Int. J. Advance Soft Compu. Appl, Vol. 13, No. 2, July 2021
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Offensive Language Detection in Social Networks for Arabic Language Using Clustering Techniques

Tarek Kanan[1], Ghassan G. Kanaan[2], Riyad Al-Shalabi[3], and Amal Aldaaja[4]

[1]Computer Science Department, AlZaytoonah University, Amman, Jordan
[2, 3, 4]International Center for Scientific Research and Studies, Amman, Jordan
[1] tarek.kanan@zuj.edu.jo, [2] ghassan.kanaan@yahoo.com, [3] rfs64@yahoo.com,
[4] amalmohammad555@yahoo.com

Abstract

With the advent of social networks, the users have obtained a golden opportunity to express their opinions using text and multimedia. However, some users abused these platforms by introducing acts such as Cyber-Bullying and Cyber-Harassment. Despite the various negative health and social effects, the works proposed toward the detection of these acts are still limited, especially in non-English languages. In Arabic, few works studied this phenomenon. These works had limited datasets. As the number of available training datasets are limited, it is still hard to train classifiers to detect these acts. Therefore, clustering has posed as an alternative solution to tackle this difficulty. In this work, we propose the use of clustering to detect Cyber-Bullying and Cyber-Harassment. We adopted various clustering algorithms including K-Means and Expectation Maximization (EM). Moreover, we used various natural language processing (NLP) tools for this objective. The results illustrate that the training time of K-Means is significantly smaller than that of EM in all the conducted experiments. As for the accuracy, the two clustering methods showed different performance based on the variance in the used NLP settings.

Keywords*: Machine Learning, Clustering, Social Media, Natural language Processing, Arabic Text*

1 Introduction

The social networks have been increasing rapidly in the last two decades. With this increase, their impact on the society has become significant. These platforms are playing important roles in many aspects of our lives. They provided the chance to users to express their opinions and emotions. Unfortunately, these platforms have been abused by certain users by performing cyber-bullying and cyber-harassment.

These acts have negative effects in the health and in the social aspects. It has become more and more vital to fight against these acts.

Despite the importance of this topic, the proposed works to detect cyber-bullying and cyber-harassment are still limited, especially in non-English languages. In Arabic language, few works only studied this issue [1-3], but these works suffered from the limited availability of datasets in this domain. However, as having sufficient training datasets is important for the classification process, clustering could pose as an alternative solution when training datasets are not sufficient.

In our research, we used unsupervised machine to separate negative posts from regular ones. In details, we used K-Means [4] and Expectation Maximization [5] for this sake. As evaluation measurements, we used clustering time, Sum of Squared Error, and Log Likelihood. We used the same dataset used in our previous work in [3]. We studied the performance of these clustering methods in isolating such negative posts. Furthermore, we studied the effect of various NLP techniques such as the stemming and stopword removal on the performance of the clustering.

The contributions of this work are as follows:

- Providing an efficient Arabic cyber-bullying and cyber-harassment method on social media when labeled data is not available.
- Providing a comprehensive study using various NLP techniques and multiple social media datasets.
- Insisting on the importance of Arabic offensive language detection on social media.

The rest of the study is planned as follows: Section two is the Literature Review section. Section three provides the used methodology. Section four is the experimental results section along with the discussion. Finally, section five is the conclusion and the future work section.

2 Related Work

[6] proposed the use of SVM to correct wrong Arabic words. The author's utilized 1,300,000 tweets and created a bigram-words list for the misspelled words. [7] compared stemming and light stemming methods. The results proved that the light stemmer had more accuracy. [8] proposed the P-Stemmer, which is a new stemmer that can be used in text preprocessing. [9] provided Shami corpus, the first Levantine Dialect Corpus (SDC) that contains four Arabic dialects from Palestine, Jordan, Lebanon, and Syria. [10] used the Frequency Ratio Accumulation Method (FRAM) as a classifier. [11] presented AlKhalil Morpho analyzer version two. Its accuracy was around 99% for the words that have been analyzed. [12] compared various classifiers for Arabic text categorization. In the results, NB outperformed the other classifiers. [13] extracted useful information from big data. SVM classifier proved to have the superior performance. [14] combined Naïve Bayes algorithm with Support vector machine by stacking to improve text classification. [15]

proposed ITDGM, a new multi-label text classification. [16] proposed a document denoising solution using a novel Key phrases extraction algorithm. [17] compared BTO and TF-IDF with clustering. The results show that BTO is better than TF-IDF in the clustering process. [18] used sentiment analysis to detect cyber-bullying on Twitter. The result was achieved around 70%. [19] examined the prevalence of cyber-bullying among university on the Internet. They showed that text messages and media social communication are the main sources of electronic bullying. Specifically, comments and forum participation. [20] argued that the cause of the cyberbullying is the distress. [1] proposed a predictive modeling detection of negative posts in Arab social media. In this context, SVM had the top performance using the N-gram feature. [2] provided a cyberbullying detection method using a PHP language script for Twitter data and script in python to extract data from Facebook. [3] studied this problem from a classification point of view. They collected an Arabic dataset for this sake, and compared the performance of various classifiers in detecting offensive language in Arabic social networks. [21] used various machine learning techniques to classify Arabic posts based on their political orientation. [22-27] surveys the main natural language processing techniques in Arabic language; specifically, in social media.

