Towards Natural Disasters Detection from Twitter using Topic Modelling

Mohamed Hagras
Computer Science Department
The British University in Egypt
Cairo, Egypt
Email: mohamed.hagrass@outlook.com

Ghada Hassan
Computer Science Department
The British University in Egypt and
Ain Shams University
Cairo, Egypt
Email: ghada.hassan@bue.edu.eg

Nadine Farag
Computer Science Department
The British University in Egypt
Cairo, Egypt
Email: nadine.farag@bue.edu.eg

***Abstract*—Micro-blogging services bring about large streams of posts of social media users as they share their updates and opinions. Twitter is popular for its micro-blogging service, as it has millions of active users interacting through it. Using the Twitter stream for natural disasters detection is a quick and useful detection method for SOS, TV News and disaster relief organization. However, a tweets text is unstructured, sparse, has no exact defined schema and may contain a lot of noise. In addition, after the latest Twitter update, historical data longer than 14 days is no longer accessible which limits the effectiveness of supervised learning techniques. This paper discusses results of using the Latent Dircherilet Allocation LDA topic modelling technique to detect tweets related to the 2011 Japan Tsunami. A training set of 6700 tweets and a test set of 196 tweets were filtered from a total set of collected 700,000 tweets. Accuracy results achieved was 76% using 10-fold cross validation. To evaluate how well the model fits the documents, the perplexity statistical function is compared for different values of K. Starting at $K > 1$ and $K < 10$ resulted in having $K = 2$ with the lowest perplexity of 7.5125 for training data and 60.594 for testing data. The gap between the training and testing data means that as the training data increases, the perplexity decreases and the accuracy of detection will improve. Eliminating model over fitting and creating a more generative model will positively influence the accuracy level of the prediction.**

***Index Terms*—LDA; micro blogging; Twitter; topic modeling; machine learning; latent; Dircherilet; Allocation.**

I. Introduction

As soon as a natural disaster occurs, people experiencing the hazard turn to social media to share it with the world. Immediately, posts reporting the disaster spread around the globe, sometimes even before the hazard is recorded by the governments or reported by the media, as it usually takes time for traditional media and official organizations to locate and validate the accuracy of news. Detecting natural disaster upon occurrence using micro blogging has become an area of research as micro blogging services use is growing and gaining popularity globally. Search engines designed interfaces for related queries to obtain such information, but it was not fast enough [5].

Twitter enables users to share their news and ideas in 140 words. Having 500 million users registered and over 100 million users actively posting in the community, Twitter has become a valuable source of information for data mining. In addition to the tweet text, it is possible to obtain user name, length, the number of people who shared this tweet. It is also possible to obtain some Geo-graphical information that represent the location of the tweet [11]. Due to these characteristics, Twitter has drawn the attention to its potential use in situational awareness during different hazards such as natural disasters [8]. Despite Twitter's promise to collect and mine huge amounts of real-time data, the huge size of data has its draw backs. First, researchers find it hard to obtain usable knowledge due to noise, irrelevant and sparse content. Second, after the latest API release, Twitter enforced limitations on data collection by limiting requests for accessing the stream. This has limited the researchers abilities to retrieve users' data through the text of the tweet. Additionally, due to the limitation of the search API, historical data older than two weeks cannot be retrieved, resulting in a usage limitation of supervised machine learning algorithms. Lastly, the content of the tweets is unstructured, hence, the tweets need to be filtered for relevance. When identified as relevant, related useful information such as locations and damages can to be extracted. The process should be running on real time news feed and avoiding the limitations of Twitters API by neglecting users profiles, and tweets location, then later extract locations from the text itself.

In this paper, a novel approach in natural language processing techniques was conducted to enhance event detection with the aim of improving response in disasters. The Latent Dirichlet Allocation algorithm LDA was employed for text mining and clustering to gain useful knowledge from the Twitter stream on a disaster, like time of occurrence, location of the disaster, number of injuries and lost lives. The information can then be put to use by interested parties.

The paper is organized as follows. Section II presents previous similar research. Section III presents our dataset, the solution methodology, implementation details of the LDA algorithm using spark machine learning framework and the evaluation technique. Section IV presents the results and the evaluation of the experiments. Lastly, Section V summarizes the results and discusses possible future work.

978-1-5386-2085-4/17 $31.00 © 2017 IEEE
DOI 10.1109/EECS.2017.57

II. Related Work

Text mining, also known as text analytics; is the process of gaining knowledge and extracting useful information from textual content. According to Prato [20], 80% of the textual content is qualitative and unstructured. Aggarwal described text analytics to be the process that combines information retrieval, natural language processing NLP, data mining and information extraction techniques to form a single structure for text mining [1] [2]. Text mining does not only involve historic text in large databases, but also makes use of text that is generated on-line from social media or various other lengthy data streams [2]. Topic modelling is the area of research that uses text clustering to organize topic related documents into groups. Topic modelling is a generalized probabilistic model where each topic is represented as probabilities of existence of words in document corpus. The documents are represented as probability distribution of each topic generated. There are many algorithms for topic modelling, the selection of the suitable algorithm depends on the data used and the level of pre-processing applied to data [2]. Topic modelling aims to discover the latent semantics in the corpus, that is accurate enough to identify topics better than clustering term features. In this section we aim to review research that mines Twitter stream with the aim of detection of a natural disaster.

