Classification of Arabic-Speaking Website Pages with Unscrupulous Intentions and Questionable Language

Haya Mesfer Alshahrani
Lecturer in Information System Department
College of Computer Science and Information
Princess Nourah Bint Abdulrahman University

***Abstract*—This study aims to put forward a comprehensive and detailed classification system to categorize different Arabic-speaking website pages with unscrupulous intentions and questionable language. The methodology of this is based on a quantitative approach by using different algorithms (supervised) to build a model for data classification by using manually categorized information. The classification algorithm used to construct the model uses quantitative information extracted by Posit or SAFAR textual analysis framework. This model functions with (58) features combined from Posit – n-grams and morphological SAFAR V2 POS tools. This model achieved more than (94 %) success in the level of precision. The results of this study revealed that the best results reaching 94% precision have been achieved by combining Posit + SAFAR + (18 attributes Posit+ SAFAR N-Gram). Moreover, the most reliable results have been achieved by applying a Random Forest classification algorithm using regression. The research recommends working more on this topic and using new algorithms and techniques.**

Keywords*—Extremism; textual analysis; classification; Posit; SAFAR

I. Introduction

The last few years have seen a significant increase in the activities of radicalized extremists, launching terrorist attacks around the whole world. They exploit modern technologies such as the Internet and social media, widely used by the public, to plan and maintain contact with their group [1]. Social media like Facebook and Twitter are currently being utilized by extremist groups to create direct contact with their worldwide groups. By the very nature of these applications (i.e., free and unregulated), encouraged extremists to quickly form virtual societies and disseminate their thoughts and their coaching tools without paying attention to the usual means of censorship in the general media [1].

Therefore, social networks have started to intervene by implementing countermeasures against these groups. Twitter was considered the main promotional vehicle for ISIS, so in August 2016, Twitter started taking more stringent measures by closing more than 36,000 feeds that were believed to belong to ISIS [1, 2].

Fundamentally, the benefits of using data collected from social media depend on the factual accuracy of the statements being collected from the users or their groups [3]. However, it was established that additional effective procedures such as utilizing algorithms to uncover clues in the content that points to violence automatically supported this feedback and improved its performance [4]. Notwithstanding, the feedback resulted in social websites closing a significant number of accounts, however, it was not guaranteed to be accurate, because owners of the pending accounts can create new accounts and resume their activities, or are able to relocate to different social websites. More research needs to be carried out by governments in order to counter the radicalized extremists and stop, or at least reduce their threat [1].

The study seeks to establish a comprehensive system to reveal any publication or entity that has malicious intentions emanating from extremism or seeks terrorism, and that is in various Arabic web pages. Through this study, an individually combined corpus of (5,100) text files, and more than (1,000,000) Arabic words were built. A new enhanced POS (part of speech) from the Posit tool developed by Weir (2007, 2009) was introduced with modifications on the code to deal with Arabic content. Finally, through this study, a classification model that functions with (58) features combined from Posit – n-grams and morphological SAFAR V2 POS tools was developed where this model achieved more than (99 %) success in the level of precision.

This study is divided into an introduction, a literature review of previous work and Arabic morphological classification. Then, the methodology used in this study will be presented beginning with data preparation, compiling, and building of the Arabic corpus, and covering the classification methodology. Part four explains the implementation of this methodology detailing the code used to split the massive corpus into individual text files. Information preparation began by extracting quantitative information from Arabic corpus using Posit and SAFAR frameworks. Part five discusses the experiments and the setup required - starting with the software and ending with WEKA and how it was edited to fit with our approach with the details of POS and SAVAR and the attribute of N-gram. Finally, we tackled the classification results, analysis of the eight hypotheses that have been put forward.

