Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods

Waleed Alabbas, Haider M. al-Khateeb, Ali Mansour, Gregory Epiphaniou, Ingo Frommholz
Institute for Research in Applicable Computing (IRAC)
University of Bedfordshire
Luton, United Kingdom
waleed.alabbas@study.beds.ac.uk, haider.alkhateeb@beds.ac.uk, ali.mansour@beds.ac.uk, gregory.epiphaniou@beds.ac.uk, ingo.frommholz@beds.ac.uk

Abstract*– Twitter has eased real-time information flow for decision makers, it is also one of the key enablers for Open-source Intelligence (OSINT). Tweets mining has recently been used in the context of incident response to estimate the location and damage caused by hurricanes and earthquakes. We aim to research the detection of a specific type of high-risk natural disasters frequently occurring and causing casualties in the Arabian Peninsula, namely 'floods'. Researching how we could achieve accurate classification suitable for short informal (colloquial) Arabic text (usually used on Twitter), which is highly inconsistent and received very little attention in this field. First, we provide a thorough technical demonstration consisting of the following stages: data collection (Twitter REST API), labelling, text pre-processing, data division and representation, and training models. This has been deployed using 'R' in our experiment. We then evaluate classifiers' performance via four experiments conducted to measure the impact of different stemming techniques on the following classifiers SVM, J48, C5.0, NNET, NB and k-NN. The dataset used consisted of 1434 tweets in total. Our findings show that Support Vector Machine (SVM) was prominent in terms of accuracy (F1=0.933). Furthermore, applying McNemar's test shows that using SVM without stemming on Colloquial Arabic is significantly better than using stemming techniques.

Keywords: Arabic text classification; big data; Colloquialism; Event detection; Twitter; Real-time; Stemming; SVM.

I. INTRODUCTION

Twitter remains one of the most popular social media platforms to date. This can be evidenced with the huge and increasing real-time text posted by users on a day-to-day basis. Moreover, the correlation of this data could translate into a wealth of information [1]. Twitter's popularity and features (e.g. available APIs and metadata) have paved traditional offline knowledge discovery methods in favour of data-driven science focusing on online freely-available unstructured data. Twitter revolves around the concept of microblogging allowing users to post short texts with an option to include a link to a website, photos, or videos.

This study investigates the detection of a specific type of high-risk natural disasters, namely 'floods' by classifying informal (colloquial) Arabic text which have received very limited attention [2]. Colloquial Arabic text is a collective term for the spoken languages or dialects of people throughout the Arab world, and is therefore widely used on social media. It is generally characterized to be highly unstructured, inconsistent, and difficult to process. In contrast, Modern Standard Arabic (MSA) is the official Arabic language that is taught in schools and defined as the accepted medium for formal communications. MSA is therefore more understood, has specific grammar rules, structured and more consistent.

However, the purpose of this work is event detection with a focus on remote regions where people would tweet in colloquial Arabic. Those who are living in remote rural areas are usually very close to an event as it happens, which means they would be able to sense and report events first and in real-time. Therefore, early critical evidence of a specific natural disaster such as floods could come from updates and tweets in colloquial Arabic. We argue in favour of further contribution to this research area, and consider a specific case study to investigate different aspects of event detection based on a corpus consisted of informal Arabic tweets.

Emergencies related to high-risk events require exceptionally fast incident response and decision-taking based on first-hand information, and Twitter has eased information flow for decision makers and enabled Open-source Intelligence (OSINT). Short messages posted on social media (Tweets) can typically reflect these events as they happen; Kryvasheyeu et al. [3] discussed how Tweets' analysis outperformed traditional methods used by formal agencies such as the Federal Emergency Management Agency (FEMA) in estimating the location and severity of damages by the Hurricane Sandy. People are using Twitter to report real-life events which gives researchers the opportunity to design and develop quality event detection systems. For instance, Sakaki et al. [4] have developed an earthquake reporting system in Japan by monitoring and filtering English and Japanese tweets. Their system detected earthquakes much faster than Japan Meteorological Agency (JMA).

While this approach towards event detection may be customised to various case studies, we will focus on floods in this work. Recently, floods were the most frequent natural disasters in the Arabian Peninsula, they occur several times a year and cause severe damage to both life and property. In one of the worse scenarios in 2009, over 120 civilians were reported to have been killed [5]. Part of the reason to have severe damages was interpreted as the lack of real-time information on the event and inefficient Decision Support

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

Systems (DSS) to utilize available resources more effectively [5] [6].

Many researchers have proposed models and techniques for the purpose of identifying specific events occurring on social media, mostly fixated on English as the communication medium; however, to the best of our knowledge, and further to the very limited number of research on Colloquial Arabic in general, there is no research addressing flood detection based on Arabic tweets. We evaluate the performance of machine learning methods and techniques on Colloquial Arabic text collected and classified directly from Twitter and intend to answer the following questions: which of the supervised learning classification algorithms can more accurately outperform others while detecting high-risk floods from Colloquial Arabic tweets? And to what extent could stemming techniques be useful to enhance classification?

