Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 142 (2018) 315–320

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Arabic Computational Linguistics (ACLing 2018),
November 17-19 2018, Dubai, United Arab Emirates

Towards Accurate Detection of Offensive Language in Online Communication in Arabic

Azalden Alakrot[a], Liam Murray[b], Nikola S. Nikolov[a]

[a] *Department of Computer Science and Information Systems, University of Limerick, Ireland*
[b] *School of Languages, University of Limerick, Ireland*

Abstract

We present the results of predictive modelling for the detection of anti-social behaviour in online communication in Arabic, such as comments which contain obscene or offensive words and phrases. We collected and labelled a large dataset of YouTube comments in Arabic which contains a broad range of both offensive and inoffensive comments. We used this dataset to train a Support Vector Machine classifier and experimented with combinations of word-level features, N-gram features and a variety of pre-processing techniques. We summarise the pre-processing steps and features that allow training a classifier which is more precise, with 90.05% accuracy, than classifiers reported by previous studies on Arabic text.

Keywords: Anti-social behaviour online; offensive language detection; harassment detection; Arabic dataset; text mining; SVM for offensive language detection in Arabic;

1. Introduction

Several studies in the early years of the Web defined the term *flaming* as hostile intentions characterised by words of profanity, obscenity, curse and insults resulting from reckless behaviour and which hurt a person or group of people [4]. Some studies suggest that flaming is a social or cultural tendency [21]. However, others suggest that flaming depends on the topic of the debate, the confidence in the provision of anonymity, the participants' proximity and familiarity with the group members [17]. Unlike direct talk, in electronic text communication there is no eye-to-

* Corresponding author.
E-mail addresses: Azalden.Alakrot@gmail.com (Azalden Alakrot)., Liam.Murray@ul.ie (Liam Murray)., Nikola.Nikolov@ul.ie (Nikola S. Nikolov).

10.1016/j.procs.2018.10.491

eye contact or sense of personal presence that dictate a certain social etiquette. People who are separated from each other geographically can attack each other verbally without fear of physical harm [11].

The advance of communication technologies in recent years has further increased the impact and the awareness of the negative effects of flaming. While early online communication was limited to chat rooms, the emergency of new technologies, such as instant messaging and highly popular social media platforms, such as Instagram, Facebook, Twitter and YouTube, significantly increased the volume and variety of online social interaction. Consequently, the negative impact of offensive language in online communication has become a problem affecting virtually every household with access to the Internet.

The majority of previous studies concerning offensive language in online communication are in English, while there are only a few studies that tackle the same issue in Arabic [2, 24]. Moreover, these studies conduct their experiments on relatively small datasets collected from *Twitter*. For instance, Mubarak *et al.* employ a dataset of 1100 annotated tweets [24]. Furthermore, the data employed in these studies is collected based on few predefined profane words, which lessens the predictive ability of the proposed models.

In our work, we employ a dataset collected from another social media platform (YouTube) with the size of the labelled portion of it being significantly larger than the size of the datasets employed in the aforementioned studies. In addition to that, we collected data from YouTube videos where a large number of users comment offensively (comments on controversial video footages of celebrities) without putting a limit on the range of abusive/profane words being recognised as such.

2. Dataset

YouTube enables communication between people who can choose to remain anonymous, thus creating an environment for its users where they can speak without restrictions and *misbehave* toward other users. It is common for reckless users to use obscene words and phrases to offend others, and such incidents occur repeatedly [26, 20]. We took the opportunity to collect a great deal of such comments and build a dataset of 15,050 comments, suitable for training predictive models. This dataset was collected in July 2017, the method of collection is by selecting some YouTube channels that upload videos about celebrities in the Arab world. These videos display celebrities on the media in controversial footage. This type of footages provoke viewers, leading some of them to utilise offensive/abusive language in their posts. These videos were uploaded in the period from 2015 to 2017. Out of the corpus of comments on 150 videos, we selected 9 videos with a high number of comments for building the dataset. We assume that a greater number of comments may indicate more intense debate and higher diversity of offensive language used.

The labelling process was accomplished by three annotators from three different Arab countries. Our intention was to ensure that comments labelled as offensive are understood as such by people from different Arab regions. The number of positives is 5,817 comments, i.e. 39% of the entire dataset. These are comments that at least two of the three annotators consider offensive. The number of comments labelled the same (either offensive or inoffensive) by all three annotators is 10,715, i.e. the inter-annotator agreement is 71%. We present details about the data collection process and a statistical analysis of the dataset in an accompanying paper [3]. We have also made the dataset publicly available[1].

3. Text Pre-processing

Text pre-processing is a typically beneficial first step in the text mining process. It may involve operations, such as *tokenization*, *filtering* and *normalisation* [23, 18, 25].