3 Methodology

3.1 Introduction

In this work, we used the same dataset that was used in [3], which is composed of 6,138 records. The data was gathered from Facebook and Twitter platforms. More details about the dataset is provided in subsection 4.1.

3.2 Preprocessing

Before performing the clustering operation, we conducted a set of preprocessing steps, which include normalization, stopword removal, and stemming.

3.2.1 Stemming

Stemmer is used to convert all words to their stem. This step is specifically important as it aims at increasing the frequency of terms that appear frequently but in various formats such as verbs, nouns, adjectives, and so on. Returning these formats to their stem contributes in increasing the importance of these terms. In our work, P-Stemmer was used [8].

3.3 Clustering Methods

In this work, we compared two clustering methods; K-Means [4] and Expectation Maximization [5]. These methods are used as they are very widely used in the literature. As our goal is to detect offensive from non-offensive posts, we set K=2 for the K-Means. It is worth mentioning that we compared the performance of the two clustering methods based on various NLP preprocessing settings as illustrated in the experimental section.

4 Experimental Results

4.1 Dataset

The used dataset contained 6,138 records. The sources of the dataset were Facebook and Twitter. In details, we gathered 2,138 Facebook records; 1,000 of which were positive and 1,138 were negative. Regarding Twitter, we collected 4,000 records; 2,100 of which were positive and 1,900 were negative.

4.2 Evaluation Measurements

To evaluate the clustering algorithms, we used Training Time, Sum of Squared Error (SSE), and Log likelihood measurements. They are defined as follows:

Clustering Time: It is the time needed by the clustering method to perform the clustering operation on the dataset.

Sum of Squared Error: It is the sum of the squared distances between each point in the dataset and its cluster centroid. This measurement is used to indicate the accuracy of the clustering.

Log likelihood measurement: It is another measurement to find the accuracy of the clustering by finding the probability of each point belonging to its cluster centroid.

4.3 System Settings

In this research, we used WEKA 3.8 toolkit [28]. As for K-means, K=2 was used. We used Quad Core i7 with 3.1GHZ CPU speed, 16G Ram memory.

4.4 Results

we used two clustering algorithms include; K-Means and EM clustering. We use the option "Use Training Set" in WEKA, where the clustering independently trains input (datasets) without external human intervention, datasets are grouped into two groups, based on the number of clusters determined from each cluster's options. The results display the time is taken for the training process and cluster sum of squared errors. Fig 1 shows the options of clustering available in WEKA that include; Choose Clustering, Number of Clusters, and Cluster-Mode. Our methods were standardized with many related previous researches [29-31]

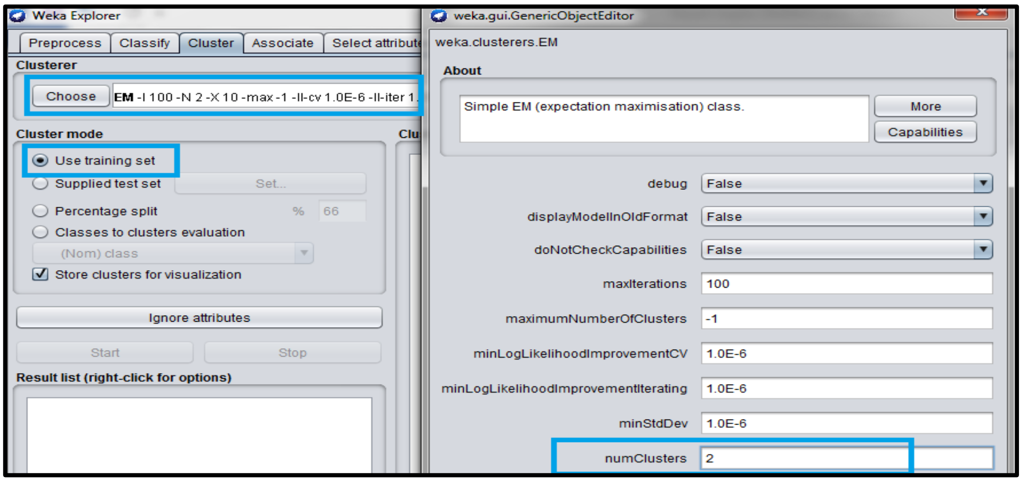

Fig 1: Options of Clustering in WEKA

4.4.1 Apply Clustering Algorithms with all ANLP Tools on all Dataset

Table 1 shows the results of applying the K-Means clustering with all NLP tools on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Cluster Sum of Squared Errors.

