Classification of Arabic Tweets for Damage Event Detection

Yasmeen Ali Ameen [1], Khaled Bahnasy [2], Adel Elmahdy [3]
[1] Business Information System Department, Faculty of Commerce
Helwan University
Dr.yasmeen@commerce.helwan.edu.eg
[2] Faculty of Computer and Information, Computer Science
Department, Ain Shams University
khaled.bahnasy@aoi.edu.eg
[3] Economic Department, Faculty of Commerce, Helwan University
Adelmahdy@link.net
Cairo, Egypt

Abstract. This paper proposes a model to analyze Arabic tweets to harness valuable information for first responders during an emergency e.g. Flood disaster. Our proposed model is designed to detect floods and assess damage during disasters using Tweets to concentrate on rescue operations. For this, we used the common classification algorithms such as SVM, RF, J48 and NB in order to classify these tweets and detect those with relevant information regarding damage. We have conducted two experiments. In the first experiment, we have implemented two classification *models A* and *B*. *Model A* classifies the tweets into *relevant* and *non-relevant*, while *model B* classifies the relevant tweets into *damage* or *not damage* (where the former refers to tweets that have information about damage and the later refers to tweets that do not have such information). Our obtained results show that Random Forest achieves best accuracy in classifying tweets into *relevant* and *non-relevant* of 83.95%, while SVM achieves the best accuracy in classifying tweets into *damage* or *not damage* of 93.39%. In the experiment B, we re-implement the two classification models of the first experiment, but we increase the size of the datasets used with both models. Therefore, we generate other two classification *models C* and *D*. We compare the performance of learning *model A* and *C* and the performance of learning *model B* and *D* in the first and second experiments respectively. Our results show that learning *model A* achieves accuracy of 83.95%, while learning *model C* achieves accuracy of 84.25% with an increase in dataset size using Random Forest classifier. Also learning *model B* achieves accuracy of 93.06%, while learning *model D* achieves accuracy of 94.52% with an increase in dataset size using SVM classifier.

Index Terms: classification, damage detection, supervised learning, twitter.

———————————— ◆ ————————————

1. Introduction

The velocity and volume of messages (tweets) in Twitter during mass emergencies make it difficult to identify useful and information, such as road closure locations, casualties, damaged infrastructures or where food and water are needed for survivors [13]. Our goal is to leverage the different machine learning techniques (e.g., information classification, and extraction) to identify and extract this information automatically. Moreover, we want people (i.e. volunteers) to label part of the incoming data to be used for the training purposes of machine learning algorithm [10]. Recent works had demonstrated the possibility to create crisis maps solely using geolocated data from Social Media (SM), to understand better and monitor the unfolding consequences of disasters [9][11][7]. All these SM-based crises mapping systems face the fundamental challenge of geoparsing the textual content of SM users to extract keywords of places/locations, thus increases the number of messages to exploit [8]. Arabic is the official language of 21 countries, and it is the major language in several areas of world that has ever increasing volume of SM users, the potential to harvest large amounts of relevant information in times of crises is real. That said, the number of Arabic social media users is not far behind. According to the Arab Social Media Report in [12], the total number of active Twitter users in the Arab world would have increased and reached around 11.1 million by March 2017. Saudi Arabia, alone, produced 40% of all tweets in the Arab world, while Egypt and Kuwait come in behind with 17% and 10 % respectively[1]. This begs for automated methods to extract SM user's interactive information from tweets during natural disasters. To this end, in this paper

- *Yasmeen Ali Ameen is currently the Lecturer in El-Gazeera High Institute for Computer and Information Systems, pursuing PhD degree in Business Information System in Hwlwan University, Egypt, E-mail: Dr.yasmeen@commerce.helwan.edu.eg*