Accuracy, recall, F1-measure, and precision have been reported to measure the quality of tested classifiers. These can be calculated as shared below while Table 1 includes further details to clarify our measures through a confusion matrix:

1) Accuracy = (TP+TN) / (TP+FP+TN+FN)
2) Precision = TP / (TP+FP)
3) Recall = TP / (TP+FN)
4) F1 = 2× (Precision × Recall) / (Precision + Recall)

Table 1- Confusion matrix

Confusion matrix		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative(FN)
	Negative	False Positive (FP)	True Negative (TN)

In addition to the above measures, McNemar's test will be used to calculate statistical significance. It has been used in various studies in the area of machine learning and proved reliable to determine if the best classifier is significantly better than others [7] [8].

The remainder of this paper covers related works in section II, methodology in section III which also discusses data collection and extraction from Twitter, data labelling, text cleaning, building a Term Frequency–Inverse Document Frequency (TF-IDF) matrix, training models, and finally classification of documents. Section IV contains discussion and visualisation of classifiers' performance. Finally, conclusions are shared in Section V.

II. Related works

A. Colloquial Arabic text classification

Although Text Classification (TC) remains an active research area with novel techniques designed and thoroughly tested on English scripts [9], there seems to be very little work done on Arabic text compared to other languages, especially colloquial text [2].

However, as a growing area of research, recent publications include Huang's work [10] who aimed to improve the classification of Arabic dialects through the utilisation of metadata such as records related to the geographical information of social media users. Another technique combined three classifiers by computing the model scores of a weakly supervised, strongly supervised, and semi-supervised classifiers. This combination provided significant improvement by means of classification accuracy to both MSA and Colloquial Arabic test sets. An experiment by Kaati et al. [11] was conducted to detect any tweet supporting the act of terrorism (a malicious agenda). The authors used data-dependent and data-independent features as part of the 'features selection' process on Arabic and English tweets. Their results show that utilising AdaBoost (Adaptive Boosting) improves the performance on English datasets but not in the case of Arabic tweets. Further, a framework combining classification and clustering techniques [12] was proposed to enable the detection of real-world events from Arabic tweets. The authors have assessed their framework against [13] and [14]; the outcome of this comparison demonstrated the effectiveness of their proposed framework, it outperformed other approaches in the Normalized Discounted Cumulative Gain (NDCG) and precision evaluation measures. Further, the work of Al-Badarneh et al. [15] investigated the impact of using different indexing techniques (full-word, stem, and root) when classifying Arabic text. It concludes that using 'full-word' or 'stem' outperforms 'root' when applied with the Naïve Bayes (NB) classifier.

Likewise, Shoukry and Rafea [16] examined the sentiment classification of Arabic text at a sentence-level by comparing Support Vector Machine (SVM) and NB Classifiers before and after removing stop words. Their corpus of 1000 tweets comprised the usual classes of positive and negative tweets for training purposes. The conclusion of this study suggests that SVM has better results compared to NB. In [17], the performance of two classifiers were tested, namely NB and Decision Tree. The best classification model was obtained with the help of Decision Tree. The results demonstrate that light stemming is more suitable to use with Colloquial Arabic text than other features selection techniques. Another comparison study examining different classifiers was presented in [18]. A corpus of 3700 Arabic tweets was collected and partitioned into three different classes. The conducted experiments showed that SVM outperformed k-NN, NB and Decision Tree in terms of accuracy. In [19] three classifiers SVM, NB, and k-NN were used to investigate the impact of feature selection on the performance of these classifiers. The authors applied different feature selection techniques including light stemming, root stemming, and character n-grams. The experiment utilised a dataset consisting of 1000 positives and 1000 negative tweets. Results concludes that 3-ngram and 4-ngram models without or combined with tokens (the word model) yield the best results. Regarding the classifiers performance, SVM outperformed other classifiers when applying all the feature selection techniques included in their study.

B. Event Detection

The notion of an event refers to a unique incident or occurrence happening at some point in time [20]. Event detection aims at finding, and following, events from

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

conventional media sources or social media sources. Atefeh and Khreich [21] presented a survey of techniques applied for Twitter-based event detection. They classified event types as 'specified' or 'unspecified', detection task as 'Retrospective Event Detection (RED)' or 'New Event Detection (NED)', and detection method as 'supervised' or 'unsupervised' methods. Since our work focuses on detecting flood events in real-time, we will then be more interested in 'specified events' utilising 'supervised' methods to facilitate NED.

Specified events rely on partially or fully adding predefined filters based on information already known about the targeted event. These filters (or metadata) could include location, time and keywords [21] [4]. While *NED* on Twitter involves the discovery of new Twitter streams in near real-time, and *Supervised learning methods* rely on labelled training examples.