Table 1: Training Time and Cluster Sum of Squared Error of K-means Clustering with all NLP Tools

Clustering	Training Time in Seconds	Cluster Sum of Squared Error
K-Means	22.77 s	7796.363

Table 2 shows the results of applying the EM clustering with all NLP tools on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Log-likelihood.

Table 2: Training Time and Log-likelihood of EM Clustering with all NLP Tools

Clustering	Training Time in Seconds	Log-likelihood
EM	164.15 s	2648.158

4.4.2 Apply Clustering Algorithms without Stemming on all Dataset

Table 3 shows the results of applying the K-Means without Stemming on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Cluster Sum of Squared Errors.

Table 3: Training Time and Cluster Sum of Squared Error of K-means Clustering without Stemming

Clustering	Training Time in Seconds	Cluster Sum of Squared Error
K-Means	3.87 s	75615.567

Table 4 shows the results of applying the EM clustering without stemming on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Log-likelihood.

Table 4: Training Time and Log-likelihood of EM Clustering without Stemming

Clustering	Training Time in Seconds	Log-likelihood
EM	150.87 s	3004.61931

4.4.3 Apply Clustering Algorithms without Stop-Word Removal on all Dataset

Table 5 shows the results of applying the K-Means without Stop-word Removal on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Cluster Sum of Squared Errors.

Table 5: Training Time and Cluster Sum of Squared Error of K-means Clustering without Stop-Word Removal

Clustering	Training Time in Seconds	Cluster Sum of Squared Error
K-Means	18.73 s	90990.639

Table 6 shows the results of applying the EM clustering without Stop-Word Removal on all datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Log-likelihood.

Table 6: Training Time and Log-likelihood of EM Clustering without Stop-Word Removal

Clustering	Training Time in Seconds	Log-likelihood
EM	162.71 s	2258.2683

4.4.4 Apply Clustering Algorithms with all ANLP Tools on Facebook Dataset

Table 7 shows the results of applying the K-Means with all NLP on Facebook datasets. The results shown in the table was measured using Time to Take to Build a Model (Training Time) and Cluster Sum of Squared Errors.

Table 7: Training Time and Cluster Sum of Squared Error of K-means with all NLP Tools on Facebook Dataset

Clustering	Training Time in Seconds	Cluster Sum of Squared Error
K-Means	4.64 s	26071.567

Table 8 shows the results of applying the EM clustering with all NLP tools on Facebook datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Log-likelihood.

Table 8: Training Time and Log-likelihood of EM Clustering with all NLP Tools on Facebook Dataset

Clustering	Training Time in Seconds	Log-likelihood
EM	55.09 s	3227.71917

4.4.5 Apply Clustering with all ANLP Tools on Twitter Dataset

Table 9 shows the results of applying the K-Means with all NLP on Twitter datasets. The results shown in the table was measured using Time to Take to Build a Model (Training Time) and Cluster Sum of Squared Errors.

Table 9: Training Time and Cluster Sum of Squared Error of K-means with all NLP Tools on Twitter Dataset

Clustering	Training Time in Seconds	Cluster Sum of Squared Error
K-Means	3.55 s	18414.02

Table 10 shows the results of applying the EM clustering with all NLP tools on Twitter datasets. The results shown in the mentioned table was measured using Time to Take to Build a Model (Training Time) and Log-likelihood.

Table 10: Training Time and Log-likelihood of EM Clustering with all NLP Tools on Twitter Dataset

Clustering	Training Time in Seconds	Log-likelihood
EM	51.34 s	3606.4669

Table 11 and Fig 2 show the comparison of the training time of K-Means clustering on all datasets between with all NLP tools, without stemming, and without stop-word removal.

Table 8: Comparison of Training Time for K-means with different NLP Tools on all Dataset

Clustering	Time of Training with all ANLP	Time of Training without Stemming	Time of Training with Stop-Word Removal
K-Means	22.77 s	3.87 s	18.73 s