[1] http://arabsocialmediareport.com/Twitter/LineChart.aspx

we propose an approach that is different from their approach [2] in that it handles two binary classification tasks: In the first it detects Arabic tweets that are relevant to the disaster/risk by classifying tweets into *relevant* or *not relevant*. In the second it detects Arabic tweets that have information about the *damage* caused by the disaster/risk by classifying tweets into *damage* or *not damage.* The structure of this paper is organized as follows: In section 2, we present related work about event detection, event mapping model and Arabic text mining classification. In section 3, we explain the functional workflow which is divided into five steps: (1) Data collection, (2) Data filtering, (3) Data preprocessing, (4) Data labeling, (5) Data classification. In section 4, we present Arabic floods datasets that were used. In section 5, we discuss the Results of Mining Arabic Text for Damage of the first and the second experiment. In section 6, we discuss conclusion and future work. Many researchers have proposed models for the purpose of identifying emergency events occurring on social media, mostly focused on English. Currently, there is no research addressing flood events detection based on Arabic tweets. We evaluate the performance of machine learning techniques on Arabic text collected and classified directly from Twitter and in order to answer the following question: which of the supervised learning classification algorithms can outperform more others accurately while detecting damages floods from Arabic tweets?

2. Related work

2.1 Event detection

Nasser Alsaedi [4] has presented a new detection framework for identifying ‘disruptive’ events using Tweets data. He used a Naïve Bayes classification model and an Online Clustering method. Atefeh et al. [6] presented a specific event detector that depends on specific features, such as, place, time, type, and description of tweet for Detecting floods events in real-time using supervised methods to make easy event detection. Takeshi Sakaki [16] proposed an algorithm to detect an earthquake event. He devised a classifier of tweets based on features such as the keywords. He produced a spatiotemporal model for the event that can find the center location of the earthquake.

2.2 Event mapping Model

Ashktorab et. al. [5] used the Tweedr tool to extract relevant information from tweets during a natural disaster. Using classification, extraction and clustering. It was collected from 12 different crises in the United States since 2006. Middleton et. al. [11] presented state-of- the-art system that matches preloaded location data for areas at risk to geoparse real-time tweet. The system's data was collected in the New York’s flooding in United State 2012 and Oklahoma’s tornado United States 2013. Avvenuti et. al. [8] presented CrisMap to extract disasters from tweet by adopting the classification based on the word embeddings. The maps help to identify areas that have been severely struck. It was a performed study on a recent devastating earthquake occurred in Central Italy.

2.3 Arabic text mining classification

Alabbas et. al. [2] used Arabic text classification on Twitter and Applied SVM without stemming[2] on Arabic. Mustafa et. al. [12] presented a new lexicon approach for Arabic sentimental analysis that used supervised and unsupervised technique. It was tested and evaluated using MIKA corpus.
We propose approach that is different from Alabbas approach [2] in that it handles two binary classification tasks: (see section 1). We evaluate our models to classify Arabic tweets about event detector and identify the heights performance for training models accuracy.

3. Methodology

In this paper, we used one of, the supervised Machine Learning Techniques, to classify Arabic tweets about natural flood disasters as they occur, to detect and assess damage event in these areas. The functional workflow is divided into five steps: (1) Data collection, (2) Data filtering, (3) Data preprocessing, (4) Data labeling, (5) Data classification. This architecture is presented in Figure.1

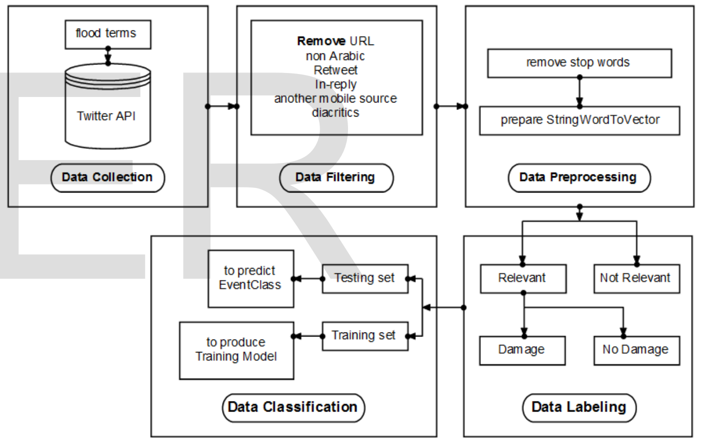


Fig. 1. The Overall structure of the Classification Model.

3.1 Data collection

We have used a dataset that consists of 230,975 tweets collected during the recent 2016 to 2017 flooding in Saudi Arabia [2]. This was carried out with the help of Twitter APIs [3] which can be accessed by Twitter user credentials (OAuth)[4]. APIs data included unstructured data in the tweet content, i.e. the text of the tweet itself, and another data that refers to the structured data such as tweet ID, in reply to the user, re-tweets and tweet location. The content of text data was used to train and test the classifier that identifies the tweets that have information about the damage caused by the disaster, while external data were used to clean and preprocess the corpus.

