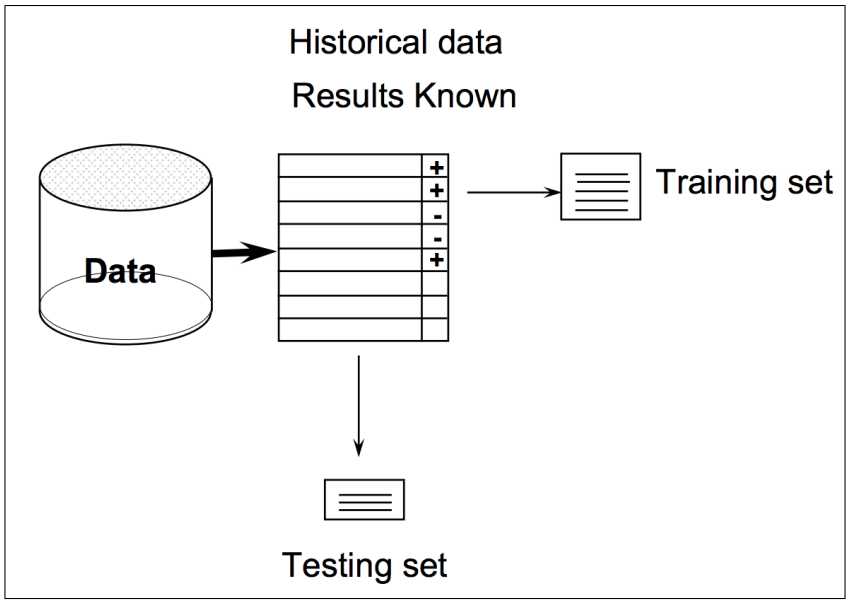

Figure 3.3: Splitting data into training & test sets

3.7 Experiments

In this section, I will discuss in detail on the goals of the research and the methodology used to achieve the goals.

3.7.1 Experiment Details

Two major experiments were conducted. The basic idea for these experiments is two folds

- analyze how training and classifying in the primary language (English / Spanish) work on the tweets in the same language (English / Spanish)
- how training and classifying in English work on the translated tweets (Spanish -> English)

First, the three smaller datasets, DS1, DS1a and DS2, corresponding to the disasters - earthquakes and floods were evaluated. The reasoning behind is to see how domain specific tweets compare against cross-domain tweets during classification.

The second experiment consisted of the larger datasets in both Spanish and English (DS3, DS3a and DS4). This dataset consists of cross - domain tweets, some of which include

Experiment No.	Training Set	Test Set
E1a	DS1	DS1
E1b	DS1a	DS1a
E1c	DS2	DS2
E1d	DS2	DS1a
E1e	DS2	(DS1a + DS2)
E1f	(DS1a + DS2)	(DS1a + DS2)

Table 3.6: Experiment 1

Wildfires, Typhoon, Shootings, Explosions, Crash, etc., apart from the Earthquakes and Floods from the first experiment.

Experiment No.	Training Set	Test Set
E2a	DS3	DS3
E2b	DS3a	DS3a
E2c	DS4	DS4
E2d	DS4	DS3a
E2e	DS4	(DS3a + DS4)
E2f	(DS3a + DS4)	(DS3a + DS4)

Table 3.7: Experiment 2

The results of both the experiments are discussed in section 4.2

3.7.2 Classifier Details

In machine learning paradigm, a classifier is a supervised function (machine learning tool) where the learned (target) attribute is categorical ("nominal"). It is used after the learning process to classify new records (data) by giving them the best target attribute (prediction) [Gerard, 2017]. In Weka, the supervised functions are derived from the abstract class - *weka.classifiers.Classifier*

To use the Weka's Naive Bayes classifier, the data should be converted from string to numeric or binary attributes. This can be completed by applying the *StringToWordVector* filter in the *weka.filters* package. StringToWordVector produces numeric attributes that represent the frequency of words in the value of a string attribute. The new attributes are determined from the full set of values in the string attribute [Frank et al., 2016]. As the StringToWord-

Vector filter places the class attribute of the output data at the beginning, Reorder filter was used to change it as the last attribute, setting the parameter of set AttributeIndices method to *2-last, first*

```
StringToWordVector stw = new StringToWordVector();
stw.setLowerCaseTokens(true);
stw.setDoNotOperateOnPerClassBasis(true);
stw.setInputFormat(train);

Reorder r = new Reorder();
r.setAttributeIndices("2-last,first");
```