The work in [5] developed *Tweedr* which is a Twitter mining tool used for extracting information from Twitter using classification, clustering and information retrieval techniques. Their dataset was obtained using the *Gnip* services for keyword historical data retrieval. In the classification step, the goal was to identify tweets that reported damages. They evaluated various techniques like SVM and Nave Bayes, and both achieved an accuracy of 80%. The LDA algorithm resulted in a 70% accuracy. Decisions trees resulted in 85% accuracy. Importantly, they used the geo-location as a feature in the classification and clustering. In clustering, filters were used to merge relevant tweets together. In the extraction phase, words and phrases are extracted, as well as information representing statistics, damages and causalities. This process was conducted using Conditional Random Fields (CRF).

Authors in [22] developed an earthquake detection system by implementing what they called social sensors using Twitter feed for extracting information. They used semantic analysis on tweets with respect to some features like: the number of keywords in the tweet, the position of the query word within the text, the keyword itself within the text, the next and previous words and the context of the text. They used the *Kalman* filter to estimate the location of the event.

In [9], the authors conducted a study to gain structured information from natural events using Wikipedia and Twitter. Their data was obtained using the Twitter search API, with average datasets consisting of three thousand records for five recent events. Using their developed algorithm: *Stanford Dependencies Algorithm* they were able to obtain the attribute value pairs for both numeric and textual attributes. This enabled them to find the grammatical relationships within text. For numeric attributes, they refined the basic Nave Bayesian algorithm for neighbouring words. For example, considering a tweet that has numeric attributes within it, but cannot be associated to the next or the previous word, and rather it is to be associated with the entire phrase. Furthermore, their hypothesis is based on conjoining the detection of numerical elements and textual grammatical relationships and hence associating them altogether. This might not always be convenient for real time Twitter feeds during natural hazards.

Jain [10] used Twitter to mine and analyse tweets to obtain information and geo-location in order to detect events. The process started by acquiring and selecting keywords. The data is then preprocessed to remove unnecessary tweets and eliminate noise. The third and last step is to extract both the tweet's time and location. A tweet characteristics can be requested but with some limitations due to security restrictions set by the users. The authors used the TagMe API to obtain such information.

TweetTracker [14] is an analysis tool created for Humanitarian Aid and Disaster Relief (HADR) organizations to gain useful situational insights. The authors collected tweets using Twitters steaming API. The tweets have certain pre-determined hashtags, key words and filters. They used a back end database system for data storage and a stream reader to read streamed data. The TweetTracker also employs an analysis and visualization mechanism, where Twitter feed is monitored using keyword filtering. The system used geographical related tagged tweet collections and grouped them to be visualized on a map using blue points with their relations to represent it. This was achieved using hard coded analysis on tweets. Foreign languages, other than English were translated using the Google Translate API. It is worth noting that the tool was not designed specifically for disasters, as it was indeed very general.

A. Analysis of the Related Work

As noted in the previous research, human language is unstructured because people have different writing styles and also tend to reflect on situations differently. Also, there is no fixed pattern in human written text apart from the grammatical rules. However, grammatical rules alone are not enough for an accurate prediction. Additionally, in the latest release of the Twitter API, it disabled the extraction of tweets older 14 days. This limitation of requests made it impossible to retrieve enough information about users. As a result, the simple supervised learning techniques as trees and Nave Bayes or SVMs are not optimal solutions for research. In addition, supervised learning makes processing more difficult and consumes time. On reflecting the previous work, this research uses the unsupervised LDA topic modelling algorithm to determine whether tweet topics are related to a disaster or not.

III. Solution Methodology

A. Dataset Collection

The dataset contains tweets pulled out from the Twitter stream during the tsunami crisis in Japan in 2011. The tweets

```
56194218414182401|Tsunami|Its a positive TSUNAMI against corruption and its name is Anna Hazare..|2011-04-07 20:18:00|142535822|||76398
56194241071820800|Tsunami|And Yu Wonder Why I Act Like That!!|2011-04-07 20:18:00|238694998|||763988
56194531250536448|Tsunami|I hate it. its a drought its a drought then tsunami! #beconsistent|2011-04-07 20:19:00|255777173|||763993
```

Fig. 1: Irrelevant Tweets Sample

```
45642720089346048,Tsunami,Four earthquakes both stronger than 6.0 strike offshore Japan: Tsunami warning after 7.2 quake near Japan The ... http://bit.ly/dO2Vxb,2011-
45656324515708928,Tsunami,Japan hit by quake tsunami threat,2011-03-09 17:24:00,237573963,,,219
45666000804388865,Tsunami,Strong quake hits northeast Japan tsunami warning issued http://rojname.com/?q=329461 #Peyamner Rojname,2011-03-09 18:03:00,243133079,,,284
```

Fig. 2: Filtered Tweets Sample

are selected based on their content where each tweet in the dataset includes either the word "tsunami" or the hashtag "#tsunami". The data fields collected for each tweet are specified in Table I.

TABLE I: Dataset Features

tweetID	The tweet ID is unique, and is used to identify the tweet.
Keyword	The keyword used to filter the stream.
Text	The tweet text with 140 characters or less.
Date	The date at which the tweet was posted.
user-id	The id associated with users on the system. This id is hashed to form "imaginary ids" in order to hide the real identities of users for security purposes.
user-name	The Twitter user name associated with users.
user-screen-name	The user name that appears on the screen.
temp-id	This ID represents a temporary Id for the stream request.

The data was collected using the Twitter search API which allows for the retrieval of historical data with the restriction of a maximum of 180 requests every 15 minutes [24]. However, Twitter has changed the API terms of services at the time of this paper writing.