II. Related Literature

A. Sentiment Analysis of Arabic Text (Opinion Mining)

Aldayel and Azmi (2016) carried out a study on sentiment analysis that connected various domains of study such as NLP, computational linguistics, and text mining [5]. It concerns the extraction of the given information from textual data. It may be called sentiment analysis or opinion mining as Pang and

Lee (2008) used the Twitter API to collect Twitter data from a specific domain in a specific language [6]. Preprocessing was done by the removal of irrelevant information, tweet cleaning and other preprocessing techniques. The classification technique is based on a Lexicon-based classifier. To extract features used in the classification process, they used the term frequency inverse document frequency (TF-IDF) weighting scheme on the n-grams (1-2-3 gram) and selected the features that have frequencies greater than a certain threshold. They used two measures to evaluate the classification process; namely (The error rate (percentage of misclassification twists and the accuracy Rate (percentage of correctly classified twists) [6].

The Twitter API for Arabic data collection was used. The data was then passed through data cleaning and attribute extraction using 1-2-gram statistical processing. This is to prepare the data to obtain the feature vector for the main purpose of research, i.e., classification. The machine learning classifiers used are Naive Bayes (NB), and Support Vector Machines (SVM). They apply both classifiers twice. First, they apply both classifiers on features extracted based on unigrams. Then, use the features extracted based on bigram statistics [7].

The SVM classifier was employed as the research classifier and the data collection used the Twitter API. Data cleaning and normalizing, with stemming, and stop words removed were applied to make data suitable for feature extraction. The data sets were organized using 1- Unigrams, 2- Bigrams + Unigrams and 3- Unigrams + Bigrams + Trigrams [7].

The SVM classifier was applied before after applying each stage of the preprocessing to test its effect on the system's performance. Sentiment analysis studies vary in pre-treatment techniques, analysis methods, and review design. Some have used the supervised method, others the unsupervised learning method. A multi-level technique based on semantic orientation (lexical classifier to handling unnamed tweets) and ML (SVM classifier) was suggested by Aldayel and Azmi (2016) to identify the polarity of Arabic tweets. The biggest challenge of this mixed approach, however, is to deal with the application of Twitter in dialectical Arabic [5].

Moraes, Valiati and Neto (2013) compared the execution of SVM (support vector machines) and NN (neural networks) at document-level sentimental Arabic analysis. They have found that NN execution is better than SVM on the same records [8].

Li and Li (2013) have gauged the objectivity and the truthfulness by utilizing SVM as a method [9]. Cherif, Madani and Kissi (2015) worked on the execution of three famous techniques (bagging, boosting and random subspace). This was instituted on five algorithms, which are (Naive Bayes, Maximum Entropy, Decision Tree, K Nearest Neighbor, and Support Vector Machines) for sentiment categorization. The results showed that the random subspace was more accurate [10].

Duwairi and Qarqaz (2014) studied the effects of stemming feature correlation and n-gram models for Arabic text on sentiment analysis. They used Support Vector Machines, Naive Bayes, and K-nearest neighbor classifiers, while the results of the experiments suggested that choosing the method of preprocessing on the reviews would enhance the performance of the classifiers [11].

B. Classification and Comparing Algorithms on Arabic Text

El Kourdi Bensaid and Rachidi (2004) categorized Arabic documents on the internet automatically by using an NB classifier with ML algorithms to classify soundless Arabic documents into one of five pre-determined classes. The results of the experiments confirmed the effectiveness of the NB classifier. El Koudri utilized groups of 1500 documents under five categories each with 300 text documents. Through 2000 expressions and roots, the precision of the classification varies in-between categories with an average precision overall for the classifiers of 68.78 %. Moreover, the highest performance of categories in these experiments reached 92.8% [12].

KNN algorithm (K-Nearest Neighbor) is one of the best classifiers for categorizing text documents in English with the SVMs algorithm. This was used by Al-Shalabi Kanaan and Gharaibeh (2006) on the Arabic language for text classification. They utilized the DF (Document Frequency) technique to extract the main words and minimize dimensions. The results proved that the KNN is suitable to categorize Arabic documents [13].