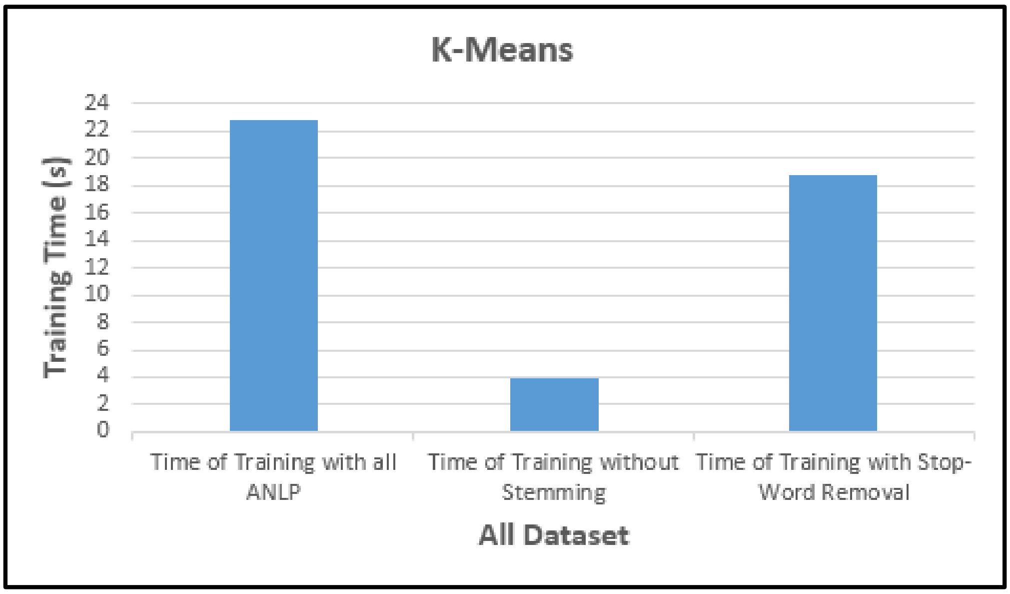

Fig 2: Comparison of Training Time for K-means with different NLP Tools on all Dataset

Table 12 and Fig 3 show the comparison of the training time of K-Means clustering with all NLP tools on Twitter datasets and Facebook datasets.

Table 9: Comparison of Training Time for K-means with all NLP Tools on Facebook and Twitter Dataset

Clustering	Time of Training for Facebook	Time of Training for Twitter
K-Means	4.64 s	3.55 s

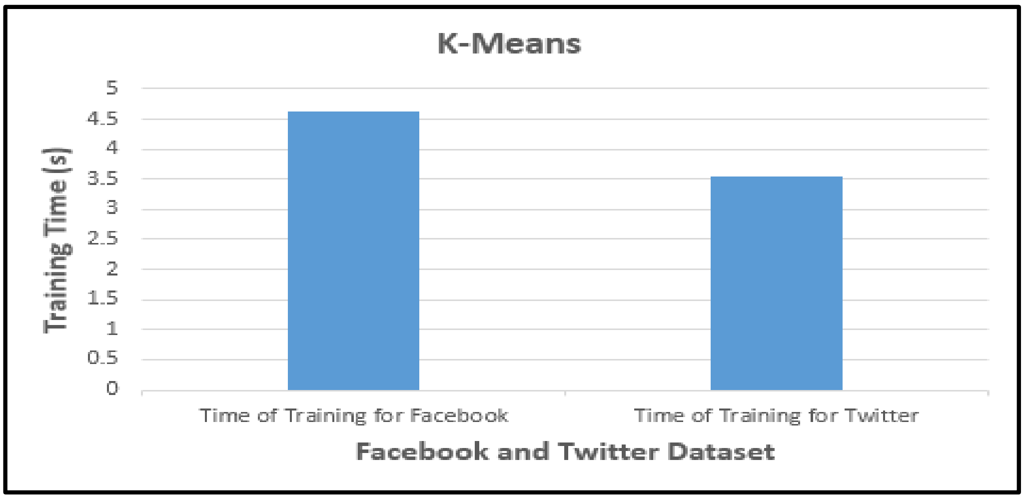

Fig 3: Comparison of Training Time for K-means with all NLP Tools on Facebook and Twitter Dataset

Table 13 and Fig 4 show the comparison of the cluster of sums squared error of K-Means clustering on all datasets between with all ANLP tools, without stemming, and without stop-word removal.

Table 10: Comparison of Squared Error for K-means with different NLP Tools on all Dataset

Clustering	Squared Error with all ANLP	Squared Error without Stemming	Squared Error with Stop-Word Removal
K-Means	7796.363	7561.567	90990.639