[2] A process of producing a root/base word. It reduces the words “retrieval”, “retrieved”, “retrieves” to the stem “retrieve”.)
[3] https://developer.twitter.com/
[4] https://developer.twitter.com/en/docs/basics/authentication/overview/oauth.html

3.2 Data filtering

We removed the following types of URLs at this step only tweets, repeated tweets and non-Arabic tweet. Some tweets contain only URL(s), this URL is in English letters and not useful in our Arabic dataset. For an example, if that tweet contains an Arabic text plus URLs, digits, non-Arabic letters, this tweet will be filtered, the Arabic text only will be saved, and the other parts will be removed. The only tweets that are directly mentioning 'torrents' or 'floods'... 'فيضان' OR 'سيول' in Arabic are used and only original tweets are included; Retweets were removed as the focus of the paper is on real-time content published for the first time. In-reply tweets and Duplicates were removed from dataset.

3.3 Data preprocessing

In the pre-processing step, and as to improve text classification by removing worthless data from training and test set, so it may include the removal of numbers and stop-words (e.g. prepositions and pronouns) [1], which do not affect the meaning of the sentence in Arabic such as (من ، فوق ، على ، و ، أنا ، شيء ، هذه ، هي، هو). Arabic requires careful strategies to normalize writing forms. Therefore, we prepare text dataset and convert it to vectors value by unsupervised attribute String-WordToVector filtering in WEKA[5], a free software machine learning tool.

3.4 Data labeling

The detection of damage in tweet is a challenging task because of the unstructured nature of the data. We need to analyze the content of tweets, discarding not relevant tweets and labeling the relevant ones according to the presence or the absence of the damage mentioned. The "*Damage*" here refers to damage to buildings and other infrastructures for example, railways, villages, towns and industrial plants, etc., and also includes injuries, missing people and casualties. In other words, damage encompasses all harmful consequences of a disaster that befell upon communities and infrastructures. The approach used for the damage detection problem is based on a two-level binary classification task.

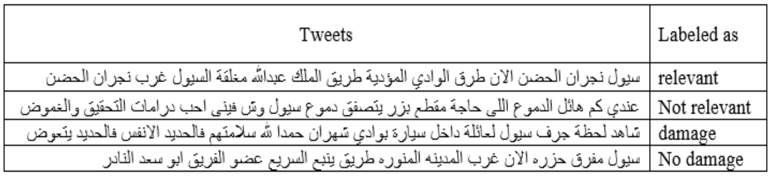
In *first level* task we are interested in identifying two classes of tweets:

Not relevant: tweets that are not related to a flood disaster. These are tweets which include relevant keywords (*flood/ torrent*) but in a different context (e.g. jokes, poems etc.). *Relevant:* tweets directly related to a flood's disaster.

In *second level* task we are also interested in identifying two classes of tweets:

No damage: tweets related to a flood disaster, but which is not carrying any information relevant to the damage evaluation. The tweet could be discussing the occurrence of a minor flooding and therefore low risk. Hence, no damage or service disruptions (e.g. transport disruption) would be expected. *Damage*: tweets related to flood disasters which carry information relevant to damage evaluation.

TABLE 1
LABELING PHASE



Tweets	Labeled as
سيول نجران الحضن الان طرق الوادي المؤدية طريق الملك عبدالله مغلقة السيول غرب نجران الحضن	relevant
عندي كم هائل الدموع اللى حاجة مقطع بزر يتصفق دموع سيول وش فيني احب درامات التحقيق والغموض	Not relevant
شاهد لحظة جرف سيول لعائلة داخل سيارة بوادي شهران حمدا لله سلامتهم فالحديد الانفس فالحديد يتعوض	damage
سيول مفرق حزره الان غرب المدينه المنوره طريق ينبع السريع عضو الفريق ابو سعد النادر	No damage

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This was done by applying two metrics: How many times a word appears in a document and the inverse document frequency of the word across a set of documents [15]. We used this measure by weka tool in preprocessing phase in order to improve SVM classifier performance we will see in the first experiment (section 5.1).