Maximum Entropy (ME) was applied by El-Halees (2015) and Sawaf, Zaplo and Ney (2001) to categorize Arabic news articles. El-Halees pre-processes data, utilizing natural language processing methods such as tokenizing, stemming, and part of speech then uses the maximum entropy method to categorize Arabic documents. The best-reported accuracy was 80.41% and 62.7% when using statistical methods by Sawaf without morphological analysis [14, 15].

Al-Zoghby, Eldin, Ismail and Hamza (2007) proposed a novel system that was developed to determine association rules using similarity measurements based on the derivation of the Arabic language. It also offered the advantage of using the "Frequent Closed Item sets" (FCI) concept when extracting the association rules instead of "Frequent Item sets" (FI) [16].

III. METHODOLOGY

The methodology of this is based on a quantitative approach by using different algorithms (supervised) to build a model for data classification by using manually categorized information. Through this study, a 'seed list' of Arabic words and sentences was used in an input list box and the sketch engine that fetched around one million words per search. The data range was the most likely used words for extremism websites, tweets and any social media website, e.g., the Arabic equivalent of " kill the disbeliever and you enter heaven". Through this process, more than 7000 Arabic text files were collected and processed to form the downloaded corpus of individual files, in which everyone represents pro-extreme text, with associated id and URL. The same approach was followed for Anti-extremism and neutral data.

The research has been divided into several stages, and in each stage, certain methods and tools have been deployed. The following sections will explain these methods and equipment.

A. Data Collection and Preprocessing Stage

The proposed system depends on the analysis of an Arabic dataset but a specific one; it must contain text data for encouraging extremism, anti-extremism and neutral data, in Arabic language. Since such an existing resource proved elusive, we had to develop our own means of gathering such a dataset (using tools like sketch engine). To this end, we used a 'seed list' of Arabic words and sentences in an input list box and the sketch engine will fetch around one million words per search. The data range was the most likely used words for extremism websites, tweets and any social media website, e.g., the Arabic equivalent of " kill the disbeliever and you enter heaven".

Through this process, we collected more than 7000 Arabic text files and processed them to form the downloaded corpus of individual files, in which everyone represents pro-extreme text, with associated id and URL. The same approach was followed for Anti-extremism and neutral data.

The data collection was the first step in this study where it was based on a mixture of locations that considered being antiterrorism, pro-terrorism, and neutral sites to ensure balanced datasets for training and test datasets. Then, preprocessing was performed for the data that include but are not limited to removing non-Arabic text; removing HTML tags; excluding empty files; splitting pages of websites, and adding file ID to each file.

B. Data Analysis Stage

The step following data preprocessing is to apply text analysis toolkits to derive detailed information on the Arabic file content. The result of this process is to generate summary files containing all numeric, quantitative information about the Arabic text files. In addition, an N-Gram file is created to be used for the classification process and prediction calculations.

The main competition among the different data processing tools available lies in the number of distinct features that can be extracted from Arabic text. The more features, the more quantitative information, and, potentially, the more precise will be the classification. We should note the need for an Arabic language expert working side-by-side with the developer to review and audit the results coming out of each tool, to make sure they are semantically correct.

The main operations on the Arabic text should include the following:

- Stem counting,
- Sentence Processing
- Morpho-Syntactic Processing,
- Summarizing,
- Arabic Parsing, and
- Morphological analyzing.

We used the Posit and SAFAR tools, which gave more than 58 features together, to create Summary files. Once we get summary and N-Gram files we are ready for data classification. The classification process is divided into two main steps; the first step is to ensure that the training data set, which is manually classified, will produce a high-quality model for future use. Then, in the second step, we can test and create the model.

C. Data Classification Stage

The next step is to use the model file that is created in the first step for classification of the unseen dataset to calculate a prediction for each file individually. To calculate prediction and to construct a confusion matrix, we store the extracted quantitative data as well as N-Gram data in a suitable format for training and test datasets. The classification is done by studying the attribute parameters during the training phase, and then considers the hidden files in order to predict the class for each new data item. The classification process can be divided into two main steps.