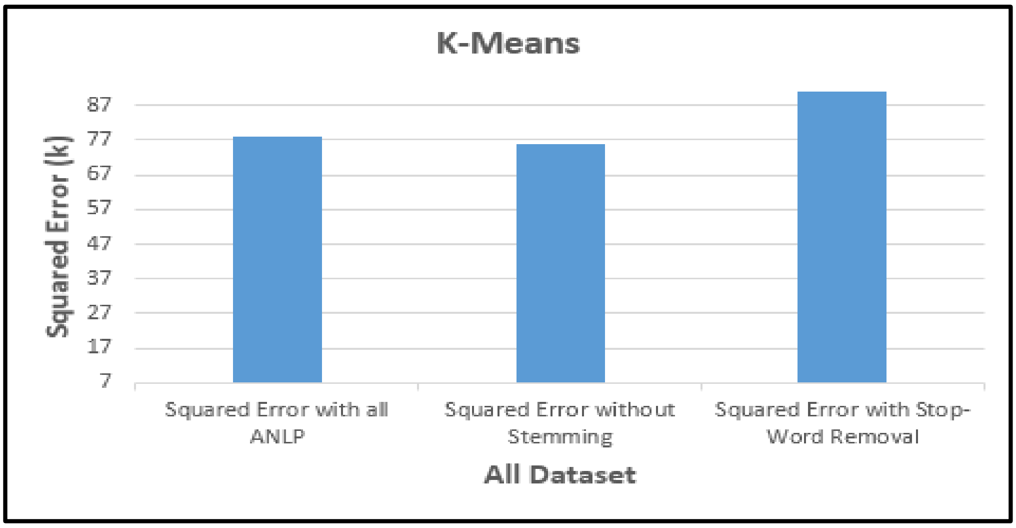

Fig 4: Comparison of Squared Error for K-means with different NLP Tools on all Dataset

Table 14 and Fig 5 show the comparison of the cluster of sums squared error of K-Means clustering with all NLP tools on Twitter datasets and Facebook datasets.

Table 14: Comparison of Squared Error for K-means with all NLP Tools on Facebook and Twitter Dataset

Clustering	Squared error for Facebook	Squared error for Twitter
K-Means	26071.567	18414.02

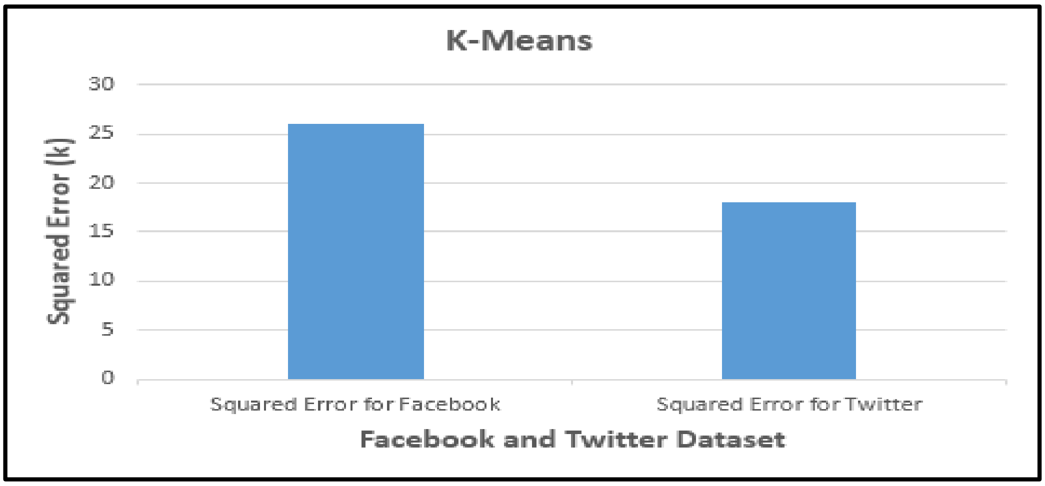

Fig 5: Comparison of Squared Error for K-means with all NLP Tools on Facebook and Twitter Dataset

Table 15 and Fig 6 show the comparison of the training time of EM clustering on all datasets between with all ANLP tools, without stemming, and without stop-word removal.

Table 11: Comparison of Training Time for EM with different NLP Tools on all Dataset

Clustering	Time of Training with all ANLP	Time of Training without Stemming	Time of Training with Stop-Word Removal
EM	164.15 s	150.87 s	162.71 s

Fig 6: Comparison of Training Time for EM with different NLP Tools on all Dataset

Table 16 and Fig 7 show the comparison of the training time of EM clustering with all NLP tools on Twitter datasets and Facebook datasets.

Table 12: Comparison of Training Time for EM with all NLP Tools on Facebook and Twitter Dataset

Clustering	Time of Training for Facebook	Time of Training for Twitter
EM	55.09 s	51.34 s