3.5 Data Classification

Data classification is a process that has two steps: (1) the training phase and (2) the testing phase where the actual event class of the instance is compared with the predicted event class. If the hit rate is acceptable, the classifier is accepted as being capable of classifying future instances with unlabeled event class [14]. This classification aims to distinguish events from irrelevant tweets. Words from each tweet are considered as features and a support vector machine (SVM) classifier was chosen for the classification task, where it was referred to its high performance in previous extensive similar experiments as demonstrated in [8].

4. DATASET

The insertion of a class for tweets that are not related to a natural disaster is necessary for the automatically collected data because not all data are related to the disaster. Therefore, the manual annotation of disaster with damage and not damage tweets is exploited to train and validate our damage detection classifier. Furthermore, following the same approach adopted in [11], we carried out an additional manual annotation of 2277 random tweets of the Arabic floods datasets with regards to mentioning the locations/places. A ten-fold cross-validation approach was adopted to train and test the methods using the WEKA toolkit for the preprocessing and classification task. Accuracy, precision, recall and F1-measure have been reported to measure the quality of tested classifiers. The next section presents the experimental results.

[5] https://www.cs.waikato.ac.nz/ml/weka/

5. Evaluation Results

5.1 First Experiment

5.1.1 Manual labeling of tweets

We run this experiment on two datasets. The first consists of some tweets that are relevant to the disaster /risk, while the second dataset is generated by considering the relevant tweets from the first dataset and manually classifying them into "damage or "not damage" in order to identify the tweets that have explicit information about damage among those relevant tweets. The first dataset consists of 2277 tweets that are manually labeled with either "relevant" or "not relevant" labels. This results in 1515 relevant tweets and 762 not relevant tweets. Every tweet in the relevant tweets is manually classified into damage or not damage by manually labeling them with either "damage" or "not damage" labels. This resuls in 270 damage tweets and 1245 not damage tweets.

5.1.2 Building supervised learning model

In this subsection we used two techniques: a fold 10 cross-validations *technique A* and split 80% train *technique B* as shown in table 2 (a), (b) and table 3 (a), (b).

In 10-fold cross-validation, the original sample is randomly partitioned into 10 equal sized subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. A 10-fold cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

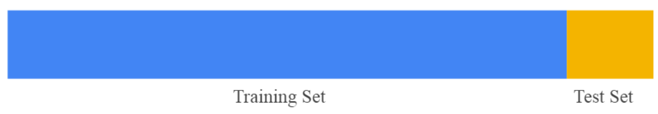


Fig. 2. Split Training and testing set.

In the split 80% train technique, we divide a dataset into training and testing data. As we work with datasets, a machine learning algorithm works in two stages. We usually split the data around 20%-80% between testing and training stages respectively (see figure 2) [6].

Table.2. (a) presents the results of training the classifiers using the first dataset in order to make them able to classify 2277 tweets into relevant or not relevant using either 10-fold cross validation or split 80% train previously mentioned. We show accuracy, precision, recall, and F1-measure (the four) metrics results for each class separately. We use SVM, Random Forest, Decision tree (DT) (J48 in Weka), Naïve Bayes (NB) classifiers to generate the training model for performing tweets classification.

Table 2 (a)
Classifying 2277 tweets into Relevant/Not Relevant using two techniques for each class.

Method	Algorithm	Accuracy	Not Relevant			Relevant		
			Pr	Re	F1	Pr	Re	F1
Cross - Validation	SVM	80.41%	0.768	0.599	0.673	0.817	0.908	0.860
	RF	80.50%	0.784	0.580	0.667	0.812	0.919	0.862
	J48	76.15%	0.709	0.493	0.582	0.778	0.897	0.833
	NB	78.91%	0.701	0.651	0.675	0.829	0.850	0.844
Split 80% train	SVM	80.43%	0.752	0.556	0.640	0.820	0.917	0.866
	RF	83.95%	0.817	0.627	0.709	0.847	0.936	0.889
	J48	77.14%	0.707	0.458	0.556	0.788	0.914	0.846
	NB	79.78%	0.671	0.690	0.681	0.858	0.847	0.852

As shown in table.2. (a) the Random Forest classifier achieves the highest accuracy, precision, recall and F1-measue with 83.95%, 83.7%, 84.0% and 83.3% respectively by *technique B*.