In order to collect the tweets, some of the tweets were supplied by other researchers working on the Tsunami topic, and we collected the others by querying the Twitter search API. The gathered tweets were examined and we noticed that some of the tweets were generic and irrelevant. An illustration of such tweets is shown in Figure 1.

In order to create our training and test sets, tweets were filtered for relevancy based on the following criteria:

- Is the tweet related to tsunami or any natural disaster?
- Does the tweet report any events about tsunami?
- Does the tweet report any loss in lives or damages?

Only tweets satisfying at least one of the above criteria were chosen for further processing. Figure 2 shows a sample of the filtered tweets. The filtered dataset is then divided into separate training and test sets and stored in CSV files.

B. Dataset Preprocessing

The preprocessing was applied on the filtered dataset, where the text was converted to tokens by splitting the text at the occurrences of white spaces. Stop words were removed, such as: the, my, and his. In addition, all the text was converted to lower case. Finally, each word/term in the corpus was converted to a numeric value that represents the term frequency.

In order to make the dataset readable by Spark machine learning framework, the data frame schema that represents the dataset was created. First, the CSV file was parsed to a data frame into the spark SQL context. We used the setting that allows the data frame to automatically build its own schema based on the headers in the CSV file of the training set. The schema is presented in Figure 3.

1) Bag of Words: The Bag of Words is a model used to represent a document text as set of words. In natural language processing, the bag of words is used to describe text by a multi-set of its words. For example, two tweet texts such as Fresh quake triggers tsunami and Japan quake triggers tsunami, are converted to a general set of words where the set is {Fresh, quake, triggers, tsunami, Japan }. Then for a corpus of more than a text, the multi-set is represented by the frequency of presence of words in the text. For example, the multi-set for the first tweet is represented as [1,1,1,1] where each number the frequency of each word from the bag. The words need to be represented as tokens before creating the bag.

The second step in preprocessing is to create tokens for each word using the spark frame work built in tokenization. Using white spaces as split character, we applied the regex expression in order to clean tweets from special character such as: /,;, —, by using the set [a-zA-Z0-9] which accepts words and numbers only. The words were then converted to lower case using tweet text as input and resulting in a bag of words into a new column created in the schema named as word. An extract is shown in Figure 4.

2) Regex and Count Tokenization: In order to have a corpus ready for creating bags or using it for topic modeling, text must be converted into tokens. This is achieved through *tokenization*, which splits text into words and may apply some regex expressions for data cleaning. These regex expressions are used to filter the text according to a designed filter. Tokenizers do not differentiate between different data types or text content as they create tokens regardless of the type of content. Tokenizers split words using various methods like splitting on white space, splitting on a special character or a statistical model according to the problem specifications [16].

In order to pass the parameters to topic modelling algorithms, the documents need to be converted to a normalized numeric feature set. The count vectors are used to convert strings corpus, which is then converted to a bag of words. Tokens are converted to tokens count form where there is no prior dictionary present already. On applying

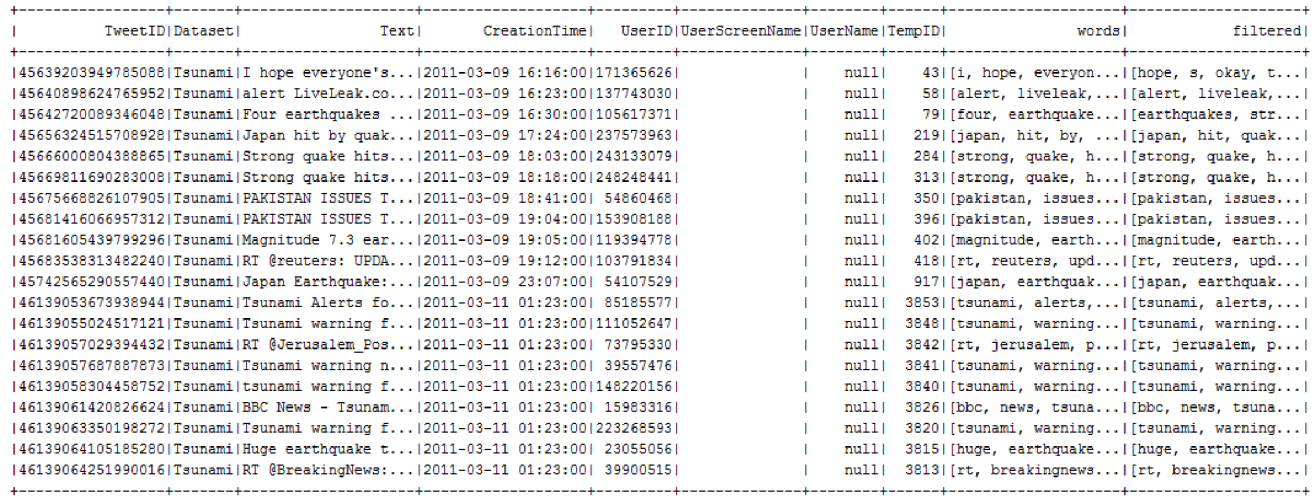
TweetID	Dataset	Text	CreationTime	UserID	UserScreenName	UserName	TempID
456392039497850 88	Tsunami	I hope everyone's...	2011-03-09 16:16:00	171365626		null	43
456408986247659 52	Tsunami	alert LiveLeak.co...	2011-03-09 16:23:00	137743030		null	58
456427200893460 48	Tsunami	Four earthquakes ...	2011-03-09 16:30:00	105617371		null	79
456563245157089 28	Tsunami	Japan hit by quak...	2011-03-09 17:24:00	237573963		null	219
456660008043888 65	Tsunami	Strong quake hits...	2011-03-09 18:03:00	243133079		null	284
456698116902830 08	Tsunami	Strong quake hits...	2011-03-09 18:18:00	248248441		null	313
456756688261079 05	Tsunami	PAKISTAN ISSUES T...	2011-03-09 18:41:00	54860468		null	350
456814160669573 12	Tsunami	PAKISTAN ISSUES T...	2011-03-09 19:04:00	153908188		null	396
456816054397992 96	Tsunami	Magnitude 7.3 ear...	2011-03-09 19:05:00	119394778		null	402
456835388134822 40	Tsunami	RT @reuters: UPDA...	2011-03-09 19:12:00	103791834		null	418
457425652905574 40	Tsunami	Japan Earthquake:...	2011-03-09 23:07:00	54107529		null	917
461390536739389 44	Tsunami	Tsunami Alerts fo...	2011-03-11 01:23:00	85185577		null	3853
461390550245171 21	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	111052647		null	3848
461390570293944 32	Tsunami	RT @Jerusalem_Pos...	2011-03-11 01:23:00	73795330		null	3842
461390576878878 73	Tsunami	Tsunami warning n...	2011-03-11 01:23:00	39557476		null	3841
461390583044587 52	Tsunami	tsunami warning f...	2011-03-11 01:23:00	148220156		null	3840
461390614208266 24	Tsunami	BBC News - Tsunam...	2011-03-11 01:23:00	15983316		null	3826
461390633501982 72	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	223268593		null	3820
461390641051852 80	Tsunami	Huge earthquake t...	2011-03-11 01:23:00	23055056		null	3815
461390642519900 16	Tsunami	RT @BreakingNews:...	2011-03-11 01:23:00	39900515		null	3813