To perform the classification process for the unseen dataset, we follow these detailed steps:

- Divide the collected corpus into a training dataset and test (unseen) dataset.
- Put all information collected that relates to each file into a suitable format for the classification process (ARFF file format).
- Manually classify the data samples by a high-qualified person for the training dataset. The purpose of the training dataset is to create a classification model used subsequently in the classification of the unseen dataset.
- Examine the training dataset for the quality of classification. We divide the data into 70% and 30% subsets. We use 70% of datasets as a self-training dataset and 30% as self-test datasets.
- Explore the use of different algorithms for classification. To choose the most suitable classification algorithm, we study many classifiers that can be listed under different classification concepts. The results are not selected based on the classifier only but also depend on dataset combinations of the two text analysis toolkits and the use of N-Grams generated by both text analysis toolkits.
- Select the best combination of dataset and classifier, based upon the precision, Recall, and F-Measure.
- We selected WEKA (machine learning environment) as a basis for our classification work because it is rich with a classification environment with attribute processing like attribute selection and a rich library of machine learning algorithms. Moreover, it has an API to be used to throw user-made applications.
- The programming language selected for creating the user interface is JAVA. It can utilize WEKA API to produce an efficient application that can fulfill all research requirements, including classification, and put the results in a suitable form for analysis.

D. Research Tools

In this study, three tools were employed including Posit Toolset that contains POS Profiler, Vocabulary Profiler and Readability Profiler. The second tool was SAFAR (Software Architecture For Arabic language processing) program, which was used in the stage of data processing in the proposed system, as it worked on extracting quantitative information from text file data. Finally, WEKA API was used in the proposed system in order to classify the processed text data into three types (terrorist, anti-terrorist, and neutral).

E. Sample

WEKA API classifier needs training in a pre-categorized dataset to learn how to differentiate between our three categories (pro-terrorism, anti-terrorism, neutral) and the distinct features and words for each category. In this system, a train data set of 300 files of textual data containing the three categories were used to train the classifier, and then this was tested to check its accuracy and effectiveness.

F. Validity and Reliability

Validity and reliability were taken into consideration in each step of the system design and implementation. First, collecting data was done by using several different sources (neutral sources, sources supporting terrorism, and anti-terrorism sources), whether it is on the Internet, social media, and elsewhere.

The manual data classification stage was performed by five different specialized people. Once classified in a category by at least four people the file is considered classified and added to the Corpus, otherwise, it is removed from the Corpus group. In the next stage, a program was used to process the Arabic texts.

The next stage was to train the classifier through a group consisting of 300 text files of the three types. Finally, to ensure the validity of the training, progress and design of the program, the program was tested through a test dataset consisting of 200 text files of the three categories. The 200 files were completely correctly categorized, which demonstrated the validity of the proposed program's work for this message.

IV. Results and Discussion

In this test, the used ARFF file was created depending on summary files resulting from each of the data sets dataset. Each classification process requires training and test datasets. Fig. 1 shows test classification results obtained by applying desired classifiers on each dataset.

A. Test Results

A review of unseen dataset test only to summarize the results for useful information.

1) Final posit datasets discussion: Table I shows Posit dataset classification best results.

By adding the N-Gram attributes to the Posit attribute, the Random Forest classifier dominated over other classifiers with considerable value (about 5%) for the supervised dataset.