Listing 3.2: Unsupervised filters code snippet

MultiFilters class was used to apply both the filters together as shown in the following code snippet.

```
MultiFilter mf = new MultiFilter();
mf.setInputFormat(train);
mf.setFilters(new Filter[]{stw, r});

Instances newTrain = Filter.useFilter(train, mf);
Instances newTest = Filter.useFilter(test, mf);
```

Listing 3.3: MultiFilter code snippet

As I have created both the training sets and test sets, I trained the classifier using the training dataset and used it to evaluate the test dataset.

```
Instances randTrainData = randomize(seed, folds, newTrain);
```

```
Evaluation eval = new Evaluation(randTrainData);

Classifier clsCopy = new NaiveBayes();
clsCopy. buildClassifier (newTrain);
eval.evaluateModel(clsCopy, newTest);
```

Listing 3.4: Classifier code snippet

Training data is randomized using the random number generator (java.util.Random) before cross-validation can be performed. The above code snippet performs 5-fold cross-validation with a Naive Bayes algorithm. The Evaluation object is initialized with *randTrainData* dataset to ensure that the structure of the training data is understood. The trained classifier is then employed in the classification of unknown / unlabeled tweets and, for each tweet, it assigns the probability of belonging to either of the class: Eyewitness or NotEyewitness.

Chapter 4

Experimental Results

In the previous chapter, I described about the various stages of the experiment until the evaluation of the datasets. In this chapter, I will discuss the about the various statistical options provided by Weka and how it was used to arrive at the results.

4.1 Statistics

In 3.7, I mentioned how Weka's Evaluation class was used to build a classifier and how to use it on the test dataset. Evaluation method also provides with various methods to generate statistics regarding the classifier to help us better understand it. The summary methods that I used to comprehend the classifier are as follows:

toSummaryString	Outputs the performance statistics in summary form
toMatrixString	Outputs the performance statistics as a classification confusion matrix.
toClassDetailsString	Generates a breakdown of the accuracy for each class, incorporating various information-retrieval statistics, such as true/false positive rate, precision/recall/F-Measure.

---

writer.println(eval.toSummaryString(``=== Fold'' + (i + 1) + ``/'' + folds +``
Cross−validation ===='', **false**));

```
writer.println(eval.toClassDetailsString("=== Detailed Accuracy by Class\n"));
writer.println(eval.toMatrixString("=== Confusion matrix for fold " + (i + 1)
    + " / " + folds + " ===\n "));
```

Listing 4.1: Summary methods code snippet

```
=== Setup ===
Classifier: Naive Bayes
Dataset: /Users/bhavs/Documents/GitHub/DataFiles/ARFF/DA_arff/CLT26_EN/Training Set
Folds: 5
Seed: 1


=== Fold 1/5 Cross-validation ===
Correctly Classified Instances        3104               82.7072 %
Incorrectly Classified Instances       649               17.2928 %
Kappa statistic                          0.2989
Mean absolute error                      0.1856
Root mean squared error                  0.3735
Relative absolute error                115.4762 %
Root relative squared error            131.6945 %
Total Number of Instances             3753

=== Detailed Accuracy by Class

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.610    0.152    0.280      0.610    0.384      0.330    0.817     0.360     yes_class
                 0.848    0.390    0.957      0.848    0.899      0.330    0.817     0.975     no_class
Weighted Avg.    0.827    0.369    0.898      0.827    0.854      0.330    0.817     0.921

=== Confusion matrix for fold 1/5 ===

    a    b   <-- classified as
  202  129 |    a = yes_class
  520 2902 |    b = no_class
```

Figure 4.1: Sample result generated by the classifier

4.2 Results and Discussion

The goal is to bridge the gap between the source and target by learning a classifier from a single language (English / Spanish) instances to predict the labels for the new target instances (English / Spanish). The results of my experiments consists of the percentage accuracy of correctly classified instances for each of the five sets in all the 12 experiments. The average accuracy across all the five folds for each experiment are provided in the Figures 4.6 and 4.7.