only showing top 20 rows

Fig. 3: Parsed Data Schema

TweetID	Dataset	Text	CreationTime	UserID	UserScreenName	UserName	TempID	words
456392039497850 88	Tsunami	I hope everyone's...	2011-03-09 16:16:00	171365626		null	43	[i, hope, everyon...
456408986247659 52	Tsunami	alert LiveLeak.co...	2011-03-09 16:23:00	137743030		null	58	[alert, liveleak,...
456427200893460 48	Tsunami	Four earthquakes ...	2011-03-09 16:30:00	105617371		null	79	[four, earthquake...
456563245157089 28	Tsunami	Japan hit by quak...	2011-03-09 17:24:00	237573963		null	219	[japan, hit, by, ...
456660008043888 65	Tsunami	Strong quake hits...	2011-03-09 18:03:00	243133079		null	284	[strong, quake, h...
456698116902830 08	Tsunami	Strong quake hits...	2011-03-09 18:18:00	248248441		null	313	[strong, quake, h...
456756688261079 05	Tsunami	PAKISTAN ISSUES T...	2011-03-09 18:41:00	54860468		null	350	[pakistan, issues...
456814160669573 12	Tsunami	PAKISTAN ISSUES T...	2011-03-09 19:04:00	153908188		null	396	[pakistan, issues...
456816054397992 96	Tsunami	Magnitude 7.3 ear...	2011-03-09 19:05:00	119394778		null	402	[magnitude, earth...
456835388134822 40	Tsunami	RT @reuters: UPDA...	2011-03-09 19:12:00	103791834		null	418	[rt, reuters, upd...
457425652905574 40	Tsunami	Japan Earthquake:...	2011-03-09 23:07:00	54107529		null	917	[japan, earthquak...
461390536739389 44	Tsunami	Tsunami Alerts fo...	2011-03-11 01:23:00	85185577		null	3853	[tsunami, alerts,...
461390550245171 21	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	111052647		null	3848	[tsunami, warning...
461390570293944 32	Tsunami	RT @Jerusalem_Pos...	2011-03-11 01:23:00	73795330		null	3842	[rt, jerusalem, p...
461390576878878 73	Tsunami	Tsunami warning n...	2011-03-11 01:23:00	39557476		null	3841	[tsunami, warning...
461390583044587 52	Tsunami	tsunami warning f...	2011-03-11 01:23:00	148220156		null	3840	[tsunami, warning...
461390614208266 24	Tsunami	BBC News - Tsunam...	2011-03-11 01:23:00	15983316		null	3826	[bbc, news, tsuna...
461390633501982 72	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	223268593		null	3820	[tsunami, warning...
461390641051852 80	Tsunami	Huge earthquake t...	2011-03-11 01:23:00	23055056		null	3815	[huge, earthquake...
461390642519900 16	Tsunami	RT @BreakingNews:...	2011-03-11 01:23:00	39900515		null	3813	[rt, breakingnews...