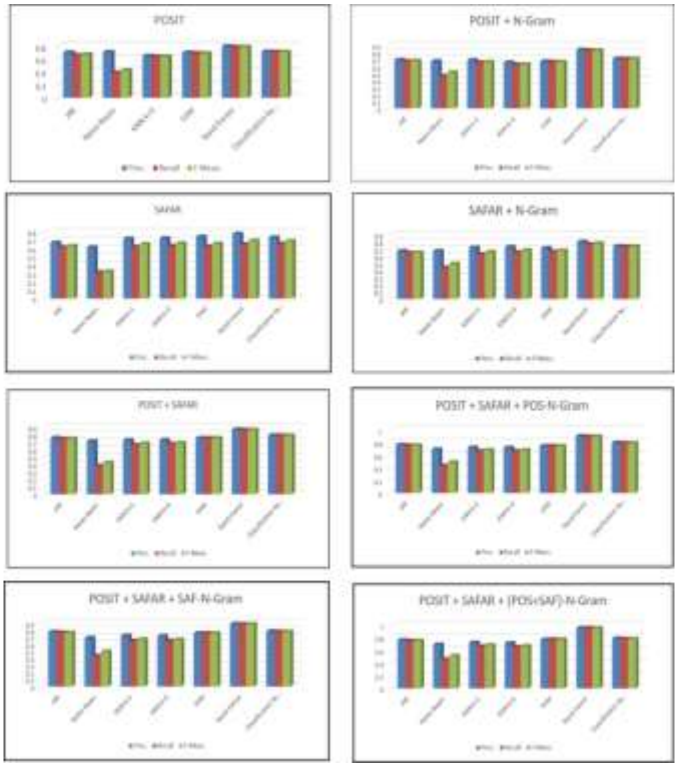

Fig. 1. Shows Test Classification Results Obtained by Applying Desired Classifiers on each Dataset.

TABLE I. Posit Classification Results

Supervised				
		Prec.	**Recall**	**F-Meas.**
Posit	**Rand-Forest**	0.797	0.785	0.786
Posit+ N-Gram	**Rand-Forest**	0.840	0.831	0.832

2) Final SAFAR dataset discussion: Table II shows Posit dataset classification best results.

TABLE II. SAFAR Classification Results

Supervised				
		Prec.	**Recall**	**F-Meas.**
SAFAR	**Rand-Forest**	0.778	0.654	0.696
SAFAR + N-Gram	**Rand-Forest**	0.805	0.771	0.783

The above results show that less precision of 80% for SAFAR + N-Gram dataset. SAFAR dataset without N-Gram attribute offers lower precision by about 3%.

3) Posit + SAFAR dataset discussion: Table III shows Posit + SAFAR datasets classification results.

Results show the minimum precision of 87.2% for (Posit+SAFAR) dataset and best precision by using the Posit+SAFAR + (Posit+SAFAR N-Gram) dataset of value 95.4%.

TABLE III. SHOWS POSIT + SAFAR DATASETS CLASSIFICATION RESULTS

Supervised				
	Classifier	Prec.	Recall	F-Meas.
Posit+SAFAR	Rand-Forest	0.872	0.864	0.867
Posit+SAFAR + (Posit N-Gram)	Rand-Forest	0.904	0.894	0.896
Posit+SAFAR + (SAFAR N-Gram)	Rand-Forest	0.890	0.886	0.887
Posit+SAFAR + (Posit+SAFAR N-Gram)	Rand-Forest	0.954	0.953	0.952

B. Final Results

Here the best results from all tests and achieved the best performance (Precision 95%, Recall = 95%, F-Measure = 95%) by applying Random Forest classifier on (Posit+SAFAR) + (Posit+SAFAR) N-Gram. Table IV shows the classifier results sorted in ascending order of performance.

C. Discussion

Many researchers have tried to obtain the optimum classification algorithm for different languages, especially Arabic. The common toolkit was set up, Posit, to work for the Arabic language. This helped enhance the overall process by finding more than one toolkit to extract meaningful quantum information from Arabic text. The use of N-Gram was another way to process the amount of information used to learn the parameters of attributes and to calculate the prediction of unseen data. WEKA data processing environment is a rich environment for the Random Forest classification algorithm.