Table.3. (a) presents the results of training the classifiers using the second dataset in order to make them able to classify 1515 tweets into damage or not damage using either 10-fold cross validation or split 80% train. The table shows accuracy, precision, recall, and F1-measure results for each class separately. Also, by using SVM, Random Forest, J48, NB classifier to generate the training model about tweets classification for the highest accuracy.

TABLE 3 (A)
Classifying tweets into Damage /No Damage using two techniques for each class.

Tech	Algorithm	Accuracy	No Damage			Damage		
			Pr	Re	F1	Pr	Re	F1
Cross - Validation	SVM	92.08 %	0.923	0.987	0.954	0.908	0.602	0.724
	RF	89.43 %	0.890	0.995	0.940	0.947	0.415	0.572
	J48	85.61 %	0.875	0.963	0.917	0.659	0.341	0.449
	NB	86.53 %	0.925	0.911	0.918	0.601	0.648	0.624
Split 80% train	SVM	93.39 %	0.931	0.996	0.963	0.962	0.568	0.714
	RF	93.06 %	0.925	1.000	0.961	1.000	0.523	0.687
	J48	87.78 %	0.902	0.961	0.931	0.930	0.386	0.479
	NB	86.13 %	0.943	0.892	0.917	0.517	0.682	0.588

As shown in table.3. (a) the RF classifier achieves the highest Recall value by *technique B* with 1.0% not damage class. However, other classifiers which are lower than RF Precision value indicates less false positives for the RF classifier. Hence, we calculate the F1-measure which is the harmonic mean of Precision and Recall.

The highest accuracy in table.3. (a) is 93.39 % when using *technique B* by the SVM classifier. The table shows that SVM achieves the highest accuracy, precision, recall and F1-measure with 93.39%, 1.0 for damage class, 1.0 for not damage class and 0.963 for not damage class respectively using *technique B* while the performance of the J48 classifier using *technique A* scored the lowest F1-measure with 0.449. We compared our results with the work of Al Abbas et. al. [2], where

[6] https://developers.google.com/machine-learning/

they found that testing Colloquial Arabic without stemming had achieved the highest precision with 95.9% in their study for the SVM classifier, while with stemming had achieved the lowest accuracy with 90.7%. In contrast, our results indicated that testing informal Arabic text without stemming, but using TF-IDF technique, achieves the highest accuracy with 93.39% for the SVM classifier in *technique B* while it achieves the highest F1-measure with 0.963 Our test accuracy increased by 2.69 % to 93.39% as compared to Al abbas's accuracy of 90.7%.

5.1.3 Testing the model

We classify dataset automatically by inserting non labeled tweets into the model and by seeing how the model will identify either the relevant tweets among tweets in the first dataset or the damaged tweets among tweets in the second dataset (see figure.3).

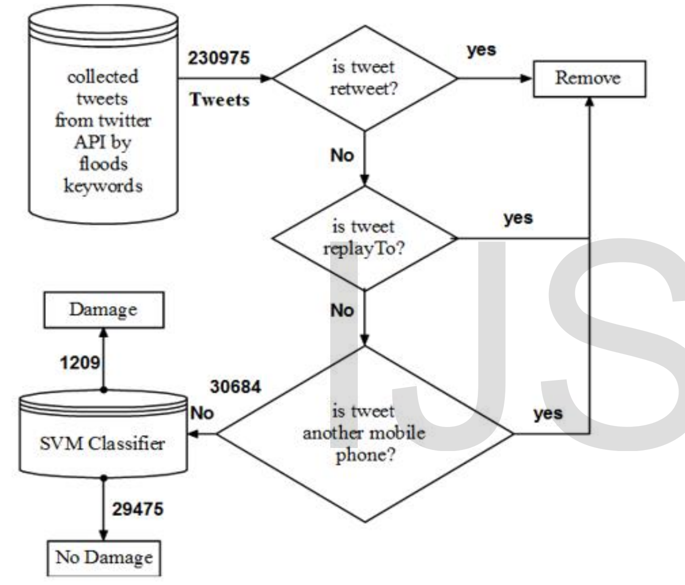


Fig.3. Building a model for filtering and classifying tweets into damage and no damage.