Fig. 4: Tokenized Data and Bag of Words



TweetID	Dataset	Text	CreationTime	UserID	UserScreenName	UserName	TempID	words	filtered
456392039497850 88	Tsunami	I hope everyone's...	2011-03-09 16:16:00	171365626		null	43	[i, hope, everyon...	[hope, s, okay, t...
456408986247659 52	Tsunami	alert LiveLeak.co...	2011-03-09 16:23:00	137743030		null	58	[alert, liveleak,...	[alert, liveleak,...
456427200893460 48	Tsunami	Four earthquakes ...	2011-03-09 16:30:00	105617371		null	79	[four, earthquake...	[earthquakes, str...
456563245157089 28	Tsunami	Japan hit by quak...	2011-03-09 17:24:00	237573963		null	219	[japan, hit, by, ...	[japan, hit, quak...
456660008043888 65	Tsunami	Strong quake hits...	2011-03-09 18:03:00	243133079		null	284	[strong, quake, h...	[strong, quake, h...
456698116902830 08	Tsunami	Strong quake hits...	2011-03-09 18:18:00	248248441		null	313	[strong, quake, h...	[strong, quake, h...
456756688261079 05	Tsunami	PAKISTAN ISSUES T...	2011-03-09 18:41:00	54860468		null	350	[pakistan, issues...	[pakistan, issues...
456814160669573 12	Tsunami	PAKISTAN ISSUES T...	2011-03-09 19:04:00	153908188		null	396	[pakistan, issues...	[pakistan, issues...
456816054397992 96	Tsunami	Magnitude 7.3 ear...	2011-03-09 19:05:00	119394778		null	402	[magnitude, earth...	[magnitude, earth...
456835388134822 40	Tsunami	RT @reuters: UPDA...	2011-03-09 19:12:00	103791834		null	418	[rt, reuters, upd...	[rt, reuters, upd...
457425652905574 40	Tsunami	Japan Earthquake:...	2011-03-09 23:07:00	54107529		null	917	[japan, earthquak...	[japan, earthquak...
461390536739389 44	Tsunami	Tsunami Alerts fo...	2011-03-11 01:23:00	85185577		null	3853	[tsunami, alerts,...	[tsunami, alerts,...
461390550245171 21	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	111052647		null	3848	[tsunami, warning...	[tsunami, warning...
461390570293944 32	Tsunami	RT @Jerusalem_Pos...	2011-03-11 01:23:00	73795330		null	3842	[rt, jerusalem, p...	[rt, jerusalem, p...
461390576878878 73	Tsunami	Tsunami warning n...	2011-03-11 01:23:00	39557476		null	3841	[tsunami, warning...	[tsunami, warning...
461390583044587 52	Tsunami	tsunami warning f...	2011-03-11 01:23:00	148220156		null	3840	[tsunami, warning...	[tsunami, warning...
461390614208266 24	Tsunami	BBC News - Tsunam...	2011-03-11 01:23:00	15983316		null	3826	[bbc, news, tsuna...	[bbc, news, tsuna...
461390633501982 72	Tsunami	Tsunami warning f...	2011-03-11 01:23:00	223268593		null	3820	[tsunami, warning...	[tsunami, warning...
461390641051852 80	Tsunami	Huge earthquake t...	2011-03-11 01:23:00	23055056		null	3815	[huge, earthquake...	[huge, earthquake...
461390642519900 16	Tsunami	RT @BreakingNews:...	2011-03-11 01:23:00	39900515		null	3813	[rt, breakingnews...	[rt, breakingnews...

Fig. 5: The Cleaned Corpus

the model, it produces a sparse representation of date in double formats [17]. When the model fits the data, it takes the biggest vocabulary size of words and the user may need to set the minDF which sets the lowest documents counts as the term must exist in in order to be used as a feature. Assuming we have a tweet id 1 with Fresh quake triggers tsunami as text, and tweet id 2 with Japan quake triggers tsunami as text, hence each tweet is considered as an independent document. In order to later pass the data to LDA or other topic modelling techniques, the text should be count vectorized. The documents are saved in a data frame represented as:

TABLE II: Tweets Regex and Count Tokenization Example

TweetID	Text	Vector
1	Array(”fresh”,”quake”, ”triggers”,”tsunami”)	(5,[0,1,2,3,4], [1,1,1,1,0])
2	Array(”japan”,quake, ”triggers”,”tsunami”)	(5,[0,1,2,3,4], [0,1,1,1,1,0])

At each row, the count vecotrization is invoked resulting in a spare vector as an output. The application first calculates the maximum number of terms (4 in this example), then it counts the number of tokens in the array and their frequency resulting for the following vector:

The third step is to clean the corpus to remove the stop words as these words affect the words distributions and the accuracy of the learning model. The spark built-in remover algorithm was used to remove the stop words. The algorithm takes the new words column as input and results in a new schema with a filtered column as an output as shown in Figure 5.

In the last step of data preparation, the corpus is converted to a double sparse vector of words representations and frequencies. Using the count vector model of spark, it takes the filtered column as input and results in a new column of features named as ”feature” which is the input for LDA. The feature corpus is shown in Figure 6.

C. LDA Algorithm

LDA stands for Latent Dirichlet Allocation, a topic modeling technique used to identify the topics that a document contains. It can be used for text clustering and classification. In this paper, LDA was used as a text clustering algorithm to obtain the topic distributions for tweets using a top down approach. The top-down approach traces documents from the beginning till the end. LDA describes the documents D as a set of topics T [15]. The algorithm iterates and randomly assigns words W to K topics T where K is the number of topics. In order to obtain the right K, several trial and error iterations are performed until the best combination of high log likelihood and a low perplexity are achieved. This random assignment results in the generalization of the topics as it represents all topics and distribution of words in topics [7].

In the second step, the algorithm goes through iterative steps, where in each topic T, the percentage of words is computed W in documents D : $P(T|D)$. The percentage of assignments to topics T for all documents D which is represented by $words$: $P(W|T)$. This step is performed iteratively until the algorithm provides a strong relation of assignments. Subsequently, the distribution of words over topics and other assignments calculated are used to calculate the distribution of topic mixture over input documents.