1) Comparison: A similar approach to the classification task is reported in the following sections:

Aldayel and Azmi (2016) used a Hybrid approach classifier Algorithm for the Arabic language. The approach is based on using a lexical classifier for training data for the SVM classifier, Lexical classifier. Used for first step classification to produce the training dataset for model creation. Dataset: 1103 tweets (576 positives, 527 negatives). Then the SVM classifier was used for the classification of an unclassified dataset. The hybrid classifier (Lexical + SVM) produce results as follows Table V:

This research uses a lexical classifier for learning datasets rather than manual classification to apply the SVM classifier to classify the unseen tweets dataset. This combination enhances the overall operation.

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN}$$

Shoukry and Rafea (2012) worked to process tweets to provide their sentiments polarity (positive or negative). SVM and Naïve Bayes (NB) used for both training and classification, one by one. Dataset: 1000 tweets (500 positives, 500 negatives). Results obtained were as follows Table VI:

This research use sentiments classification to produce a learning dataset and supervised test for unseen tweets dataset using two different classification algorithms. SVM gave better results over Naïve Bayes by 7.4%.

TABLE IV. ALL DATASETS CLASSIFICATION SORTED RESULTS

Supervised				
	Classifier	Prec.	Recall	F-Meas.
(Posit+SAFAR) +(Posit+SAFAR) N-Gram	Rand-Forest	0.95	0.95	0.95
Posit + SAFAR + (Posit 2-3-4 Gram)	Rand-Forest	0.90	0.89	0.9
Posit + SAFAR + (SAFAR 2-3-4 Gram)	Rand-Forest	0.89	0.89	0.89
Posit + SAFAR	Rand-Forest	0.87	0.86	0.87
Posit + N-Gram	Rand-Forest	0.84	0.83	0.83
SAFAR + N-Gram	Rand-Forest	0.81	0.77	0.78
Posit	Rand-Forest	0.80	0.79	0.79
SAFAR	Rand-Forest	0.78	0.65	0.7

TABLE V. TWEET CLASSIFICATION

		Prec.	Recall	Accuracy
Tweet datasets	Lexical + SVM	0.847	0. 838	0. 840

TABLE VI. SENTENCE LEVEL ARABIC SENTIMENT (1-2 GRAMS) SVM AND NAÏVE BAYES

		Prec.	Recall	F-Meas.	Accuracy
Tweet datasets (Unigrams and Bigrams)	SVM	0.726	0.728	0.726	0.725
Tweet datasets (Unigrams and Bigrams)	Naïve Bayes	0.652	0.652	0.652	0.652

Its Arabic sentiment considers normalization, stemming, and stop word removal for datasets (during the preprocessing phase) (Shoukry & Rafea, 2012). SVM is used for both training and classification. Dataset: 1000 tweets (500 positives, 500 negatives). The results obtained are as follows in Table VII.

TABLE VII. SENTENCE LEVEL ARABIC SENTIMENT (2-3 GRAMS) SVM (NORMALIZATION, STEMMING, AND STOP WORDS REMOVAL)

Supervised					
Dataset	Classifier	Prec.	Recall	F-Meas.	Accuracy
Unigrams - raw tweets	SVM	0.740	0.740	0.740	0.740
Unigrams – normalized tweets	SVM	0.756	0.756	0.756	0.756
Unigrams + Bigrams + Trigrams stemmed tweets	SVM	0.787	0.787	0.787	0.787
Unigrams + Bigrams + Trigrams after stop words removal	SVM	0.788	0.788	0.788	0.788

Adding 2-3-4 Gram information enriches the training and test datasets. Preprocessing enhances the classification process by different factors.

El-Halees (2015) used a combined approach for Arabic language classification in the beginning; the lexicon-based method is used to classify as many documents as possible. The resultant classified documents are used as a training set for the maximum entropy method, which subsequently classifies some other documents. Finally, the k-nearest method used the classified documents from the lexicon-based method and maximum entropy as a training set and classified the rest of the documents.

Dataset: 949 tweets (415 positives, 534 negatives) belong to "education", "politics" and "sports" categories. Results collected as follows in Table VIII.