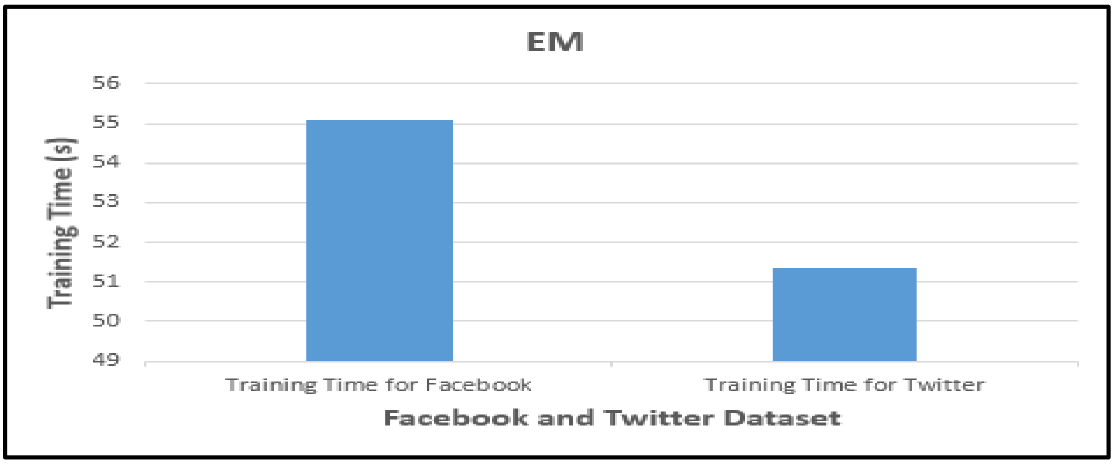

Fig 7: Comparison of Training Time for EM with all NLP Tools on Facebook and Twitter Dataset

Table 17 and Fig 8 show the comparison of the log-likelihood of EM clustering on all datasets between with all NLP tools, without stemming, and without stop-word removal.

Table 13: Comparison of Log-likelihood for EM with different NLP Tools on all Dataset

Clustering	Log-likelihood with all ANLP	Log-likelihood without Stemming	Log-likelihood with Stop-Word Removal
EM	2648.158	3004.61931	2258.2683

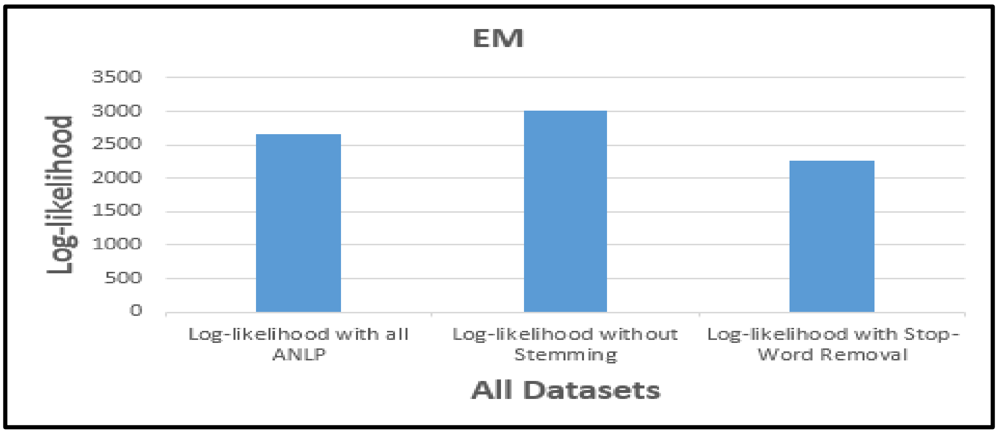

Fig 8: Comparison of Log-likelihood for EM with different NLP Tools on all Dataset

Table 18 and Fig 9 show the comparison of the log-likelihood of EM clustering with all NLP tools on Twitter datasets and Facebook datasets.

Table 18: Comparison of Log-likelihood for EM with all NLP Tools on Facebook and Twitter Dataset

Clustering	Log-likelihood for Facebook	Log-likelihood for Twitter
EM	3227.71917	3606.4669

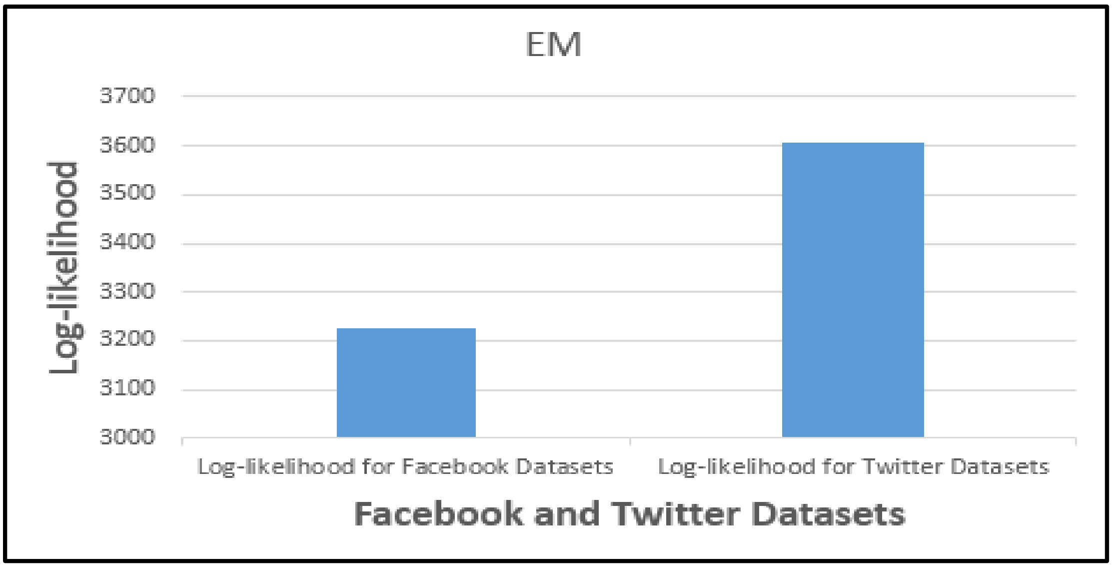

Fig 9: Comparison of Log-likelihood for EM with all NLP Tools on Facebook and Twitter Dataset