Manually labelling a large number of Twitter messages for training purposes is a labour-intensive and time-consuming task, It is also more feasible for specified events compared to unspecified events. Nevertheless, for experts to annotate a dataset of reasonable size or for processing resources to be utilised efficiently, good event descriptors and thoroughly tested filters became a necessity to reduce the amounts of irrelevant messages.

Several studies have been conducted to investigate the usability of NED. For instance, real-time detection of earthquakes and typhoons using tweets as 'social sensors' [4] was tested on Japanese (considering the geographical location of interest). The research in question formulated NED as a classification problem and trained a SVM classifier on a manually labelled Twitter dataset comprising positive events (earthquakes and typhoons) and negative events (other events or non-events). Likewise, Popescu and Pennacchiotti [22] used a large set of linguistic, structural, sentiment, and controversy features from Twitter and external features such as Web-news controversy for the detection of controversial events about celebrities. Furthermore, Alsaedi and Burnap [23] proposed a framework based on NB to detect 'disruptive' events from Arabic tweets.

III. METHODOLOGY

This section presents the methodology used to classify Arabic tweets to determine events related to high-risk floods. Although there are several data mining tools available such as Weka, RapidMiner, and Orange; R tools were selected for data analysis since all of the required classification algorithms are integrated in R. R is a free language and environment for statistical computing and to graphics expedites the text classification process: everything from tweets extraction to training and classifying has been supported to make machine learning with textual data more accessible [24]. R has the capability to extract and manipulate tweets from Twitter automatically using the TwitterR package [25]. The core functions of the program will be discussed and the use of R packages with the relevant coding examples will be demonstrated.

The key phases in the methodology have been illustrated in Figure 1. As an overview, we have identified how to connect and access Twitter API, how to search for queries and how to remove retweets, which formed part of the data collection stage. In the second phase, data labelling was performed to distinguish between event related and non-event tweets. To mitigate bias, the text pre-processing stage has been applied to remove noise, this has helped to clean the dataset and reduce size.