Arabic is a Semitic language with script from right to left, and it has twenty-eight letters that form words; these letters are also used in other languages, such as Persian and Urdu [5]. The three predominant types of words in Arabic are nouns, verbs and particles [14]. Therefore, results in tokenization, filtering and normalisation from previous research with other languages are generally applicable to Arabic too. However, there are some notable differences. For

[1] The dataset used in this work is publicly available at https://goo.gl/27EVbU.

example, since some Arabic letters are similar phonetically, users on social media typically misspell words by using the wrong but phonetically similar letters (details presented in Section 3.4).

3.1. Tokenization

Words can be formed as a stream of letters and separated by a set of delimiters [31]. The first step in tokenization is to separate the alphabetic sequences into tokens. In the tokenization phase, choosing characters as delimiters depends on the application as some characters may or may not be delimiters in different scenarios [30, 31]. Characters, such as space, tab and newline are most of the time regarded as delimiters and are not counted as tokens; these are often called *white space*. Other characters can be used as delimiters too, such as ( ) ¡ ¿ ! ? and ,. Arabic text has the same characteristics as English in this respect. Words in Arabic are separated by white space. However, in casual writing online, words are sometimes separated by comma only (without white space). Some examples of words separated by comma only in our dataset are "يفوز،بصراحة", "برواحكم,soso", "التاريخ،،فلة" and "لطيفة،يابنتي". Occasionally, in informal writing online, there is little attention paid to the formally correct use of delimiters between words. In this work we also consider the case when the comma character is the only delimiter between words.

Other factors also affect tokenization, such as N-gram co-occurrence and stemming [31]. In this study, we utilised the light Arabic stemmer ARLSTem to examine the effect of stemming on the predictive model's accuracy. ARLSTem eliminates the prefixes, suffixes and infixes from a word and reportedly compares well to other available stemmers [1].

3.2. Filtering

Filtering is the removal of punctuation marks, commas, diacritics (in Arabic) and selected words from the documents. Standard filtering excludes the most frequent words in the documents, such as articles, conjunctions and prepositions; these are known as *stop words*. Stop words occur excessively and typically do not contribute significant information for the purpose of text mining. Also, words that occur very rarely are likely to have no statistical significance and can be removed as well as outliers [22, 8, 31]. The process of filtering helps minimising the size of the number of features in the dataset, which would otherwise represent impediments to text mining [27]. The experiments in this study are conducted by removing the list of stop words provided by the Natural Language Toolkit (NLTK) [7], which list contains 248 words. Additionally, we only keep alphabetic characters, both Arabic and Latin, while other characters, such as punctuation characters and all other special characters, including numbers, are removed. We also remove the kashidas, as kashidas are mere word elongation characters [15].

3.3. Normalisation

In the normalisation phase letters are replaced by other letters for further improvement of the performance of text mining operations. This includes removing diacritics, replacing أ,إ , آ by ا, ى by ي and ة by ه [15, 28]. Furthermore, we replaced the originally Persian and Urdu letters that appear in the text by equivalent Arabic letters [15].

3.4. Extra Normalisation

Alongside the normalisation described above, we examined a few other normalisations that may enhance the results of text mining. There are some letters in Arabic that are confusing for many users, who use them interchangeably by mistake, because of the phonetic similarity between them. Such letters are ض and ظ, and also س and ص. Table 1 contains examples of misspelled words in our dataset.

4. Predictive Modelling Approach

In our experiments we utilise a supervised machine learning approach to build a classifier that can be deployed for detecting offensive language in online communication. The training dataset consists of YouTube comments, labelled

Table 1. Examples of words misspelled by interchangeably using phonetically similar letters.

Translation of the word	**Correction of the word**	**Misspelled word**
Scandal	فضيحة	فظيحه
She laughs	تضحك	تظحك
Favourite	مفضل	مفظل
She learnt by heart	حافظة	حافضه
Great	عظيم	عضيم
Person name Kazem	كاظم	كاضم
Dirty	وسخه	وصخه
Slut	الفاسقة	الفاصقة

and pre-processed as described in the previous sections. We handled each comment separately, regardless of its interconnections to other comments in a conversation. Previous research on text classification suggests that Support Vector Machines (SVM) is the classification algorithm that performs best in the case of text classification [19, 10, 12, 29, 13]. Accordingly, we selected SVM for our experiments and built an SVM classifier using the Python machine learning library [9] with word-level features. In addition, we examined N-gram features as well.