6 Conclusion

This work used clustering in the detection of cyber-bullying and cyber-harassment due to the difficulty in the use of classification as training datasets are still limited. As our concentration is on social networks, we used a dataset collected from Facebook and Twitter platforms.

The experimental work showed that the training time of K-Means is significantly smaller than that of EM in all the conducted experiments. As for the accuracy, the two clustering methods showed different performance based on the variance in the used NLP settings. As for the EM, it had the best log-likelihood accuracy when applying NLP without removing Stopwords, while for K-means, its best SSE accuracy was when using all NLP without stemming. This is reasonable as different accuracy measurements can act differently with different preprocessing settings.

When separating datasets according to their origin, twitter data showed faster performance than Facebook data. This is due to the shorter size of twitter data posts in general when compared with Facebook data. As for the accuracy, K-Means showed a better accuracy on Twitter data, while EM performance was similar with a slight improvement when using Facebook data.

Future work can be conducted in many directions. First, the use of semi-supervised clustering to combine classification and clustering would be an interesting topic. Second, using more clustering methods would contribute in making the comparison more comprehensive. Finally, the detection of these phenomena in dialect Arabic needs more concentration

References

[1] A. Alakrot, L. Murray, N. S. Nikolov, Towards Accurate Detection of Offensive Language in Online Communication in Arabic, Procedia Computer Science, Vol. 142, pp. 315-320, November, 2018.

[2] B. Haidar, M. Chamoun, A. Serhrouchni, A Multilingual System for Cyberbullying Detection: Arabic Content Detection Using Machine Learning, Advances in Science, Technology and Engineering Systems Journal, Vol. 2, No. 6, pp. 275-284, December, 2017.

[3] T. Kanan, A. Aldaaja, and B. Hawashin, 2020. Cyber-Bullying and Cyber-Harassment Detection Using Supervised Machine Learning Techniques in Arabic Social Media Contents. Journal of Internet Technology, Vol. 21, No. 5, pp.1409-1421.

[4] J. MacQueen, 1967, June. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

[5] A.P., Dempster, N.M. Laird, and D.B. Rubin, 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), pp.1-22.
[6] E. A. Abozinadah, J. Jones, Improved Micro-blog Classification for Detecting Abusive Arabic Twitter Accounts, International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol. 6, No. 6, pp. 17-28, November, 2016.
[7] R. Duwairi, M. Al-Refai, N. Khasawneh, Stemming versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization, 2007 IEEE Innovations in Information Technologies (IIT), Dubai, UAE, 2007, pp. 446-450.
[8] T. Kanan, E. A. Fox, Automated Arabic Text Classification with p-stemmer, Machine Learning, and a Tailored News Article Taxonomy, Journal of the Association for Information Science and Technology, Vol. 67, No. 11, pp. 2667-2683, November, 2016.
[9] K. Abu Kwaik, M. K. Saad, S. Chatzikyriakidis, S. Dobnik, Shami: A Corpus of Levantine Arabic Dialects, Eleventh International Conference on Language Resources and Evaluation (LREC18), Miyazaki, Japan, 2018, pp. 3645-3652.
[10] R. M. Sallam, H. M. Mousa, M. Hussein, Improving Arabic Text Categorization Using Normalization and Stemming Techniques, International Journal of Computer Applications, Vol. 135, No. 2, pp. 38-43, February, 2016.
[11] M. Boudchiche, A. Mazroui, M. O. A. O. Bebah, A. Lakhouaja, A. Boudlal, Alkhalil Morpho Sys 2: A Robust Arabic Morpho-syntactic Analyzer, Journal of King Saud University-Computer and Information Sciences, Vol. 29, No. 2, pp. 141-146, April, 2017.
[12] R. M. Duwairi, R. Al-Zubaidi, A Hierarchical k-nn Classifier for Textual Data, International Arab Journal of Information Technology, Vol. 8, No. 3, pp. 251-259, July, 2011.
[13] R. Elhassan, M. Ahmed, Arabic Text Classification on Full Word, International Journal of Computer Science and Software Engineering (IJCSSE), Vol. 4, No. 5, pp. 114-120, May, 2015.
[14] A. O. Christiana, O. S. Oladeji, A. T. Oladele, Binary Text Classification Using an Ensemble of Naïve Bayes and Support Vector Machines, Computer Science & Telecommunications, Vol. 52, No. 2, pp. 37-45, October, 2017.
[15] L. Peng, Y. Liu, Gravitation Theory Based Model for Multilabel Classification, International Journal of Computers, Communications & Control, Vol. 12, No. 5, pp. 689-703, October, 2017.
[16] H. Froud, I. Sahmoudi, A. Lachkar, An Efficient Approach to Improve Arabic Documents Clustering Based on a New Keyphrases Extraction Algorithm, Second International Conference on Advanced Information Technologies and Applications (ICAITA-2013), Dubai, UAE, 2013, pp. 243-256.
[17] H. Al-Rubaiee, K. Alomar, Clustering Students' Arabic Tweets Using Different Schemes, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 4, pp. 276-280, April, 2017.