5.2 Analysis and discussion of the Second Experiment

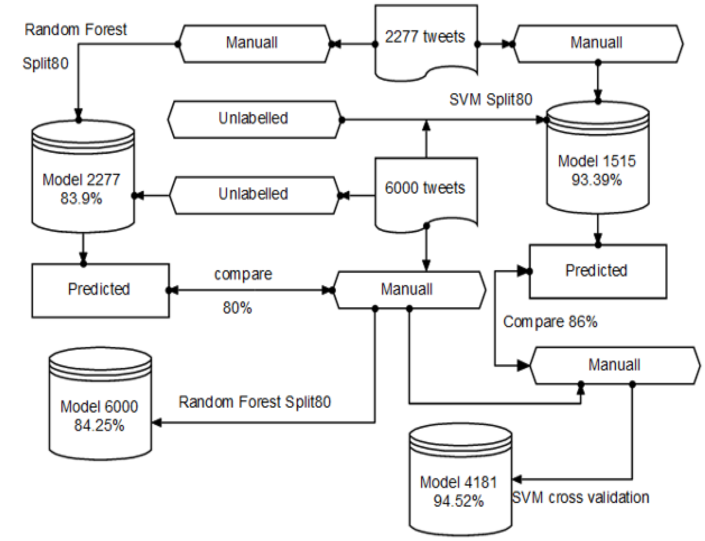


Fig.4. Comparison of four Models.

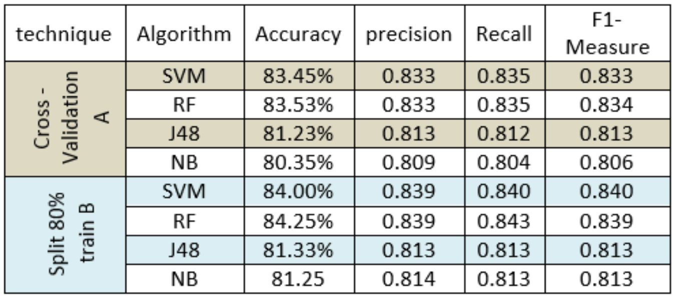
Figure 4 shows classifiers' performance and evaluation in classifying the tweets when increasing the size of the two datasets used in the first experiment. After generating the two learning models *A* and *B*, We also generated another two classification models in second experiment, but The size of first dataset is increased to be 6000 tweets (4181 relevant versus non relevant 1819 tweets), while the size of the second dataset is increased to be 4181 tweets (1070 damage tweet versus 3111 non damage tweet). The same procedure is used to build and test the classifiers on both resized datasets. Table 4 and 5 show the performance evaluation results on both resized datasets (see tabels below).

Also, figure 4 shows 6000 unlabeled tweets test sets that are entered to learning *model A*. Then, the predicted data are compared to the same 6000 labeled tweets that are manually classified where the accuracy is 80%. Similarly, 4181 unlabeled tweets test set on learning *model B* and the predicted data are compared to the same 4181 labeled tweets that are manually classified where the accuracy is 86%.

We noticed that when learning *Model A* was evaluated after entering 6000 unlabeled tweets to it to test its performance, its accuracy decreased from 83% to 80%. Similarly, when learning *Model B* was evaluated after entering 4181 unlabeled tweets to test its performance, its accuracy decreased from 93.39% to 86%. It does not a negative outcome (see figure 4).

TABLE 4

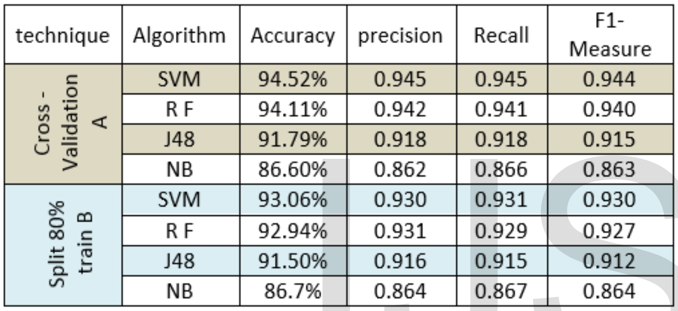
A COMPARISON OF CLASSIFYING 6000 TWEETS INTO RELEVANT/NOT RELEVANT USING SVM, RANDOM FOREST, J48, NB CLASSIFIER.