The LDA is trained on the latest corpus with $K = 10$ and decrementing K by one each time until the perfect K that gives the highest log likelihood value is found. During the testing phase, the detection logic has a threshold value to detect new documents. This step is illustrated in details in the next section.

Text	CreationTime	UserID	UserScreenName	UserName	TempID	words	filtered	feature
hope everyone's...	2011-03-09 16:16:00	171365626		null	43	[i, hope, everyon...	[hope, s, okay, t...	(183,[0,1,3,4,6,9...
ert LiveLeak.co...	2011-03-09 16:23:00	137743030		null	58	[alert, liveleak,...	[alert, liveleak,...	(183,[0,1,2,7,11,...
ır earthquakes ...	2011-03-09 16:30:00	105617371		null	79	[four, earthquake...	[earthquakes, str...	(183,[0,1,3,4,92,...
pan hit by quak...	2011-03-09 17:24:00	237573963		null	219	[japan, hit, by, ...	[japan, hit, quak...	(183,[0,1,3,7,165...
rong quake hits...	2011-03-09 18:03:00	243133079		null	284	[strong, quake, h...	[strong, quake, h...	(183,[0,1,3,4,11,...
rong quake hits...	2011-03-09 18:18:00	248248441		null	313	[strong, quake, h...	[strong, quake, h...	(183,[0,1,3,4,10,...
KISTAN ISSUES T...	2011-03-09 18:41:00	54860468		null	350	[pakistan, issues...	[pakistan, issues...	(183,[0,1,2,4,9,2...
KISTAN ISSUES T...	2011-03-09 19:04:00	153908188		null	396	[pakistan, issues...	[pakistan, issues...	(183,[0,1,2,4,9,1...
gnitude 7.3 ear...	2011-03-09 19:05:00	119394778		null	402	[magnitude, earth...	[magnitude, earth...	(183,[0,1,2,17,56...
@reuters: UPDA...	2011-03-09 19:12:00	103791834		null	418	[rt, reuters, upd...	[rt, reuters, upd...	(183,[0,1,3,4,8,1...
pan Earthquake:...	2011-03-09 23:07:00	54107529		null	917	[japan, earthquak...	[japan, earthquak...	(183,[0,1,2,6,9,1...
nami Alerts fo...	2011-03-11 01:23:00	85185577		null	3853	[tsunami, alerts,...	[tsunami, alerts,...	(183,[0,5,18,19,3...
nami warning f...	2011-03-11 01:23:00	111052647		null	3848	[tsunami, warning...	[tsunami, warning...	(183,[0,4,27,45,4...
@Jerusalem_Pos...	2011-03-11 01:23:00	73795330		null	3842	[rt, jerusalem, p...	[rt, jerusalem, p...	(183,[0,1,2,3,4,7...
nami warning n...	2011-03-11 01:23:00	39557476		null	3841	[tsunami, warning...	[tsunami, warning...	(183,[0,4,5,6,15,...
nami warning f...	2011-03-11 01:23:00	148220156		null	3840	[tsunami, warning...	[tsunami, warning...	(183,[0,1,4,5,34,...
: News - Tsunam...	2011-03-11 01:23:00	15983316		null	3826	[bbc, news, tsuna...	[bbc, news, tsuna...	(183,[0,1,3,13,16...
nami warning f...	2011-03-11 01:23:00	223268593		null	3820	[tsunami, warning...	[tsunami, warning...	(183,[0,4,6,7,74,...
ge earthquake t...	2011-03-11 01:23:00	23055056		null	3815	[huge, earthquake...	[huge, earthquake...	(183,[0,1,2,5,6,7...
@BreakingNews:...	2011-03-11 01:23:00	39900515		null	3813	[rt, breakingnews...	[rt, breakingnews...	(183,[0,4,5,8,40,...

Fig. 6: The Feature Corpus

After the pre-processing, the corpus is ready to run the learning model on the feature corpus. First, the corpus is passed into a map function which is a built-in distributed function to map the corpus in a form of a unique ID and the feature vector. In this case, the unique id used is the TweetID and the feature column is the feature vector. Several runs are used with various K values to determine the best K, and likelihood and perplexity functions are calculated for each run. The highest likelihood and lowest probability was achieved at $K = 2$ as the number of topics that fits the data. This K represents the tsunami crisis topic which means that any new document that fits into any one or two topics, is related to a tsunami report. The model is then saved using the built-in model saving method resulting in a set of inferred topics as each topic is represented as the probability distribution over terms. This is calculated in a topic matrix of $[TxW]$.

IV. Results

A. Validation results

To test the accuracy of LDA, a new dataset contains 196 manually-chosen tweets that are reporting tsunami incidents. These tweets are first pre-processed, then the topic distribution for the testing data is calculated using the topicDistributions method implemented in Spark, which gives results of the probability that the document fits topic 0 and topic 1. According to [21] and [23], when LDA fails in setting a nearly equal distribution with a difference value between .01 and .05 or having one topic of probability 1 and another of 0, and because the training data was all about tsunami, thus both topics represent a tsunami and an algorithm is designed in order to get the count of detected documents.

