Offensive Tweets Classification

1st Mohammad Muwafi
Electrical and Computer Engineering
Birzeit University
Student ID: 1180491

2nd Ahmad Hamad
Electrical and Computer Engineering
Birzeit University
Student ID: 1180060

3rd Baraa Atta
Electrical and Computer Engineering
Birzeit University
Student ID: 1180445

***Abstract*—In the last decade, the social media becomes include enormous number of users, therefore, the offensive speech is becoming apparent repeatedly. And since the AI research papers always focus on the English content, we decided to make a project to detect the Arabic offensive tweets as the twitter is one of the biggest websites that includes Arabs from different cultures, countries and religions. A dataset of 5800 tweets was combined for our project that contains informal Arabic language, and it was trained using classical machine learning algorithms: DT, RF, NB, SVC and KNN as well as deep learning algorithm such as ANN.**

***Index Terms*—NLP, AI, Offensive, TfIdf**

I. Introduction

A. NLP and IR Roles With Textual Data

Twitter is one of the biggest social media websites which has billions of users who can tweet, reply, like the tweets and many more options, so a very huge amount of data is created daily by all of these users, and very big ratio of this data is a textual data, therefore, NLP and IR techniques are playing very important role to manipulate this data and analyse it.

B. Machine Learning

Fortunately, machine learning field is strongly connected with NLP and IR concepts, so the text data can be very useful if we try to merge the AI and NLP concepts to build a model that can solve a specific task that related to textual users' data. In this project, we have used the supervised classical machine learning and deep learning with a dataset that contains about 5800 tweets of informal Arabic language used to build a model that can detect weather if the tweet is offensive or not offensive.

C. Offensive Tweet Detection

Detection of offensive tweets has become one of the major applications on machine learning, due to the wide range of the online tweets about almost all the different topics. Using tweets property on the twitter, bad people are trying to write offensive tweets about any topic depending on their goals. For-example some of them write offensive tweets in-order to increase the general replies for a specific post or tweet, while others may be totally thinking the opposite to harm someone. Due to all the mentioned reasons, we observe that tweets can affect the general view of the specific things, and hence increasing the harassment, fights, discords and collisions... etc. From here and since the science seeks to enhance the positive side of technology and development, there are many models that tried to make an offensive tweet detector. But, the problem that the weakness of the Arabic content. So, we decided to make an Arabic offensive tweet detector that uses the different concepts of NLP and AI techniques.

D. Scikit Learn and NLTK Library

The Sklearn and NLTK libraries are libraries which contains many of machine learning tools for cleaning, balancing, analysing, pre-processing, classification and clustering the data. And since it implemented on python, that means the periodically support and rich content are available because of good community and that lead us to develop our work easily in the future.

E. React And Flask Frameworks

Since we need to make an interactive project which has a user-friendly GUI, the React framework was the best choice which leads us to build a nice GUI, while the back-end server works using Flask framework, therefore a part of the work and time were spent to learn these technologies. However, with this good effort, we can deploy our project to let the people use it as usual websites.

II. Mythology

In this section, we will explain all the operations that applied on the dataset that help us to extract the features from the textual data.

A. Pre-processing

As we learned in the NLP course, there are many different processes that can be applied to the textual dataset such as: tokenization, stop-words removal, stemming, finally feature extraction.

1) Tokenization: it is the process of separating the sentences into words that called tokens separated by white space, these tokens will be stemmed later. However, we used the NLTK library to get the Arabic stop-words set.

2) Stop-words removal: a set of Arabic stop-words is used so that if any token of tweets' tokens was inside the set of the Arabic stop-words, then it will be removed.

3) Stemming: stemming is the process of deleting all of suffixes and prefixes of the word such that convert the word into its root. Of course, many words have the same root, so the total number of tokens will be decreased after the stemming operation. we used ISRI stemmer to stem the tweets, and the following steps are applied to the tokens by ISRI stemmer:

- Removes diacritics.
- Normalization of Hamza.
- Delete two-letter and three-letter prefixes in the word sequence.
- Delete connecting letters "waw" if there is a letter "waw" in the prefix of the word.
- Normalization of "Alif".
- Stemmer returns the same letter if the word 3 letters. And return the same word if the word is ambiguous.

B. Feature Extraction

The classification models need an appropriate input data such that the textual data (tweets) have to be converted to numerical data, the input of the classifier should be chosen correctly, and transformed using a good feature vector and it's important to know that the better feature vector the better results of the models.

we used both of the TF-IDF vectorizer and the Count vectorizer in our work. In TF-IDF vectorizer, TF stands for term frequency while IDF stands for inverse document frequency and these are the formules:

$$TF = n_t/n$$

$$IDF = log_2(N/N_t)$$

$$TF - IDF = TF * IDF$$

Where:

- nt means number of time that term t appears in document.
- n means number of terms appear in document.
- N means number of documents.
- Nt means number of documents that contains term t.

As a note, some terms were mentioned many times in the tweets and maybe they can be considered as a stop words but sadly did not removed using stop-words removal process so in TF-IDF vectorizer this issue can be solved since IDF will take that into account such that it gives a small value for those words which seems to be stop-words.

On the other hand, in the Count vectorizer that called BOW (Bag of words), the order of the words is not important while the only important thing is how many times the term t occurs in the document.

C. Training Models

As we said previously, the dataset has been preproccessed and converted to numerical data in both of the preproccessing and the feature extraction steps. So, the data now is ready to enter the classifiers to be trained. However, this project was implemented to classify the tweet whether if it is offensive or not offensive and since the dataset was labeled, then the classical machine learning algorithms or deep learning can be used for our target. So, we have used 6 different classifiers as following:

- Decision Tree Classifier.
- Random Forest Classifier.
- Naive Bayes Classifier.
- Support Vector Machine Classifier.
- K-Nearest Neighbors Classifier.
- Artificial Neural Network Classifier.