technique	Algorithm	Accuracy	precision	Recall	F1-Measure
Cross - Validation A	SVM	83.45%	0.833	0.835	0.833
	RF	83.53%	0.833	0.835	0.834
	J48	81.23%	0.813	0.812	0.813
	NB	80.35%	0.809	0.804	0.806
Split 80% train B	SVM	84.00%	0.839	0.840	0.840
	RF	84.25%	0.839	0.843	0.839
	J48	81.33%	0.813	0.813	0.813
	NB	81.25	0.814	0.813	0.813

TABLE 5

A COMPARISON OF CLASSIFYING 4181 TWEETS INTO DAMAGE/NO DAMAGE USING SVM, RANDOM FOREST, J48, NB CLASSIFIER.

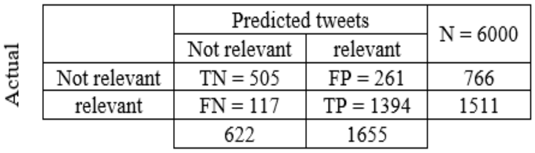


technique	Algorithm	Accuracy	precision	Recall	F1-Measure
Cross - Validation A	SVM	94.52%	0.945	0.945	0.944
	R F	94.11%	0.942	0.941	0.940
	J48	91.79%	0.918	0.918	0.915
	NB	86.60%	0.862	0.866	0.863
Split 80% train B	SVM	93.06%	0.930	0.931	0.930
	R F	92.94%	0.931	0.929	0.927
	J48	91.50%	0.916	0.915	0.912
	NB	86.7%	0.864	0.867	0.864

Thus, Table 4, 5 shows the learning models testing results. As a result, the heighest accuracy with RF relevant classifier achieves 83.95% for the learning *model A* by (split 80% train) *technique B*. When the dataset increased to 6000 tweets, the RF relevant classifier accuracy increased to 84.25% for the learning *model C* by (split 80% train) *technique B*. In contrast, the highest accuracy with SVM damage classifier achieves 93.06% for learning *model B* by split 80% train *technique B* when the dataset increased to 4181 tweets, and the SVM damage classifier accuracy becomes 94.52% for learning *model D* by (cross-validation 10 fold) *technique A*.

TABLE 6

CONFUSION MATRIX FOR COMPARING BETWEEN 2277 LABELED AND THE PREDICTED TWEETS ON LEARNING MODEL C.



		Predicted tweets		N = 6000
		Not relevant	relevant	
Actual	Not relevant	TN = 505	FP = 261	766
	relevant	FN = 117	TP = 1394	1511
		622	1655	

Table 6 presents the confusion matrix that used to describe the performance of a classification *model C (6000 tweets)* for RF relevant classifier by split 80% train *technique B* with 2277 tweets test set for which the true values are known.

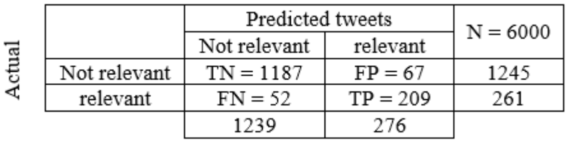
And compare between 2277 labeled tweets and the same predicted tweets resulting from learning *model C.*
Number of actual not relevant tweets is: 766.
Number of actual relevant tweets is: 1511.
True positives (TP): 1394 tweets. These are cases in which we predicted relevant tweets (they are relevant), and they are actual the relevant.
True negatives (TN): 505 tweets. We predicted not relevant, and they are not the relevant.

Accuracy = (TP + TN) / total (1)

(505+ 1394) / 2277 = 83.95%

TABLE 7

CONFUSION MATRIX FOR COMPARING BETWEEN 1515 LABELED AND THE PREDICTED TWEETS ON LEARNING MODEL D.



		Predicted tweets		N = 6000
		Not relevant	relevant	
Actual	Not relevant	TN = 1187	FP = 67	1245
	relevant	FN = 52	TP = 209	261
		1239	276	

Table 7 the performance of a classification *model D (4181 tweets)* for SVM relevant classifier by cross-validation 10-fold *technique A* with 1515 tweets test set for which the true values are known.
And compare between 1515 labeled tweets and the same predicted tweets resulting from learning *model D.*
Number of actual no damage tweets is: 1254
Number of actual damage tweets is: 261
TP: 209 damage tweets.
TN: 1187 no damage tweets.