4.1. *Experiments Setup*

We trained an SVM classifier [13] using word-level features. We first trained our model on a dataset without applying any data pre-processing. The outcome of this run is used as a baseline. Next, we trained an SVM classifier with a pre-processed version of our dataset (see Section 3). Also, we experimented training SVM with and without the extra normalisation described in Section 3.4. Finally, we experimented with N-gram level features, as well. For the purpose of training an SVM classifier, it is required to transform the corpus of text documents (comments in our case) to a document-term matrix, an entry in which is the number of occurrences of a particular word in a particular document.

For evaluating a classifier's accuracy, we split the dataset into a training set and a test set and followed the widely accepted approach to apply 10-fold cross validation and observe *precision*, *recall* and the F1-score [6]. In addition, we plotted the receiver operating characteristics (ROC) curves. ROC curves are two-dimensional graphs in which the false positives rate *fp* is plotted on the x-axis and the true positives rate *tp* is plotted on the y-axis. They show the relative trade-off between benefit *tp* and cost *fp* and are a further indicator for the accuracy of a predictive model [16].

4.2. *Results*

Our baseline experiment is conducted with word-level features and without applying any data pre-processing. Its precision and recall for the class of offensive comments are 0.83 and 0.65, respectively, while the overall accuracy is 0.85. Compared to the results of Mubarak *et al.* [24], the recall is 20% better, while the precision is 15% worse. The difference in the results can be due to the different training datasets, thus it is hard to draw definite conclusions based on it.

Next, we applied data pre-processing both with and without stemming. A summary of our experimental results is presented in Table 2. When no stemming is applied, the precision is 2%, and recall 4% better than the baseline. Stemming further improves precision by 3%, and recall by 8%. We also experimented with applying extra normalisation (see Section 3.4) after the pre-processing step, however, we observed that it does not improve the results by more than 1% for precision and 2% for recall.

Overall, the results give evidence of the usefulness of the data pre-processing step and noise eliminating in terms of stemming and extra normalisation. The classifier performance using 10-fold cross validation is 90.05%. The ROC curves (for the two classes of offensive and inoffensive comments) in the case when the SVM classifier is built after

Table 2. Comparative performance of trained SVM classifiers.

	Precision	Recall	F1-score
Baseline	0.83	0.65	0.74
Pre-processing applied	0.85	0.69	0.76
Pre-processing applied with stemming	**0.88**	0.77	**0.82**
N-grams (1-5)	0.83	**0.80**	0.81
N-grams (1-5) and stemming	0.81	0.78	0.80

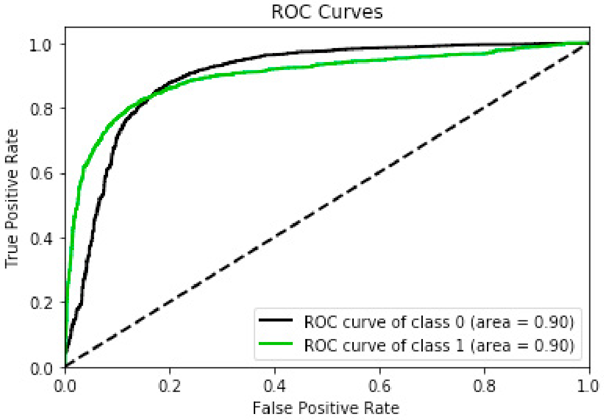

Fig. 1. ROC curves for the two classes (offensive/class 0 and inoffensive/class 1 comments) when an SVM classifier is applied after pre-processing and stemming.

data pre-processing and stemming are shown in Figure 1. The curves are closer to the left and top borders of the plot, which is another indicator that the accuracy of our classifier is high.

Additionally, we conducted experiments to study the impact of N-gram co-occurrence as features on the classifier performance. We attempted a range of values of N from 1 to 10 with the best results obtained for N ∈ (1..5). We observed a major improvement of the recall value, 15% improvement over the baseline; however, no improvement of precision. In comparison to the baseline, stemming combined with the use of N-gram features increases recall by 13%; though, precision decreases by 2%.

5. Conclusion

As the impact of anti-social behaviour in social networking platforms is growing with the increasing popularity of these platforms, the problem cannot be ignored. The main focus of text mining research for detection of offensive/abusive language in online communication conducted so far is mainly for English. There is no much attention being paid to the same problem in Arabic. Therefore, this study is conducted to enrich the current results in finding a solution that would contribute to the reduction of this phenomenon.

In this work, we conduct machine learning experiments with a dataset of YouTube comments in Arabic. To the best of our knowledge, our dataset is the largest dataset of Arabic text specifically designed for training predictive models for detection of offensive language in online communication. We report the impact of word-level features and popular pre-processing methods, including extra normalisation, on the performance of an SVM classifier trained to detect offensive comments. The results presented in this paper give evidence that our classifier improves previous results. We have observed that data pre-processing with stemming can be leveraged to enhance the detection of offensive language in casual Arabic text used in social media platforms. In addition, the utilisation of N-gram features improves the classifier's performance. However, the combination between stemming and N-gram features has negative effect on precision and recall in our experiments, thus we conclude that it is not beneficial to use both stemming and N-gram features within the same machine learning process.