The evaluation technique gets a temporary array of topics distributed over documents, and iterates over each array element, comparing the difference between topic 0 and topic 1 distributions, then checks whether these topics difference is smaller than the threshold value used to eliminate data. This threshold value is set to be 0.1, which means there is at least an 0.55 probability of one topic and 0.45 of the other. With this value, we ensure that the documents that do not represent disasters are eliminated [9] [25]. In order to get the right threshold value between different topics, the similarity function is used to calculate the similarity between the nearest neighbours and obtain the threshold value. However in this paper, both topics represent disasters with two different distributions of words, thus a threshold value chosen based on the elimination of mistakes is misidentifying. Then, each time a tweet is eliminated, the counter is decremented by 1, and each time a tweet is detected, the counter is incremented by 1. After each inner loop iteration, if the counter is 1, this means a tweet is detected, thus incrementing the detected tweets counter by 1, then the iteration continues over the loops for the next document when the counter is reset to 0. Following that, the number of detected documents is printed, in order to get the average accuracy over 10 iterations, the number of detected documents is calculated for 10 runs and divided in each run by the number of total tweets resulting in the accuracy of detection, then the accuracies are summed and divided by n, where n is the count of documents.

B. Evaluation of Results and Discussion

In order to evaluate the LDA, how good the model fits the training data, and whether it is a generative or normalized fit, where a generative model means the model can detect unstructured data that does not have a certain pattern, the perplexity statistical model is used to evaluate topic modelling algorithms using the formula below [6]:

$$preplexity(D_{test}) = exp\left\{-\frac{\sum_{d=1}^{M} logp(w_d)}{\sum_{d=1}^{M} N_d}\right\}$$

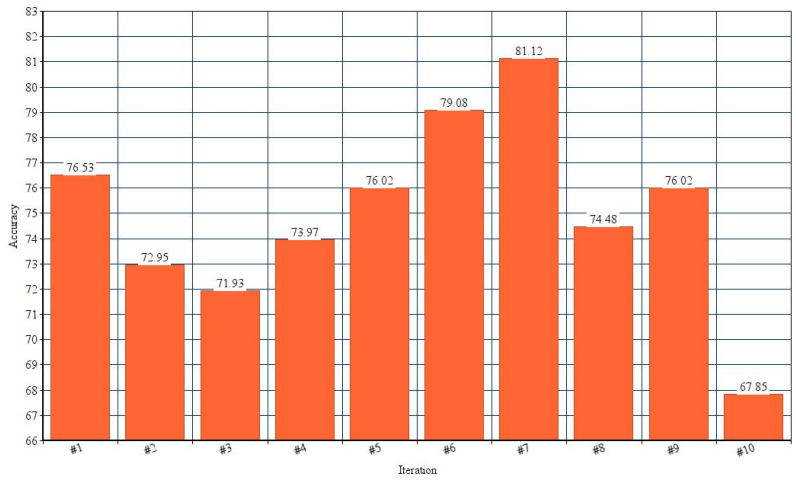


Fig. 7: Detection Accuracy using 10-fold cross validations

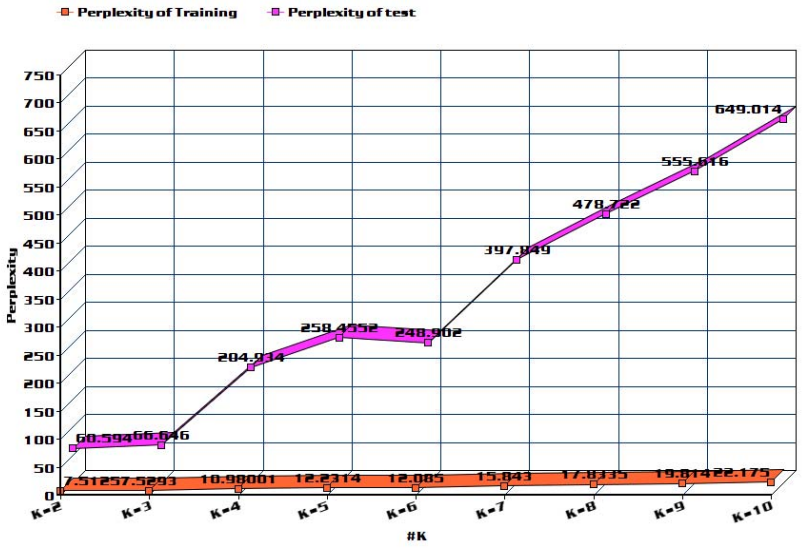


Fig. 8: Perplexity Results

Where M is the number of documents in the corpus, N_d is the number of words in the topic, W_d is the words in the documents D. The perplexity of the training data and the test data is computed for each K, where K is the number of topics, ranging from$K = 2$ to $K = 10$.

In the first test, the results have different accuracies each time, thus the average accuracy is calculated, resulting in a 76% accuracy of successful detection of natural disasters. On the other hand, the real average accuracy of the algorithm is 90% using a smaller threshold value. However, a user might write a tweet about a tsunami and the tweet might not be reporting a real-time disaster, which may result in different topic distributions that are not equal. At the same time, the difference between the probabilities is a small fraction since Twitter provides lengthy streams of data. Thus losing a small part of the accuracy compared to ensuring the filtering of data would be a better choice, as the loss is considered insignificant.

On evaluating the model, as shown in Figure 17, which represents the K and the perplexity of both the training and test datasets, the $K = 2$ gave the lowest perplexity for both training and test data, and therefore the number of topics represented was set at 2. In addition, through the observation of the training and test data results, it is shown that the perplexity of the testing data is slightly greater than the training datas, which means there is a slight over-fitting over the topics, and the training model is not very generative. However, the perplexity value is still considered to be a very promising result, according to previous research, where the best perplexity of training data achieved was 1.5 on a 1-million-word dictionary of various topics and genres [2]. Through the use of more training data, the perplexity will drop until the over-fitting is diminished.