[18] H. Sanchez, S. Kumar, Twitter Bullying Detection, https://users.soe.ucsc.edu/~shreyask/ism245-rpt.pdf, 2011.

[19] E. Whittaker, R. M. Kowalski, Cyberbullying via Social Media, Journal of School Violence, Vol. 14, No. 1, pp. 11-29, 2015.

[20] B. O'Dea, A. Campbell, Online Social Networking and the Experience of Cyber-bullying, in: B. Wiederhold, G. Riva (Eds.), Annual Review of Cyber therapy and Telemedicine 2012: Advanced Technologies in the Behavioral, Social and Neurosciences, Vol. 181 of Studies in Health Technology and Informatics, IOS Press, 2012, pp. 212-217.

[21] R. Abooraig, S., Al-Zu'bi, T., Kanan, B., Hawashin, M., Al Ayoub, and I. Hmeidi, 2018. Automatic categorization of arabic articles based on their political orientation. Digital Investigation, Vol. 25, pp.24-41.

[22] T. Kanan, et. al., 2019, April. A review of natural language processing and machine learning tools used to analyze Arabic social media. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 622-628). IEEE.

[23] Kanan, Tarek, and Edward A. Fox. "Automated Arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy." Journal of the Association for Information Science and Technology 67, no. 11 (2016): 2667-2683.

[24] Kanan, Tarek, Ahamd Turki Obaidat, and Majduleen Al-Lahham. "SmartCert blockchain imperative for educational certificates." In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 629-633. IEEE, 2019.

[25] Yang, Seungwon, Tarek Kanan, and Edward Fox. "Digital library educational module development strategies and sustainable enhancement by the community." In International Conference on Theory and Practice of Digital Libraries, pp. 514-517. Springer, Berlin, Heidelberg, 2010.

[26] Kanan, Tarek, Souleiman Ayoub, Eyad Saif, Ghassan Kanaan, Prashant Chandrasekarar, and Edward A. Fox. Extracting named entities using named entity recognizer and generating topics using latent Dirichlet allocation algorithm for arabic news articles. Department of Computer Science, Virginia Polytechnic Institute & State University, 2015.

[27] Kanan, Tarek, Raed Kanaan, Omar Al-Dabbas, Ghassan Kanaan, Ali Al-Dahoud, and Edward Fox. "Extracting named entities using named entity recognizer for Arabic news articles." International Journal of Advanced Studies in Computers, Science and Engineering 5, no. 11 (2016): 78-84.

[28] E. Frank, M. A. Hall, and I. H. Witten 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[29] T. Kanan, O. Sadaqa, A. Almhirat and E. Kanan, "Arabic Light Stemming: A Comparative Study between P-Stemmer, Khoja Stemmer, and Light10 Stemmer," 2019 Sixth International Conference on Social Networks Analysis,

Management and Security (SNAMS), 2019, pp. 511-515, doi: 10.1109/SNAMS.2019.8931842.
[30] Alhoori, Hamed, Sagnik Ray Choudhury, Tarek Kanan, Edward Fox, Richard Furuta, and C. Lee Giles. "On the relationship between open access and altmetrics." iConference 2015 Proceedings (2015).
[31] Kanan, Tarek, Xuan Zhang, Mohamed Magdy, and Edward Fox. "Big data text summarization for events: A problem-based learning course." In Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 87-90. 2015.

Author Biography

Dr. Tarek Kanan is an assistant professor in the Department of Artificial Intelligence at Al-Zaytoonah University of Jordan. He obtained his PhD degree in 2015 from Virginia Polytechnic Institute and State University (Virginia Tech), Virginia-USA. His research interests are in the broad areas of Artificial Intelligence. He is particularly interested in Machine Learning, Deep Learning, Natural Language Processing, and Data Science. More specifically, he is passionate about multilingual Text Classifications, Summarization, Natural Language Processing, and Information Retrieval. He worked on the theoretical analysis of computationally efficient methods for large or otherwise complex prediction problems. He had several prestigious Journal/Conference publications in collaboration with many researches all over the world. He served in many international and national conferences’ committees.