The classifiers that use one language to train and use test data from the same language to classify are identified as Monolingual Classifiers. From the figure 4.6, it looks like the classifier

Spanish on Spanish

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	103	99 (96.12%)	4 (3.88%)
1,3,4,5	2	103	100 (97.09%)	3 (2.91%)
1,2,4,5	3	103	100 (97.09%)	3 (2.91%)
1,2,3,4	4	102	99 (97.06%)	3 (2.94%)
1,2,3,5	5	102	98 (96.08%)	4 (3.92%)
		513	**496 (96.69%)**	**17 (3.31%)**

English on English

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	546	428 (78.39%)	118 (21.61%)
1,3,4,5	2	546	424 (77.66%)	122 (22.34%)
1,2,4,5	3	546	422 (77.29%)	124 (22.71%)
1,2,3,4	4	546	434 (79.49%)	112 (20.51%)
1,2,3,5	5	546	404 (73.99%)	142 (26.01%)
		2730	**2112 (77.36%)**	**618 (22.64%)**

Translated Spanish on Translated Spanish

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	103	97 (94.17%)	6 (5.83%)
1,3,4,5	2	103	98 (95.15%)	5 (4.85%)
1,2,4,5	3	103	100 (97.09%)	3 (2.91%)
1,2,3,4	4	102	101 (99.02%)	1 (0.98%)
1,2,3,5	5	102	96 (94.12%)	6 (5.88%)
		513	**492 (95.91%)**	**21 (4.09%)**

Figure 4.2: Monolingual Cross-Validation Accuracy Results - Experiments on the Smaller Datasets (DS1, DS1a, DS2)

using Spanish texts and Translated Spanish texts to train and test performed better than the classifier using English texts. But, it should be noted that the number of tweets in Spanish texts were relatively smaller than those of the English texts

The classifiers that use data that is a combination of English and translated Spanish to train and classify are identified as Bilingual Classifiers. From the figure 4.7, it looks like the classifier using pure English texts as training data and Translated Spanish texts as Test Data performed relatively better than the other classifiers. But, there is only slight difference between the accuracy of other classifiers. Though the number of texts in the combination dataset (English + Translated Spanish) is extremely high, the accuracy is only around 83%. This could be due to the fact that the translated Spanish texts contains untranslated Spanish words, which would affect the overall performance of the classifiers.

Spanish on Spanish

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	103	99 (96.12%)	4 (3.88%)
1,3,4,5	2	103	100 (97.09%)	3 (2.91%)
1,2,4,5	3	103	100 (97.09%)	3 (2.91%)
1,2,3,4	4	102	99 (97.06%)	3 (2.94%)
1,2,3,5	5	102	98 (96.08%)	4 (3.92%)
		513	**496 (96.69%)**	**17 (3.31%)**

English on English

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	546	428 (78.39%)	118 (21.61%)
1,3,4,5	2	546	424 (77.66%)	122 (22.34%)
1,2,4,5	3	546	422 (77.29%)	124 (22.71%)
1,2,3,4	4	546	434 (79.49%)	112 (20.51%)
1,2,3,5	5	546	404 (73.99%)	142 (26.01%)
		2730	**2112 (77.36%)**	**618 (22.64%)**

Translated Spanish on Translated Spanish

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	103	97 (94.17%)	6 (5.83%)
1,3,4,5	2	103	98 (95.15%)	5 (4.85%)
1,2,4,5	3	103	100 (97.09%)	3 (2.91%)
1,2,3,4	4	102	101 (99.02%)	1 (0.98%)
1,2,3,5	5	102	96 (94.12%)	6 (5.88%)
		513	**492 (95.91%)**	**21 (4.09%)**

Figure 4.3: Monolingual Cross-Validation Accuracy Results - Experiments on the Larger Datasets (DS3, DS3a, DS4)

CLT26 ES on CLT26 ES

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	584	541 (92.64%)	43 (7.36%)
1,3,4,5	2	584	531 (90.92%)	53 (9.08%)
1,2,4,5	3	584	537 (91.95%)	47 (8.05%)
1,2,3,4	4	584	531 (90.92%)	53 (9.08%)
1,2,3,5	5	584	532 (91.10%)	52 (8.90%)
		2920	**2672 (91.51%)**	**248 (8.49%)**

CLT26 EN on CLT26 EN

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	3753	3104 (82.71%)	649 (17.29%)
1,3,4,5	2	3752	3092 (82.41%)	660 (17.59%)
1,2,4,5	3	3752	3127 (83.34%)	625 (16.66%)
1,2,3,4	4	3752	3071 (81.85%)	681 (18.15%)
1,2,3,5	5	3753	3139 (83.64%)	614 (16.36%)
		18762	**15533 (82.79%)**	**3229 (17.21%)**

CLT26 TES on CLT26 TES

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	584	541 (92.64%)	43 (7.36%)
1,3,4,5	2	585	532 (90.94%)	53 (9.06%)
1,2,4,5	3	584	522 (89.38%)	62 (10.62%)
1,2,3,4	4	584	543 (92.98%)	41 (7.02%)
1,2,3,5	5	585	539 (92.14%)	46 (7.86%)
		2922	**2677 (91.62%)**	**245 (8.38%)**