V. Conclusion

Micro-blogging services generate billions of daily up-to-date streams of posts. Twitter's popularity enabled it to reach an audience of 100 million active users. It is a valuable source of information in disastrous events for detecting damage locations. Yet, a tweets text is unstructured, sparse, has no specific defined schema, and may contain a lot of noise. In this work, the topic modelling algorithm Latent Dircherilet Allocation LDA algorithm was used to detect tweets related to the tsunami event. A training set of 6700 tweets and a test set of 196 tweets, both manually chosen from a set of 700,000 tweet were collected. On the test data, the lowest perplexity of $\tilde{6}0.594$ was achieved at $K = 2$. An evaluation algorithm was designed to test the effectiveness of the LDA algorithm . It detected tweets with a threshold value of 0.1 difference between topic probabilities, resulting in a $\tilde{7}6\%$ accuracy with successful detection. Future work aims to add to the dataset information extraction of locations, death toll and damages' reports. In addition, more evaluation of the algorithm in a dataset with more variation of topics and in terms of its performance speed in case of real time feeds.

References

[1] Aggarwal, C. C. Data Mining: The Textbook. Springer, 2015.

[2] Aggarwal, C. C., & Zhai, C. X. Mining Text Data. New York: Springer, 2012.

[3] Alghamdi, R., & Alfalqi, K. " A Survey of topic modelling in text mining." International Journal of Advanced Computer Science and Applications. vol.6, no. 1, 2015.

[4] Alhajj, R., & Rokne, J. Encyclopaedia of Social Network Analysis and Mining. Springer International Publishing, 2014.

[5] Ashktorab, Z., Brown, C., Nandi, M. & Culotta, A. " Tweedr: Mining Twitter to inform disaster response." 11th Proceedings of the International Conference on Information Systems for Crisis Response and Management, University Park, Pennsylvania, USA. ISCRAM Association. May 18-21, 2014.

[6] Blei, D. M. "Introduction to probabilistic topic models." Communications of the ACM. Vol. 55, Issue 4, pp. 77-84. New York. April 2012.

[7] Brett, M. R. "Topic modelling a basic introduction." Journal of Digital Humanities. Vol. 2, No. 1. Winter, 2012.

[8] Guevara, H., Caragea, N., Caragea, K. & Tapia, A. "Twitter mining for disaster response: a domain adaptation approach." Proceedings of the ISCRAM Conference - Kristiansand, May 24-27, 2015.

[9] Gupta, V., & Lehal, G. S." A Survey of text mining techniques and applications." Journal of Emerging Technologies in Web Intelligence, Vol. 1, Issue 1, 2009.

[10] Jain, Saloni, "Real-time social network data mining for predicting the path for a disaster." Thesis, Georgia State University, 2015.

[11] Jo, A., Bhayani, R. & Huang, L. "Twitter sentiment classification using distant supervision." Processing, 2009

[12] Jockers, M. L. "Text Analysis with R for Students of Literature. Springer, 2014.

[13] Kas, M. & Suh, B." Computational Framework for Generating Visual Summaries of Topical Clusters in Twitter Streams." In: Pedrycz W., Chen SM. (eds) Social Networks: A Framework of Computational Intelligence. Studies in Computational Intelligence, vol 526. Springer, 2014.

[14] Kumar, S., Barbier, G., Abbasi, M. A., & LIU, H. " TweetTracker: an analysis tool for humanitarian and disaster relief." Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[15] McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

[16] Mladeni, D., Grobelnik, M., Fortuna, B., & Rusu, D. "Text stream processing." Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics. New York: ACM. June 13 - 15, 2012.

[17] Momtazi, S., & Naumann, F. "Topic modelling for expert finding using Latent Dirichlet Allocation." WIREs Data Mining and Knowledge Discovery. Vol. 3, Issue 5, pp. 346353, September/October 2013 . =

[18] Panem, S., Gupta, M. & Varma, V. " Structured information extraction from natural disaster events on Twitter." Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning WWW. pp. 1-8. ACM, 2014.

[19] Landwehr P.M., Carley K.M. "Social media in disaster relief." In: Chu W. (eds) Data Mining and Knowledge Discovery for Big Data. Studies in Big Data, vol 1. Springer, Berlin, Heidelberg, 2014.

[20] Prato, S. (2013, April 23). What is Text Mining? Infospace, The Official Blog of the Syracuse University iSchool. Retrieved from http://infospace.ischool.syr.edu/2013/04/23/what-is-text-mining/

[21] Rajman M., Besanon R. " Text mining: natural language techniques and text mining applications." In: Spaccapietra S., Maryanski F. (eds) Data Mining and Reverse Engineering. IFIP The International Federation for Information Processing. Springer, Boston, MA, 1998.

[22] Sakaki, M., Okazaki, M. & Matsuo, Y. " Earthquake shakes twitter users: real-time event detection by social sensors." Proceedings of the Nineteenth International WWW Conference (WWW2010). ACM, 2010.

[23] Guoyu Tang , Yunqing Xia. " Adaptive topic modeling with probabilistic pseudo feedback in online topic detection." Proceedings of the Natural language processing and information systems, and 15th international conference on Applications of natural language to information systems, June 23-25, 2010, Cardiff, UK.

[24] Twitter Documentation. Retrieved from https://developer.twitter.com/en/docs.

[25] Apachi. Apache Spark Documentation. Retrieved from http://spark.apache.org/docs/latest/.