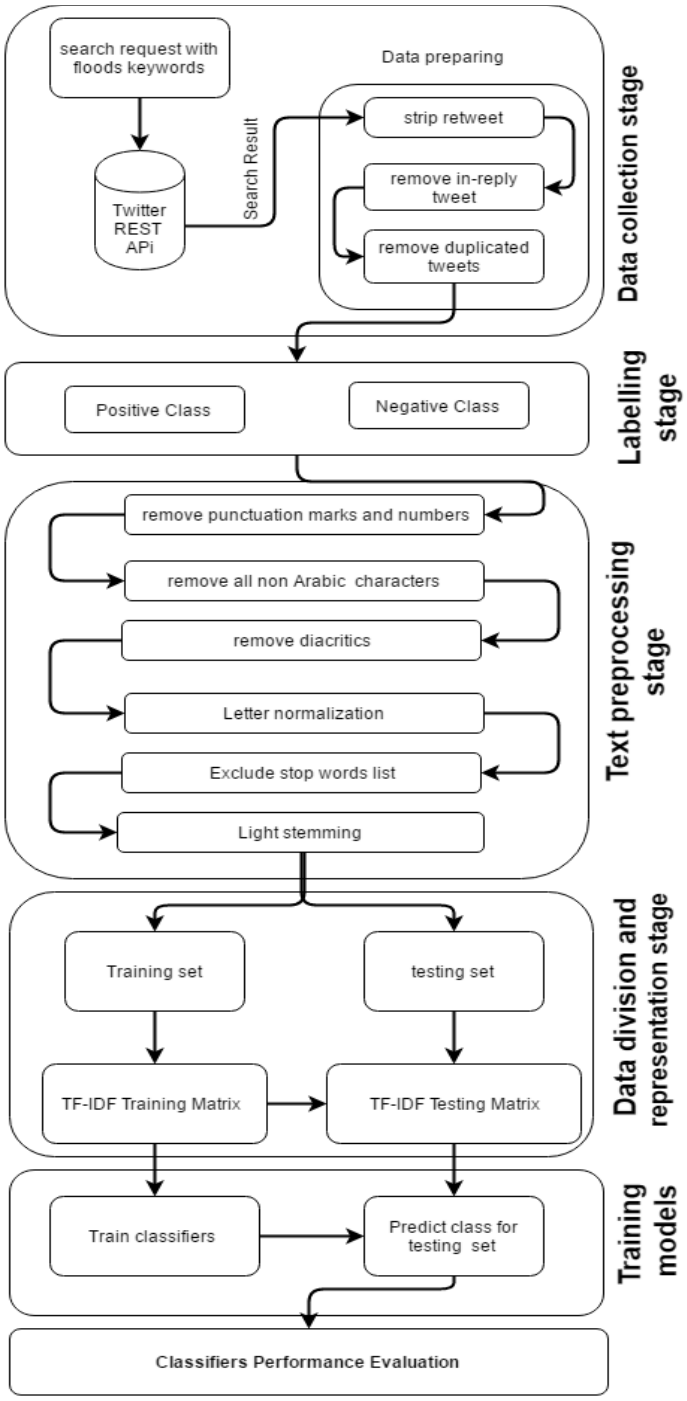

Figure 1- Main steps in tweets classification

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

In the consequent phase of this experiment, the dataset was divided to train classifiers and build the Term Frequency–Inverse Document Frequency (TF-IDF) matrix. After the classifiers were trained with the training dataset, four experiments had been designed to test the performance of each classifier. Further discussion and technical reflection on each of the phases will be provided in the subsequent sections.

A. Data collection stage

Data collection has been conducted with the help of Twitter REST APIs1 which can be accessed using Twitter user credentials via Open Authentication (OAuth). Returned data from the APIs included internal data that refers to unstructured data in the tweet content, i.e. the text of the tweet itself, and external data that refers to the structured data behind tweets such as tweet ID, retweets, in replay to user, tweet language and tweet location. The content of internal data was used to learn and test the classifiers, while external data were used to clean and prepare the corpus. Only tweets directly mentioning 'floods' or 'torrents' ('سيول OR فيضانات') in Arabic were collected 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. Duplicates and In-reply tweets were also removed from the data. The following steps demonstrate the implementation in more details:

Step1: OAuth authentication using *setup_twitter_oauth* function to connect with Twitter API.

Step2: Issue a search request using *searchTwitter* function to collect the tweets based on the key words related to the floods in Arabic language.

```
tweets <- searchTwitter("سيول OR فيضانات", n=5000,
                        lang="ar", since="2016-01-01")
```

Step3: Remove retweeted tweets from our corpus using *strip_retweets* function to provide a pure set of tweets.

```
tweets <- strip_retweets(tweets)
```

Step4: The outcome from *searchTwitter* function is list, *twListToDF* function converts the list to data frame.

```
tweets.df <- twListToDF(tweets)
```

Step5: Remove In-reply tweets.

```
tweets.df <- tweets.df[is.na(tweets.df$replyToUID),]
```

Step6: Remove Duplicated tweets.

```
tweets.df <- tweets.df [duplicated(tweets.df $text) ==
FALSE,]
```

[1] https://dev.Twitter.com/rest/public

B. Data labelling stage

The supervised learning of classification algorithms requires manual data labelling. To facilitate this, a new column *Event_Class* was inserted in the data collected to indicate the classification of tweets as positive or negative. A positive class includes tweets describing high-risk floods. These would typically cause damage to infrastructure (railways, villages, towns, industrial plants etc), occur in areas where causalities have been reported in earlier years, or described as damaging in the tweet itself. Further, a negative class would include any tweet mentioning floods but could not be labelled or referred to as a high-risk event for any of the following reasons:

- The tweet is discussing a non-real time event (floods or torrents occurred in the past).
- The tweet could be discussing the occurrence of a weak and therefore low-risk flood. Therefore, no damage or service disruptions (e.g. transport disruption) would be expected.
- A tweet including relevant keywords (flood/ torrent) but in a different context (e.g. poems, jokes etc).

C. Text pre-processing stage

Unstructured text such as tweets requires pre-processing before it can be analysed. Pre-processing is actually a trial to improve text classification by removing worthless data. Due to this, a text mining (tm) package [26] and Arabic stemmer *arabicStemR* package [27] were installed. The *arabicStemR* package contains functions to allow stemming and cleaning Arabic texts. The following steps were implemented as part of the pre-processing stage:

Step1: Remove new line characters

```
tweets.df$text = sapply(tweets.df$text
                        function (x) removeNewlineChars(x))
```

Step2: Remove punctuation marks.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) removePunctuation(x))
```

Step3: Remove numbers.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) removeNumbers(x))
```

Step4: Clean all characters that are not Latin or Arabic.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) cleanChars(x))
```

Step5: Clean Latin characters.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) cleanLatinChars(x))
```

Step6: Removes diacritics from Arabic Unicode text.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) removeDiacritics(x))
```

Step7: Reduction of the number of words through standardize different hamzas on alif seats (ا, أ, and إ are converted to ا)

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) fixAlifs(x))
```

Step8: Exclusion of words adding no value to the Text Classification scheme (stop-words) such pronouns and prepositions, which frequently occur in all tweets.

```
tweets.df$text = sapply(tweets.df$text,
                        function (x) removeStopWords(x)$text)
```

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

In this study, four experiments have been implemented with different criteria to evaluate the impact of the stemming process on Colloquial Arabic text as shared below:

Experiment A: Colloquial Arabic text without stemming.