Figure 4.4: Bilingual Cross-Validation Accuracy Results - Experiments on the Smaller Datasets (DS1a, (DS1a + DS2))

CLT26 EN on CLT26 TES

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	584	524 (89.73%)	60 (10.27%)
1,3,4,5	2	585	521 (89.06%)	64 (10.94%)
1,2,4,5	3	584	532 (91.10%)	52 (8.90%)
1,2,3,4	4	584	531 (90.92%)	53 (9.08%)
1,2,3,5	5	585	529 (90.43%)	56 (9.57%)
		2922	**2637 (90.25%)**	**285 (9.75%)**

CLT26 EN on (CLT26 EN + CLT26 TES)

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	4337	3648 (84.11%)	689 (15.89%)
1,3,4,5	2	4336	3630 (83.72%)	706 (16.28%)
1,2,4,5	3	4337	3658 (84.34%)	679 (15.66%)
1,2,3,4	4	4337	3623 (83.54%)	714 (16.46%)
1,2,3,5	5	4337	3661 (84.41%)	676 (15.59%)
		21684	**18220 (84.03%)**	**3464 (15.97%)**

(CLT26 EN + CLT26 TES) on (CLT26 EN + CLT26 TES)

Training Data	Test Data	Total No of Instances	Correctly Classified Instances	Incorrectly Classified Instances
2,3,4,5	1	4337	3623 (83.54%)	714 (16.46%)
1,3,4,5	2	4336	3661 (84.43%)	675 (15.57%)
1,2,4,5	3	4337	3630 (83.70%)	707 (16.30%)
1,2,3,4	4	4337	3590 (82.78%)	747 (17.22%)
1,2,3,5	5	4337	3606 (83.15%)	731 (16.85%)
		21684	**18110 (83.52%)**	**3574 (16.48%)**

Figure 4.5: Bilingual Cross-Validation Accuracy Results - Experiments on the Larger Datasets (DS3a, (DS3a + DS4))