In future work, other features can be considered, including contextual features of the text. Furthermore, it might be worthwhile to compare different classification approaches and analyse their performances, specifically the use of deep neural networks.

References

[1] Abainia, K., Ouamour, S., Sayoud, H., 2017. A novel robust Arabic light stemmer. Journal of Experimental & Theoretical Artificial Intelligence 29, 557–573.

[2] Abozinadah, E.A., Mbaziira, A.V., Jones, J., 2015. Detection of abusive accounts with Arabic tweets. Int. J. Knowl. Eng.-IACSIT 1, 113–119.

[3] Alakrot, A., Murray, L., Nikolov, N.S., 2018. Dataset construction for the detection of anti-social behaviour in online communication in Arabic. Procedia Computer Science , to appear.

[4] Alonzo, M., Aiken, M., 2004. Flaming in electronic communication. Decision Support Systems 36, 205–213.

[5] Atallah, A.S., Omar, K., 2008. Methods of Arabic language baseline detection–the state of art. IJCSNS 8, 137.

[6] Berry, M.W., Kogan, J., 2010. Text mining: applications and theory. John Wiley & Sons.

[7] Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.". URL: http://nltk.org/book.

[8] Blanchard, A., 2007. Understanding and customizing stopword lists for enhanced patent mapping. World Patent Information 29, 308–316.

[9] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., et al., 2013. Api design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238 .

[10] Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery 2, 121–167.

[11] Chapman, G., 1995. Cranks, fetishists and monomaniacs-flamers. New Republic 212, 13–15.

[12] Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297. URL: https://doi.org/10.1007/BF00994018, doi:10.1007/BF00994018.

[13] Cristianini, N., Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[14] Darwish, K., 2002. Building a shallow Arabic morphological analyzer in one day, in: Proceedings of the ACL-02 workshop on Computational approaches to semitic languages, Association for Computational Linguistics. pp. 1–8.

[15] Darwish, K., Magdy, W., et al., 2014. Arabic information retrieval. Foundations and Trends® in Information Retrieval 7, 239–342.

[16] Fawcett, T., 2006. An introduction to ROC analysis. Pattern recognition letters 27, 861–874.

[17] George, J.F., Easton, G.K., Nunamaker Jr, J.F., Northcraft, G.B., 1990. A study of collaborative group work with and without computer-based support. Information Systems Research 1, 394–415.

[18] Iiritano, S., Ruffolo, M., 2001. Managing the knowledge contained in electronic documents: a clustering method for text mining, in: Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on, IEEE. pp. 454–458.

[19] Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features, in: European conference on machine learning, Springer. pp. 137–142.

[20] Kawate, S., Patil, K., 2017. Analysis of foul language usage in social media text conversation. International Journal of Social Media and Interactive Learning Environments 5, 227–251.

[21] Kayany, J.M., 1998. Contexts of uninhibited online behavior: Flaming in social newsgroups on usenet. Journal of the American Society for Information Science 49, 1135–1141.

[22] Kilgarriff, A., Grefenstette, G., 2003. Introduction to the special issue on the web as corpus. Computational linguistics 29, 333–347.

[23] Mathiak, B., Eckstein, S., 2004. Five steps to text mining in biomedical literature, in: Proceedings of the second European workshop on data mining and text mining in bioinformatics, pp. 43–46.

[24] Mubarak, H., Darwish, K., Magdy, W., 2017. Abusive language detection on Arabic social media, in: Proceedings of the First Workshop on Abusive Language Online, pp. 52–56.

[25] Neto, J.L., Santos, A.D., Kaestner, C.A., Alexandre, N., Santos, D., et al., 2000. Document clustering and text summarization .

[26] Protalinski, E., 2011. 47% of Facebook walls contain profanity. URL: https://www.zdnet.com/article/47-of-facebook-walls-contain-profanity/. [Online; accessed Jun 2018].

[27] Saad, M.K., Ashour, W., 2010. Arabic text classification using decision trees, in: Proceedings of the 12th international workshop on computer science and information technologies CSIT, pp. 75–79.

[28] Sallam, R., Mousa, H., Hussein, M., 2016. Improving Arabic text categorization using normalization and stemming techniques. Int. J. Comput. Appl 135, 38–43.

[29] Vapnik, V., 1998. Statistical learning theory. 1998. Wiley, New York.

[30] Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F., 2010. Text mining: predictive methods for analyzing unstructured information. Springer Science & Business Media.

[31] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.