A combined classification (Lexical + Maximum Entropy + k-nearest) approach enhances classifier accuracy. Observing the last 4 types of research, the researcher is going forward for the Arabic classification process, which is considered to be an NP-complete problem (nondeterministic polynomial time) [https://www.ics.uci.edu/~eppstein/161/960312.html].

The overview shows that the researcher's results reach an acceptably high level of precision by using different ways of data preprocessing thereby enriching the input data by adding N-Gram or classifying by multiple classifiers. In the following Table IX, a comparison of results to reviewed researches results.

Our approach depends on manual classification for the training dataset (70% + 30% seen dataset) to ensure the best results. Note that the process of manual classification is time consuming, especially if it is carried out on several thousand-text files. This is also influenced by the scientific level and culture of those involved in the process of manual classification. After manual classification, the text-processing toolkit was applied in order to build datasets for the training and classification process. Attribute data is extracted by two different toolkits (Posit & SAFAR), which build information obtained from text files.

2) Unseen datasets: The Random Forest algorithm used for creating a classification model employing a carefully and manually classified dataset gives us the best results over other classification techniques, as in Table X.

TABLE VIII. COMBINED APPROACH FOR ARABIC LANGUAGE CLASSIFICATION

	Lexical	**Lexical + ME**	**Lexical +ME + kNN**	
Accuracy	50.08	60.73	80.29	
		Prec.	Recall	F-Meas.
Politics Sports Education datasets	**Lexical + Maximum Entropy + k-nearest**	80.7	79.805	79.895

TABLE IX. COMPARING OTHER CLASSIFICATION RESULTS TO OUR CLASSIFICATION METHODOLOGY

	Dataset	Classifier	Prec.	Recall	F-Meas.	Acc.
1	**Tweet datasets**	**Lexical + SVM**	0.847	0. 838		0. 840
2	**Tweet datasets (Unigrams and Bigrams)**	**SVM**	0.726	0.728	0.726	0.725
	Tweet datasets (Unigrams and Bigrams)	**Naïve Bayes**	0.652	0.652	0.652	0.652
3	**Unigrams - raw tweets**	**SVM**	0.740	0.740	0.740	0.740
	Unigrams – normalized tweets	**SVM**	0.756	0.756	0.756	0.756
	Unigrams + Bigrams + Trigrams stemmed tweets	**SVM**	0.787	0.787	0.787	0.787
	Unigrams + Bigrams + Trigrams after stop words removal	**SVM**	0.788	0.788	0.788	0.788
4	**Politics Sports Education datasets**	**Lexical + Maximum Entropy + k-nearest**	80.7	79.805	79.895	
5	**(POSIT+SAFAR) + (POSIT+SAFAR) N-Gram**	**Rand-Forest**	0.95	0.95	0.95	

TABLE X. RANDOM FOREST CLASSIFIER AGAINST SOME OTHER CLASSIFIERS

Algorithm	**Dataset**	**Prec.**	**Recall**	**F-Meas.**
Random Forest	**Posit + SAFAR + (Posit + SAFAR N-Gram)**	**0.95**	**0.953**	**0.952**
RF Via Regression	Posit+ SAFAR + (Posit N-Gram)	0.80	0.79	0.791
J48	Posit+ SAFAR + (SAFAR N-Gram)	0.78	0.768	0.766
SVM	Posit + SAFAR + (Posit + SAFAR N-Gram)	0.77	0.771	0.771
IBk_3	SAFAR + N-Gram	0.73	0.657	0.684
Naïve Bayes	Posit + SAFAR	0.71	0.38	0.42

The Random Forest algorithm gave the best result against the manually classified dataset (Posit + SAFAR toolkits) and other algorithms with a precision of 0.95. Other algorithms (RF via Regression, J48, SVM, IBk_3, Naïve Bayes) show good results with different datasets, all undergoing manual classification with results of (0.71-0.80).

V. Conclusion

This study showed that the classification process can be improved and automated using Random Forest and Random Forest via the Regression classification algorithm, which is integrated into JAVA application using the WEKA machine-learning environment (WEKA API). The used datasets for unseen data classification are different combinations of data extracted using Posit and SAFAR toolkits and N-Gram attributes.