Experiment B: Colloquial Arabic text with Light 10 stemmer.

```
tweets.df2$text = sapply(tweets.df$text,
                         function (x) doStemming(x)$text)
```

Experiment C: Colloquial Arabic text with the removal of common prefix.

```
tweets.df3$text = sapply(tweets.df4$text,
                         function (x) removePrefixes(x))
```

Experiment D: Colloquial Arabic text with the removal of common suffix.

```
tweets.df4$text = sapply(tweets.df5$text,
                         function (x) removeSuffixes(x))
```

In linguistic morphology, stemming is typically performed on datasets to reduce inflected words to their word stem or base. In our experiment, this step has not been used not to produce the linguistic root of a given Arabic surface form, but to remove the most frequent suffixes and prefixes. Common prefix or suffix strings such as ال، بال، ان، ات، ة are removed. We have considering implementing the latest Light 10 Stemmer because it outperforms other stemmer techniques especially with Colloquial Arabic text [15] [17].

Table 2: Data division

Set	Positive class	Negative class	Total
Training set	315	688	1003
Testing set	136	295	431
Entire set	451	983	1434

D. Data division stage

The number of tweets after the pre-processing stage is 1434 tweets. The corpus was divided into two sets, one for training (70% of the data) and the rest for testing purposes (30% of the data) using the *createDataPartition* function [28] which can be used to preserve the distribution of the classes in the training and test sets as shown in Table 2. Outlining training data is inevitable to build the classifiers and fit the parameters. On the other hand, testing data were used to compare the performances of the classifiers that were created based on the training set.

```
selected <- c("EventClass","text")
tweets.df <- tweets.df[selected]
indexes<- createDataPartition(tweets.df[,1],p=0.7,list=FALSE)
train.data <- tweets.df [indexes,]
test.data <- tweets.df [-indexes,]
```

By the end of this process, words are ready for indexing and to calculate the TF-IDF weight.

E. Data representation

The classifiers cannot directly use Text. Instead, it is a requirement that we first extract and generate the frequency list of the dataset features (single words, light stemming, words without prefixes and words without suffixes). After that TF (Term Frequency) and DF (Document Frequency) are calculated to generate the training and testing matrix cells using the TF-IDF weighting method. TF-IDF assigns higher weights to distinguish terms in a document. In other words, the more a term occurs in a document, the more it is representative of the content of the document. Moreover, the more the documents contain the terms, the less informative it becomes [29].

```
train.dtMatrix <- create_matrix(train.data[,2],
                                weighting = tm::weightTfIdf)
test.dtMatrix <- create_matrix(test.data[,2],
                               weighting=tm::weightTfIdf,
                               originalMatrix = train.dtMatrix)
```

F. Training models

In this step, the training matrix that contains the selected terms and their corresponding TF-IDF weights in each tweet of the training data is used to train the classification algorithms by learning the characteristics of every class from a training set of tweets. The training process constructs a classification model that will be tested.

1. Support Vector Machine (SVM)

```
svm.classifier <- svm(train.dtMatrix, as.factor(train.data[,1]))
```

2. Decision Tree (j48)

```
J48.classifier <- train(train.dtMatrix, as.factor(train.data[,1]),
                        method = "J48")
```

3. Decision Tree (C5.0)

```
c5.classifier <- C5.0(train.dtMatrix, as.factor(train.data[,1]))
```

4. Neural Networks (NNET)

```
Nnet.classifier<- train(train.dtMatrix, as.factor(train.data[,1])),
                        method = "pcaNNet")
```

5. Naive Bayes (NB)

```
nb.classifier <- naiveBayes(train.dtMatrix,
                            as.factor(train.data[,1]))
```

6. K-Nearest Neighbor (k-NN)

```
kknn.model <- train(train.dtMatrix,
                    as.factor(train.data[,1]), method = "kknn")
```

After that, the classification model consequently predicts a class for tweets in the testing set using the *predict* function. The same terms that were extracted from the training data and the same weighing methods were used to test the classification model.

```
predict(classifier_name, test.dtMatrix)
```

IV. RESULTS ANALYSIS AND DISCUSSION

Experiments performed included a total of 431 test tweets to be classified against 2 categories. Table 3 shows accuracy, precision, recall and F1-measure results that were obtained when running the selected classification algorithms on the processed corpus using four testing experiments. As shown in Table 3, the SVM classifier has achieved the highest F1-measure in experiment A with 93.3%. The second highest F1-measure in experiment A equals 89.2% and it was achieved using the C5.0 classifier, while the performance of the NB classifier scored the lowest F1-measure with 81.8%. A cross-comparison of the data produced in each experiment reveals that experiment A, testing Colloquial Arabic without stemming, has achieved the highest F1-measure as shown for the SVM, J48, C5.0 and NB classifiers. However, the removal of the common prefix in Experiment C has increased the performance for classifiers NNET and k-NN 1.2% and 0.6% respectively. However, it has decreased the F1-measure value 3.2% for SVM.