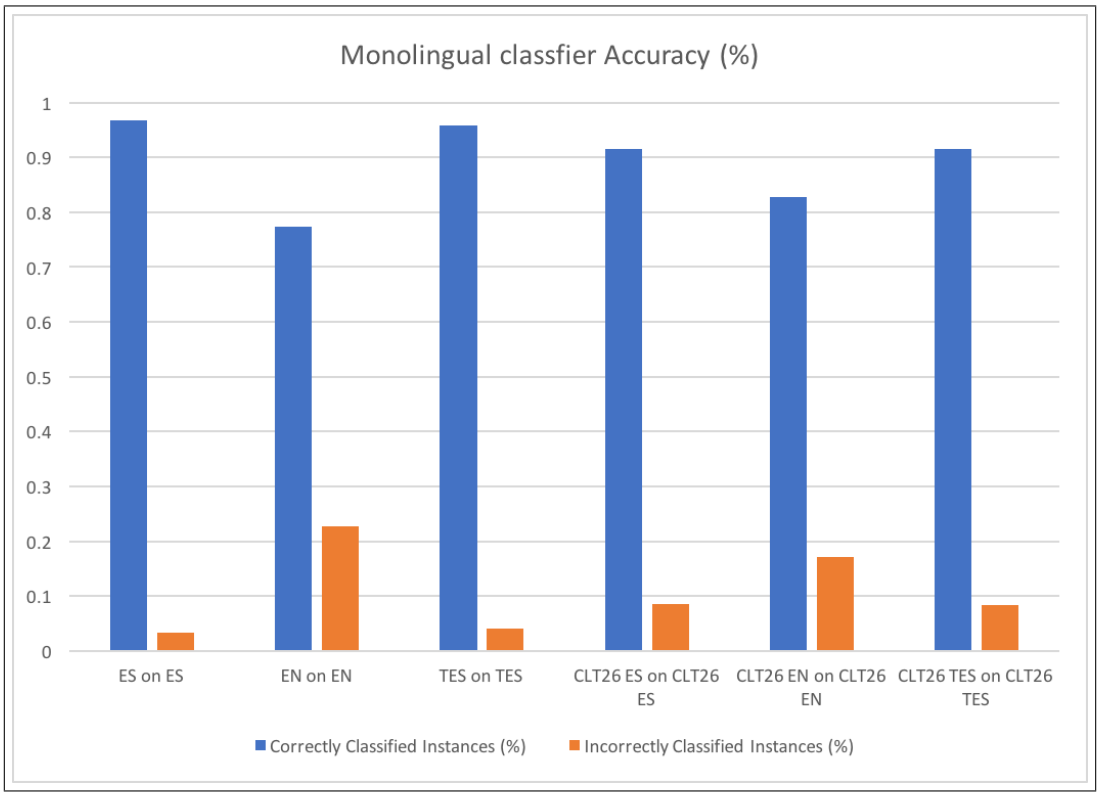

Figure 4.6: Monolingual Accuracy in Percentage

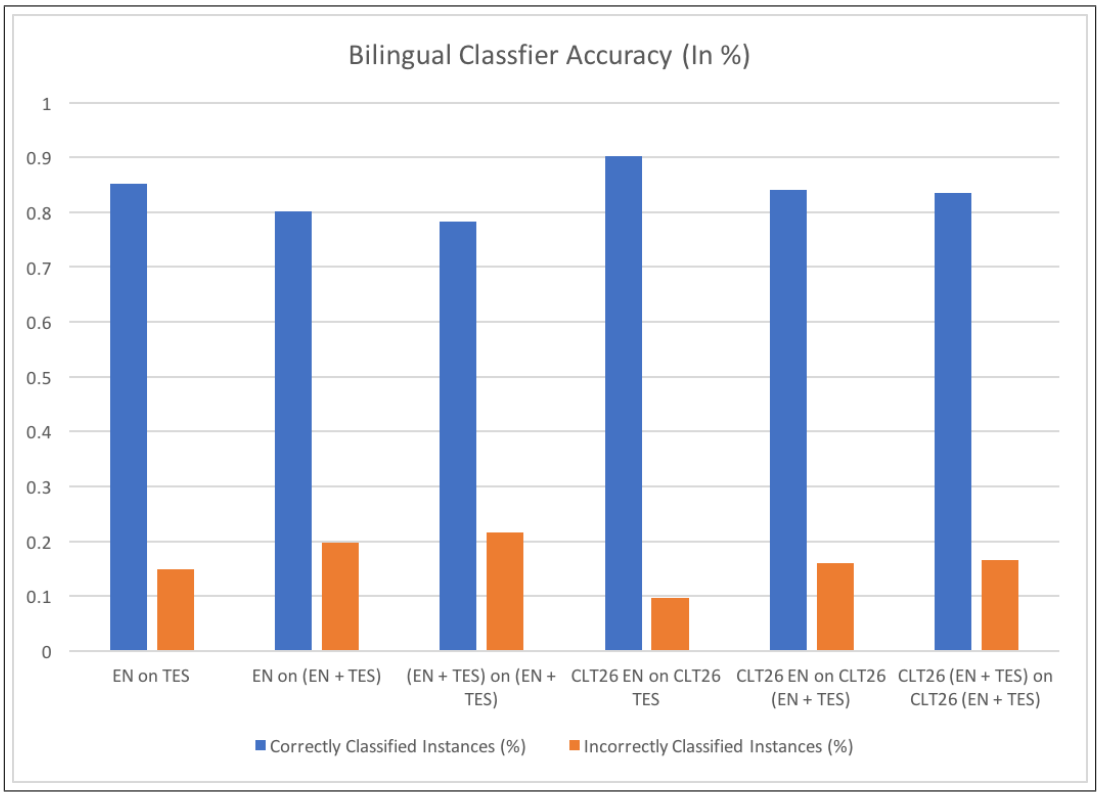

Figure 4.7: Bilingual Accuracy in Percentage

Chapter 5

Conclusions and Future Work

Social media mining for disaster response and coordination has been receiving an increasing level of attention from the research community. It is still necessary to develop automated mechanisms to find critical and actionable information on Social Media in real-time.

The classifiers used in these experiments combine effective feature extraction using machine learning approach to classify the tweets as either eyewitness reports or otherwise to improve disaster response efforts. Based on the results provided in the section 4.2, training and classifying using a different language such as Spanish and applying the classifier to learn unlabeled Spanish tweets performs similarly as the translated tweets. There is a likelihood that the accuracy might decrease as the sample size increases but having a large training set would unmistakably improve the performance.

Also, using a purely English training dataset increases the accuracy of the classifier to classify translated tweets as compared to a mix of English and translated tweets.

As a whole, I would conclude that it is possible to build a general classifier that can classify tweets from different languages with the source/training data in English. However, it requires a large dataset, and it should be continuously improved.