Items in text writing (for example, word or phrase) can be labeled under various tags (Pro extremist – Anti extremist – neutral). This makes it hard to distinguish between different classes of context using the automated classification system. The nature of the text makes it difficult to reach the maximum prediction that is equal to one, but it reduces as much as the uncertainty of determining the item class exists. Moreover, it has been shown from our practical experience that combining different attributes deduced by combined two toolkits for analyzing Arabic text can be used to enhance text categorization using a sufficient set of carefully manually classified files.

This study recommends conducting further studies that are based on the increasing diversity of a collection of data from different site categories (sports – politics – social – food – health, etc.) to get alternative ways of writing and to overcome the lack of sites supporting terrorism. Furthermore, it recommends finding an algorithm that can use in conjunction with manual sorting to reduce the effort and time required for manual classification. Using techniques like attribute selection will have better performance especially with datasets with larger n-gram data. Finally, other future work is to automate the classification process using our produced model and other models for multi-language websites including social media sites, and to propose accepted datasets enhancing the model by re-training to produce a new model file for future use.

References

[1] H. Alvari, S. Sarkar, P. Shakarian, Detection of violent extremists in social media, in: 2019 2nd International Conference on Data Intelligence and Security (ICDIS), IEEE, 2019, pp. 43-47.

[2] T.C. Helmus, E. Bodine-Baron, Empowering ISIS Opponents on Twitter, RAND Corporation, 2017.

[3] J. Peral, A. Maté, M. Marco, Application of data mining techniques to identify relevant key performance indicators, Computer Standards & Interfaces, 54 (2017) 76-85.

[4] R.K. Bakshi, N. Kaur, R. Kaur, G. Kaur, Opinion mining and sentiment analysis, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2016, pp. 452-455.

[5] H.K. Aldayel, A.M. Azmi, Arabic tweets sentiment analysis–a hybrid scheme, Journal of Information Science, 42 (2016) 782-797.

[6] B. Pang, L. Lee, Opinion mining and sentiment analysis, Comput. Linguist, 35 (2009) 311-312.

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

[8] R. Moraes, J.F. Valiati, W.P.G. Neto, Document-level sentiment classification: An empirical comparison between SVM and ANN, Expert Systems with Applications, 40 (2013) 621-633.

[9] Y.-M. Li, T.-Y. Li, Deriving market intelligence from microblogs, Decision Support Systems, 55 (2013) 206-217.

[10] W. Cherif, A. Madani, M. Kissi, A new modeling approach for Arabic opinion mining recognition, in: 2015 Intelligent Systems and Computer Vision (ISCV), IEEE, 2015, pp. 1-6.

[11] R.M. Duwairi, I. Qarqaz, Arabic sentiment analysis using supervised classification, in: 2014 International Conference on Future Internet of Things and Cloud, IEEE, 2014, pp. 579-583.

[12] M. El Kourdi, A. Bensaid, T.-e. Rachidi, Automatic Arabic document categorization based on the Naïve Bayes algorithm, in: proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, 2004, pp. 51-58.

[13] R. Al-Shalabi, G. Kanaan, M. Gharaibeh, Arabic text categorization using KNN algorithm, in: Proceedings of The 4th International Multiconference on Computer Science and Information Technology, 2006, pp. 5-7.

[14] A.M. El-Halees, Arabic text classification using maximum entropy, IUG Journal of Natural Studies, 15 (2015).

[15] H. Sawaf, J. Zaplo, H. Ney, Statistical classification methods for Arabic news articles, Natural Language Processing in ACL2001, Toulouse, France, (2001).

[16] A. Al-Zoghby, A.S. Eldin, N.A. Ismail, T. Hamza, Mining Arabic text using soft-matching association rules, in: 2007 International Conference on Computer Engineering & Systems, IEEE, 2007, pp. 421-426.