The above 6 classifiers were trained as the following combination criteria:

- using TF-IDF vectorizer with stemming.
- using TF-IDF vectorizer without stemming.
- using Count vectorizer with stemming.
- using Count vectorizer without stemming.

And of course, the data was splitted in such a way that 20% for testing and 80% for training the model.

III. Results and Testing

As we said before, the tweets were converted to numerical form in the feature extraction process and trained using 6 different classifiers as shown in the following tables.

Classifier	Precision	Recall	F1-measure	Accuracy
Decision Tree	87.2%	81.4%	84.2%	79.4%
Radnom Forest	95.1%	79.6%	86.7%	81.6%
Naive Bayes	46.1%	77.6%	57.8%	57.7%
KNN	91.7%	78.6%	84.6%	79%
SVM	94%	82%	87.8%	83.5%
ANN	81.8%	83.3%	82.5%	78.2%

TABLE I: using TF-IDF and with stemming

Classifier	Precision	Recall	F1-measure	Accuracy
Decision Tree	89.3%	78.1%	83.3%	77.5%
Radnom Forest	97.1%	76.4%	85.5%	79.3%
Naive Bayes	63.3%	81%	71.2%	67.6%
KNN	93.3%	80%	86.2%	81.2%
SVM	96.2%	80.6%	87.7%	83.6%
ANN	90.2%	82.9%	86.4%	82.1%

TABLE II: using TF-IDF and without stemming

Classifier	Precision	Recall	F1-measure	Accuracy
Decision Tree	91.2%	79.8%	85%	79.9%
Radnom Forest	96.6%	76.9%	85.7%	79.6%
Naive Bayes	60.7%	85.1%	70.9%	68.6%
KNN	86.5%	70.6%	77.8%	68.9%
SVCM	89.9%	81.5%	85.5%	80.8%
ANN	91.1%	78.9%	84.6%	79.1%

TABLE III: using BOW and with stemming

Classifier	Precision	Recall	F1-measure	Accuracy
Decision Tree	87.2%	81.4%	84.2%	79.4%
Radnom Forest	95.1%	79.6%	86.7%	81.6%
Naive Bayes	46.1%	77.6%	57.8%	57.7%
KNN	91.7%	78.6%	84.6%	79.0%
SVM	94%	82%	87.8%	83.5%
ANN	81.8%	83.3%	82.5%	78.2%

TABLE IV: using BOW and without stemming

As we see in the above tables, in general, all classifiers did good especially Support Vector Machine classifier while Naive Bayes classifier did bad in the classification process.

IV. Related Works

A lot of researches have been done about the detection of offensive languages especially Arabic. So, our results are compared with one of them. To make it fair enough, the comparison was done for the same classifiers, stemmer and feature extraction models, but the dataset was different as we didn't find the one they've used which may affect the results a little bit. The comparison is done with a paper named "HATE SPEECH CLASSIFICATION IN ARABIC TWEETS" and released in June 2020 from the University of Jordan, the paper used the Naive Bias, Random Forest Support Vector Machine and used both Bag of Words and TF – IDF feature extraction models once with ISRI stemmer and once without any stemmer. According to [Abushariah, 2020] results, they are shown in the following tables. Please note that RF stands for Random Forest model, NB stands for Naive Bayes model, and SVM stands for support vector machine model.

Algorithm	Feature Extraction	Precision	Recall	F – Measure	Accuracy
RF	BOW	89%	61%	72%	78%
	TF-IDF	88%	57%	69%	77%
NB	BOW	72%	75%	74%	75%
	TF-IDF	69%	76%	73%	73%
SVM	BOW	84%	75%	79%	82%
	TF-IDF	87%	74%	80%	83%

TABLE V: paper's results Without stemming

Algorithm	Feature Extraction	Precision	Recall	F – Measure	Accuracy
RF	BOW	85%	75%	80%	82%
	TF-IDF	88%	71%	79%	82%
NB	BOW	68%	52%	59%	66%
	TF-IDF	64%	56%	60%	65%
SVM	BOW	81%	80%	80%	82%
	TF-IDF	86%	78%	82%	84%

TABLE VI: paper's results With stemming

As seen, the results are pretty close to each other for the shared models between our work and the paper. In some cases our models had higher accuracy than them and in others they had it higher but with a really small difference. The comparison was done with this paper, because it's the closest paper to our work we could find, also we've have seen some other results on the internet and we find that the their results also fit with us, but we didn't include them, since they don't share more than one model with us. In general, our results were pretty good for current technologies used for offensive language detection with the informal Arabic language as proved by comparing with other research papers.

V. Conclusion and Future Work

In this project, we have learned a lot about NLP and IR techniques, and their libraries like NLTK and camel tool which they are very rich of functionalities that helped us in the preprocessing steps such as tokenization, stop-word removal, deleting diacritics and stemming, also we have learned how the feature extraction process can be done, and learning a lot about classification algorithms and learning about React framework as well as Flask framework to build a nice GUI which help us to host our model on the server to let Arabic people to test it.

As a future plan, this project leads us to spend more time in learning very complex models that can be used in the feature extraction process to give a notice better performance than TD-IDF model like Bert model, and trying to deploy our models on a server, and finally collecting a bigger dataset to generalize our result as much as we can.

References

[Abushariah, 2020] Abushariah, A. M. (2020). Hate speech classification in arabic tweets.