Recall values address the following question 'Given high-risk floods event, will the classifier detect it?'. On the contrary Precision Values answer the following question "Given a positive prediction from the classifier, how likely is it to be correct?". Table 3 shows that NNET has achieved the highest Recall value in experiment C with 94.5%. However, SVM, J48, C5.0 and NB classifiers outperform NNET in term of Precision, which indicates more false positives for the NNET classifier. Hence, we have calculated the F1-measure which is the harmonic mean of Recall and Precision as demonstrated earlier in section I.

Table 3- Algorithm accuracy, precision, recall and F-score.

	Algorithm	Accuracy	Precision	Recall	F1
Experiment A	SVM	0.907	0.959	0.910	0.933
	J48	0.837	0.945	0.833	0.885
	C5.0	0.846	0.946	0.844	0.892
	NNET	0.830	0.813	0.930	0.867
	NB	0.716	0.935	0.728	0.818
	k-NN	0.781	0.745	0.920	0.823
Experiment B	SVM	0.860	0.955	0.857	0.903
	J48	0.842	0.884	0.884	0.884
	C5.0	0.832	0.874	0.880	0.876
	NNET	0.801	0.803	0.897	0.847
	NB	0.693	0.867	0.733	0.794
	k-NN	0.773	0.743	0.902	0.814
Experiment C	SVM	0.856	0.969	0.843	0.901
	J48	0.835	0.935	0.841	0.885
	C5.0	0.839	0.894	0.874	0.883
	NNET	0.842	0.816	0.945	0.875
	NB	0.714	0.911	0.734	0.812
	k-NN	0.784	0.769	0.900	0.829
Experiment D	SVM	0.891	0.949	0.894	0.920
	J48	0.839	0.918	0.857	0.886
	C5.0	0.846	0.932	0.856	0.892
	NNET	0.798	0.766	0.926	0.838
	NB	0.71	0.878	0.744	0.805
	k-NN	0.763	0.739	0.897	0.810

Figure 2 demonstrates the performance of the classifiers within the conditions of all four experiments, it also shows that light stemming has a negative impact on the accuracy of the classifiers. Light stemming was designed to reduce the index terms by removing a set of prefixes and suffixes, without trying to find roots. Light stemming is mainly dependent on the understanding of the Arabic morphology [29], while this study covers informal and dialectal Arabic text (non-standard orthography, vocabulary, morphology, and syntax) which could explain the reasons behind the errors produced by the light stemming algorithm.

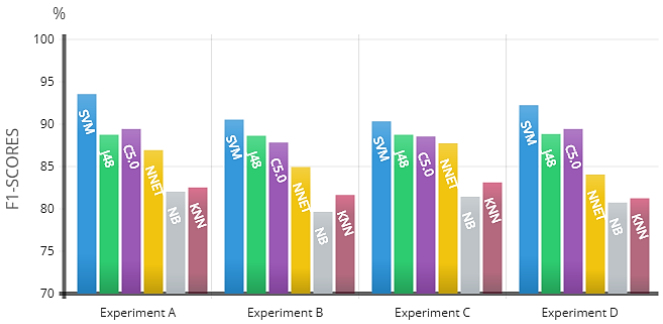

Figure 2- Classifiers F1-Scores

In a recent survey [2], SVM was reported by few papers to outperform other classifiers with MSA text, Figure 2 confirms SVM to be the most accurate method for the classification of Colloquial Arabic text as well.

Furthermore, we have applied McNemar's test on the test set of the SVM classifier for experiments A and B to determine if there is any significant difference between using light stemmer or work without stemming. Tweets from the test set that have been classified by SVM, have also been recorded for both experiments as shown in Table 4.

Table 4- classification results of classifiers

	Classifier B	
Classifier A	$n_{00} = 28$	$n_{01} = 32$
	$n_{10} = 12$	$n_{11} = 359$

Where n_{00} refers to the number of tweets misclassified by both classifiers, n_{01} is the number of tweets misclassified by classifier A but not by classifier B, n_{10} is the number of tweets misclassified by classifier B but not by classifier A and n_{11} is a number of tweets correctly classified by both classifiers. The null hypothesis (H_0) for this test suggests that classifiers A and B perform similarly whereas the alternative hypothesis (H_1) claims otherwise suggesting that the classifiers perform

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

differently, a low value of the calculated p-value (< 0.05) could be considered a significant result, rejecting the null Hypothesis.