Accuracy = (209+ 1187) / 1515 = 92.14% (2)

6. CONCLUSION AND FUTURE WORK

The main contribution of our paper was to discuss a variety of text classification techniques using Arabic text extracted from tweets as dataset. This paper used text classification technique by using floods disaster Arabic text as dataset. For this study we focused on identifying two-level binary classification task. Then identified two classes of tweets in the *first level*: "*Relevant*" and "*Not Relevant*", created a testing model and validation of dataset. We further redefined relevant tweets and classified them into two classes of tweets in *second level*: "*Damage*" or "*No Damage*". A testing model was created, and a validation of Arabic floods tweets dataset was carried out. We focused on testing model in the second level because it is the most important part of our research for detecting damages and classifying severe tweets. The SVM classification technique used here was primarily used by many researchers to classify English text. The few studies that were done on Arabic texts datasets, using the same technique, were not validating the performance of this technique. The classification algorithms that have been tested in this paper are: SVM, Random Forest, J48 and NB in the two successive experiments. SVM achieved the most accurate results. In the second experiment, we compared between four models' performance after evaluation. Our results show learning *model A* has been achieved accuracy of

83.95%, while learning *model C* has been achieved accuracy of 84.25% with an increase in dataset size using Random Forest classifier. Also learning *model B* has been achieved accuracy of 93.06%, while learning *model D* has been achieved accuracy of 94.52% with an increase in dataset size using SVM classifier.
We have recommended the use of Geoparsing for future researches to the map most stricken areas from tweets. Also, the researchers should consider time and place of floods to locate them on the map.

References

1. Alabbas, W., Al-Khateeb, H. M., & Mansour, A. (2016). Arabic text classification methods: Systematic literature review of primary studies. In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt) (pp. 361-367). IEEE.
2. Alabbas, W., al-Khateeb, H. M., Mansour, A., Epiphaniou, G., & Frommholz, I. (2017). Classification of colloquial Arabic tweets in real-time to detect high-risk floods. In 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media) (pp. 1-8). IEEE.
3. Al-Omari, A., & Abuata, B. (2014). Arabic light stemmer (ARS). Journal of Engineering Science and Technology, 9(6), 702-717.
4. Alsaedi, N., & Burnap, P. (2015). Arabic event detection in social media. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 384-401). Springer, Cham.
5. Ashktorab, Z., Brown, C., Nandi, M., & Culotta, A. (2014). Tweedr: Mining twitter to inform disaster response. In ISCRAM, International Conference on Information Systems for Crisis Response and Management.
6. Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. Computational Intelligence, 31(1), 132-164.
7. Avvenuti, M., Cresci, S., Del Vigna, F., & Tesconi, M. (2016). Impromptu crisis mapping to prioritize emergency response. Computer, 49(5), 28-37.
8. Avvenuti, M., Cresci, S., Del Vigna, F., Fagni, T., & Tesconi, M. (2018). CrisMap: a big data crisis mapping system based on damage detection and geoparsing. Information Systems Frontiers, 1-19.
9. Goolsby, R. (2010). Social media as crisis platform: The future of community maps/crisis maps. ACM Transactions on Intelligent Systems and Technology (TIST), 1(1), 7.
10. Imran, M., Lykourentzou, I., Naudet, Y., & Castillo, C. (2013). Engineering crowdsourced stream processing systems. arXiv preprint arXiv:1310.5463.
11. Middleton, S. E., Middleton, L., & Modafferi, S. (2014). Real-time crisis mapping of natural disasters using social media. IEEE Intelligent Systems, 29(2), 9-17.
12. Mustafa, H. H., Mohamed, A., & Elzanfaly, D. S. (2017). An Enhanced Approach for Arabic Sentiment Analysis. International Journal of Artificial Intelligence and Applications (IJAIA), 8(5).
13. Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014). CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In (ICWSM) Conference on Web and Social Media.
14. Omar, A., Mahmoud, T. M., & Abd-El-Hafeez, T. (2018). Building Online Social Network Dataset for Arabic Text Classification. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 486-495). Springer, Cham.
15. Paralic, J., & Bednar, P. (2003). Text mining for document annotation and ontology support. Intelligent Systems at the Service of Mankind, 237-248.
16. Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web (pp. 851-860). ACM.