In future, specific variations of terms over different disasters can be analyzed to perform annotation on all disasters. Disaster terminologies can be formalized in more detail to improve accuracy. And then it would be possible to automatically annotate the informative

tweets into more specific information types that are frequently found in natural disasters.

More robust solutions can be implemented by integrating and combining event information from multiple social sources, such as Facebook, Instagram, and Snapchat. These approaches and additional features can help detect new events from complementary social media sites. We can also use the tweets along with the geotagged information to construct maps of the affected areas for real-time situational awareness during disasters. Domain adaptation techniques can be applied to improve the performance on datasets where labels are unavailable.

Bibliography

Ademola Adimbola. The importance of social media in emergency management, 2017. URL https://mauconline.net/2017/08/22/the-importance-of-social-media-in-emergency-management/.

DHS. Innovative uses of social media in emergency management, 2013. URL https://www.dhs.gov/sites/default/files/publications/Social-Media-EM_0913-508_0.pdf.

Linda Fox, 2011. URL https://www.tnooz.com/article/how-consumers-use-social-media-in-emergencies-infographic/.

Eibe Frank, 2017. URL https://weka.wikispaces.com/Generating+cross-validation+folds+%28Filter+approach%29.

Eibe Frank, Mark A. Hall, and Ian H. Witten, 2008. URL https://www.cs.waikato.ac.nz/ml/weka/arff.html.

Eibe Frank, Mark A. Hall, and Ian H. Witten. "the weka workbench. online appendix for "data mining: Practical machine learning tools and techniques"", 2016.

Nicolas Gerard, 2017. URL https://gerardnico.com/wiki/data_mining/classification.

Aditi Gupta and Ponnurangam Kumaraguru. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st workshop on privacy and security in online social media*, page 2. ACM, 2012.

Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

Shel Holtz. Instagram emerges from sandy a major social media player, 2012. URL https://www.prdaily.com/Main/Articles/Instagram_emerges_from_Sandy_a_major_social_media_13072.aspx.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Extracting information nuggets from disaster-related messages in social media. In *ISCRAM*, 2013.

Muhammad Imran, Prasenjit Mitra, and Jaideep Srivastava. Cross-language domain adaptation for classifying crisis-related short messages. *CoRR*, abs/1602.05388, 2016. URL http://arxiv.org/abs/1602.05388.

Internet Live Stats, 2017. URL http://www.internetlivestats.com/twitter-statistics/.

Priit Kallas, 2017. URL https://www.dreamgrow.com/top-15-most-popular-social-networking-sites/.

Shamanth Kumar. *Social Media Analytics for Crisis Response*. Arizona State University, 2015.

Evan LePage. Infographic: Social media disaster response, 2013. URL https://blog.hootsuite.com/social-media-disaster-response/.

Hongmin Li, Nicolais Guevara, Nic Herndon, Doina Caragea, Kishore Neppalli, Cornelia Caragea, Anna Squicciarini, and Andrea H Tapia. Twitter mining for disaster response: A domain adaptation approach. *ISCRAM*, 2015.

Jolie OĎell. How we use social media during emergencies [infographic], 2011. URL http://mashable.com/2011/02/11/social-media-in-emergencies/#0pzjsiRNnsql.

Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW

'15, pages 994–1009, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2922-4. doi: 10.1145/2675133.2675242. URL http://doi.acm.org/10.1145/2675133.2675242.

Beverly Estephany Parilla-Ferrer, Proceso L Fernandez Jr, and Jaime T Ballena IV. Automatic classification of disaster-related tweets. In *Proc. International conference on Innovative Engineering Technologies (ICIET)*, page 62, 2014.

Viktor Pekar, Jane M Binner, and Hossein Najafi. Detecting mass emergency events on social media: One classification problem or many? In *International Conference in Data Mining*, pages 31–37, 2016. ISBN 1-60132-431-6.

Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Cross-Validation*, pages 532–538. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_565. URL https://doi.org/10.1007/978-0-387-39940-9_565.

Bruno Stecanella, 2017. URL https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/.

Chris Thornton, 2017. URL http://users.sussex.ac.uk/~christ/crs/ml/lec02b.html.

Fangzhao Wu, Sixing Wu, Yongfeng Huang, Songfang Huang, and Yong Qin. Sentiment domain adaptation with multi-level contextual sentiment knowledge. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 949–958, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4073-1. doi: 10.1145/2983323.2983851. URL http://doi.acm.org/10.1145/2983323.2983851.