McNemar's test calculate chi-square using equation listed below:

$$\chi = (|n_{01} - n_{10}| - 1)^2 / (n_{01} + n_{10})$$

χ is distributed approximately as χ^2 with 1 degree of freedom. For a 95% confidence test , $\chi^2_{1,095} = 3.84$. So if χ is larger than 3.84, then with 95% confidence, we can reject the null hypothesis. In other words, the two classifiers have the same error rate. From the above data, the McNemar's test statistic has the value of 8.2045 with 1 degree of freedom. The two-tailed P value equals 0.0042, by conventional criteria; the difference between classifiers performance is considered to be statistically significant. Hence, we reject the null hypothesis and accept that the two classifiers have significantly different performances.

The result of McNemar's test showed that the SVM classifier for experiment A (word without stemming) has performed significantly better results than SVM classifier for experiment B (light stemmer). This outcome contradicts previous studies such as [19] and [30] concluding that light stemming gives better results than words without stemming for Arabic tweets. However, we notice the dataset used by these two studies included a mixture of MSA texts and tweets with potential local Jordanian dialect which could explain the difference between both scenarios.

Stemming has been commonly used for the text classification problem in various languages. Recent research in this area shows the impact on the classification process has varied, it was therefore important to investigate the case for Colloquial Arabic. For instance, if we consider the Indonesian language, Hidayatullah et al. [31] performed an experiment on tweets and the results indicates that stemming does not consistently contribute to the accuracy for the SVM and the NB classifiers. In contrast, Basnur and Sensuse [32] used stemming with the NB classifier to categorise news articles written in Indonesian, and it was found that stemming helped to enhance the accuracy of this process, clearly the text in this scenario is long compared to tweets. However, experiments by Torunoğlu et al. [33] and Can et al. [34] utilising the SVM, NB and k-NN classifiers have shown conflicting result to the influence of stemming on Turkish text. Furthermore, in English, Aiello et al. [35] have compared six topic detection methods on three twitter datasets, their results show that the stemming phase reduces the performance of all examined methods. They refer the negative effect of words stemming to the fact that stemming partially disrupts word associations by merging too many words together. Another project by Zangerle and Specht [36] investigated how Twitter users react to having their account hacked and how they deal with compromised accounts, in their experiment it was reported that word stemming did not increase the performance for SVM. Nonetheless, Dovgopol and Nohelty [37] found that stemming has negligible impact on performance of the NB and k-NN classifiers, as a result of that they excluded the stemming phase in their final algorithm. Similar to our results on Colloquial Arabic, findings from experiments conducted on short English text (mainly tweets) agree that words stemming does not improve the performance of the mentioned classifiers when applied on informal text.

Furthermore, the Arabic script has numerous diacritics such as consonant pointing, or vowel marks which are supplementary discitis. They are used to facilitate pronunciation or to distinguish between a word and one of its homographs. In this study, we have applied a diacritics remover to normalize words for two reasons. Firstly, diacritics are very rarely used (if any) in colloquial Arabic; Habash [38] states that 98% of the Arabic text is written without diacritics. And secondly, because we have observed a similar finding in our own corpus.

V. CONCLUSION

In addition to investigating the effect of light stemming on Colloquial Arabic text for text classification, the main contribution of this paper was to investigate a variety of text classification techniques using Colloquial Arabic text as dataset. The classification techniques used in this paper have been widely used by many researchers to classify MSA text. However, to the best of our knowledge, none of the previous studies has tried to compare the performance of all these techniques when applied to datasets populated with Colloquial Arabic text, as presented in this paper. The classification algorithms tested in this study were: SVM, J48, C5.0, NNET, NB and k-NN. SVM produced the most accurate results. The next most noteworthy classification algorithms were Decision Tree (C5.0 and J48). For feature selection, we experimented with a number of conditions: without stemming, with light stemming, with removing common prefixes in the Arabic language, and finally with removing common suffixes. Our findings suggest that most classifiers perform better without applying stemming. Future work will continue to consider event detection and text classification based on Colloquial Arabic while taking into account metadata such as time and location.

REFERENCES

[1] E. Medvet and A. Bartoli, "Brand-related events detection, classification and summarization on twitter," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, 2012, pp. 297-302.

[2] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic text classification methods: Systematic literature review of primary studies," in *Information Science and Technology (CiSt), 2016 4th IEEE International Colloquium on*, 2016, pp. 361-367.

[3] Y. Kryvasheyeu, H. Chen, N. Obradovich, E. Moro, P. Van Hentenryck, J. Fowler, *et al.*, "Rapid assessment of disaster damage using social media activity," *Science advances,* vol. 2, p. e1500779, 2016.

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 851-860.

[5] Y. Al-Saggaf, "Social media and political participation in Saudi Arabia: The case of the 2009 floods in Jeddah," 2012.

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358

[6] M. Al-Saud, "Assessment of flood hazard of Jeddah area 2009, Saudi Arabia," *Journal of Water Resource and Protection,* vol. 2010, 2010.

[7] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation,* vol. 10, pp. 1895-1923, 1998.

[8] B. Bostanci and E. Bostanci, "An evaluation of classification algorithms using Mc Nemar's test," in *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, 2013, pp. 15-26.

[9] A. Singla, S. Patra, and L. Bruzzone, "A novel classification technique based on progressive transductive SVM learning," *Pattern Recognition Letters,* vol. 42, pp. 101-106, 2014.

[10] F. Huang, "Improved Arabic dialect classification with social media data," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2118-2126.

[11] L. Kaati, E. Omer, N. Prucha, and A. Shrestha, "Detecting Multipliers of Jihadism on Twitter," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 954-960.

[12] N. Alsaedi, P. Burnap, and O. Rana, "Sensing Real-World Events Using Arabic Twitter Posts," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[13] H. Becker, M. Naaman, and L. Gravano, "Beyond Trending Topics: Real-World Event Identification on Twitter," *ICWSM,* vol. 11, pp. 438-441, 2011.

[14] C.-C. Pan and P. Mitra, "Event detection with spatial latent Dirichlet allocation," in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, 2011, pp. 349-358.

[15] A. Al-Badarneh, E. Al-Shawakfa, B. Bani-Ismail, K. Al-Rababah, and S. Shatnawi, "The impact of indexing approaches on Arabic text classification," *Journal of Information Science,* p. 0165551515625030, 2016.

[16] A. Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, 2012, pp. 546-550.

[17] R. N. Al-Wehaibi and M. B. Khan, "Investigate the Context Usage of Arabic Proverbs in Twitter," in *Cloud Computing (ICCC), 2015 International Conference on*, 2015, pp. 1-8.

[18] W. e. Hadi, "Classification of Arabic Social Media Data," *Advances in Computational Sciences and Technology,* vol. 8, pp. 29-34, 2015.

[19] B. Brahimi, M. Touahria, and A. Tari, "Data and Text Mining Techniques for Classifying Arabic Tweet Polarity," *Journal of Digital Information Management,* vol. 14, p. 15, 2016.

[20] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," 1998.

[21] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence,* vol. 31, pp. 132-164, 2015.

[22] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 1873-1876.

[23] N. Alsaedi and P. Burnap, "Arabic event detection in social media," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2015, pp. 384-401.

[24] R. C. Team, "R: A language and environment for statistical computing," 2013.

[25] J. Gentry, M. J. Gentry, S. RSQLite, and R. L. Artistic, "Package 'twitteR'," ed, 2016.

[26] I. Feinerer, K. Hornik, and M. I. Feinerer, "Package 'tm'," *Corpus,* vol. 10, 2015.

[27] R. Nielsen. (19th Oct 2016). *Package 'arabicStemR'* Available: https://cran.r-project.org/web/packages/arabicStemR/arabicStemR.pdf

[28] M. Kuhn, "Caret package," *Journal of Statistical Software,* vol. 28, pp. 1-26, 2008.

[29] J. Paralic and P. Bednar, "Text mining for document annotation and ontology support," *Intelligent Systems at the Service of Mankind,* pp. 237-248, 2003.

[30] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on*, 2013, pp. 1-6.

[31] A. F. Hidayatullah, C. I. Ratnasari, and S. Wisnugroho, "Analysis of Stemming Influence on Indonesian Tweet Classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 14, pp. 665-673, 2016.

[32] P. W. Basnur and D. I. Sensuse, "Pengklasifikasian Otomatis Berbasis Ontologi Untuk Artikel Berita Berbahasa Indonesia," *MAKARA of Technology Series,* vol. 14, 2010.

[33] D. Torunoğlu, E. Çakirman, M. C. Ganiz, S. Akyokuş, and M. Z. Gürbüz, "Analysis of preprocessing methods on classification of Turkish texts," in *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, 2011, pp. 112-117.

[34] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vursavas, "Information retrieval on Turkish texts," *Journal of the American Society for Information Science and Technology,* vol. 59, pp. 407-421, 2008.

[35] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba*, et al.*, "Sensing trending topics in Twitter," *IEEE Transactions on Multimedia,* vol. 15, pp. 1268-1282, 2013.

[36] E. Zangerle and G. Specht, "Sorry, I was hacked: a classification of compromised twitter accounts," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, 2014, pp. 587-593.

[37] R. Dovgopol and M. Nohelty, "Twitter hash tag recommendation," *arXiv preprint arXiv:1502.00094,* 2015.

[38] N. Y. Habash, "Introduction to Arabic natural language processing," *Synthesis Lectures on Human Language Technologies,* vol. 3, pp. 1-187, 2010.

W. Alabbas, H. M. al-Khateeb, A. Mansour, G. Epiphaniou, I. Frommholz, "Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods", 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media), June 19-20, 2017, IEEE, London, UK. doi: 10.1109/SOCIALMEDIA.2017.8057358