Use of Natural Language Processing to Identify Inappropriate Content in Text

Sergio Merayo-Alba, Eduardo Fidalgo, Víctor González-Castro, Rocío Alaiz-Rodríguez, and Javier Velasco-Mata

Department of Electrical, Systems and Automation Engineering, Universidad de León, Spain
sergiomerayo@gmail.com, {efidf, vgonc, rocio.alaiz, jvelm}@unileon.es

Abstract. The quick development of communication through new technology media such as social networks and mobile phones has improved our lives. However, this also produces collateral problems such as the presence of insults and abusive comments. In this work, we address the problem of detecting violent content on text documents using Natural Language Processing techniques. Following an approach based on Machine Learning techniques, we have trained six models resulting from the combinations of two text encoders, Term Frequency-Inverse Document Frequency and Bag of Words, together with three classifiers: Logistic Regression, Support Vector Machines and Naïve Bayes. We have also assessed StarSpace, a Deep Learning approach proposed by Facebook and configured to use a Hit@1 accuracy. We evaluated these seven alternatives in two publicly available datasets from the Wikipedia Detox Project: Attack and Aggression. StarSpace achieved an accuracy of 0.938 and 0.937 in these datasets, respectively, being the algorithm recommended to detect violent content on text documents among the alternatives evaluated.

1 Introduction

Due to the development of the Internet over the last years and the change in habits and behaviour of people regarding technology, Social Networks have gained more and more popularity and users, generating an impressive daily amount of comments about any topic. Unfortunately, this also implies that a significant amount of these comments may contain inappropriate content such as obscene, aggressive, rude, racist, sexist or violent sentences [1].

Nowadays, more and more children have access to the Internet [2], so it is imperative to prevent them to access inappropriate contents such as those mentioned before. When it comes to text, due to the large amount of material available, reviewing all of it is an unmanageable task to be efficiently accomplished by human inspectors. Therefore, it is necessary to develop automated solutions to filter inappropriate textual contents. Machine Learning methods applied to text classification can be used to build parental filters for online content and to detect illegal activities such as hate speeches [3].

In this work, we explored the use of some Natural Language Processing (NLP) and Machine Learning techniques to detect violent content in text. We have compared six combinations of encoder-plus-classifier, where the former is either Term frequency - Inverse document frequency (TF-IDF) or Bag of Words (BoW), and the latter is either Logistic Regression (LR), Support Vector Machines (SVM) or Naïve Bayes (NB). In addition, we have also assessed an alternative based on Deep Learning called StarSpace, an algorithm proposed by Facebook[1]. We evaluated these seven methods using two public datasets, with attacking aggressive comments respectively. Finally, we made a recommendation about the best approach to detect and classify inappropriate content on text documents. A scheme of this work is shown in Fig. 1

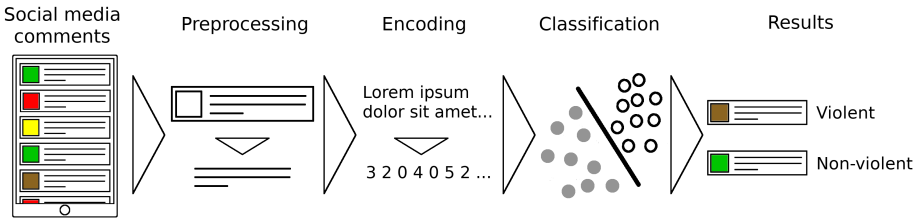

Fig. 1. Proposed pipeline to detect inappropriate content in text.

The rest of the work is presented as follows: in Section 2 we review the state-of-the-art in this field. Next, we describe the techniques used in this work in Section 3, and in Section 4 we explain the experimental settings, describe the datasets and discuss the achieved results. Finally, we summarize this work and discuss future lines of work in Section 5.

2 Related Work

2.1 Text Classification

According to the literature review carried out by Mironczuk and Protasiewicz in 2016 [4], the text classification process can be divided into six phases: (1) data acquisition, (2) data analysis and labelling, (3) feature construction and weighting, (4) feature selection and projection, (5) training of a classification model and, finally, (6) the solution evaluation. Different researches focus on different steps. For example, Bui et al. [5] focused on the data acquisition step on PDF documents where the relevant text was mixed with metadata or semi-structured

[1] https://research.fb.com/downloads/starspace/

text, while the work of Chen et al. [6] focused on the feature construction phase by comparing different features extractors for NB.

Rogati et al. [7] assessed the feature and selection phase comparing the performance of four classifiers, namely NB, Rocchio classifier, k-Nearest Neighbors (kNN) and SVM, using different feature selection strategies describing the samples of two well known datasets: RCV1 and Reuters-21578. The highest scores were achieved using only 3% of the available features.

The model training and the solution evaluation phases are usually studied together. An example is the work of Diab and El Hindi [8], where they compared different techniques for fine-tuning the NB algorithm. They used 53 text-classification datasets obtained from the UCI repository[2]. Their research concluded that a Multi Parent Differential Evolution (MPDE) allowed NB to reach a peak performance comparing to other tuning method such as regular Differential Evolution (DE), Genetic Algorithms (GA) and Simulated Annealing (SA).

2.2 Detection of Inappropriate Content in Text Documentation

Chavan and Shylaja [9] encoded text using the methods TF-IDF and N-grams. It was applied on a dataset with comments from the Kaggle website[3]. The dataset contained about 4000 comments for training and 2500 comments for testing. Next, they tested the performance of two classifiers: an SVM with a linear kernel which obtained an accuracy of 77.65%, and LR whose accuracy was 73.76%.

Later, in 2017, Hammer [10] used a logistic LASSO regression to detect violent content in threads about minorities, immigrants and homosexuals in 24840 manually tagged sentences from YouTube comments. The classifier only showed an approximate rate of 10% of violent texts classified as non-violent, and 5% of non-violent text classified as violent.

Also in 2017, Eshan and Hasan [11] classified Bengali text from Facebook comments with abusive content using TF-IDF, BoW and CountVectorizer along with the classifiers Random Forest (RF), multinomial NB, and SVM with different kernels: Linear, Radial Basis Function (RBF), Polynomial and Sigmoid. A TF-IDF encoder combined with a SVM with linear kernel achieved the best performance.

Recently, Deep Learning techniques [12] have also been used for NLP. Chu et al. [13] compared two Recurrent Neural Networks, Long-Short Term Memory (LSTM) and Convolutional Neural Networks (CNN), against two datasets of 150,000 Wikipedia comments where the highest accuracy (94%) was achieved by CNN with character embedding. Moreover, Badjatiya et al. [14] achieved a F1 score of 93% detecting sexist and racist tweets over a dataset with 16,000 samples using a CNN with a random embedding.

[2] http://archive.ics.uci.edu/ml/index.php
[3] https://kaggle.com

3 Methodology

In this work, we compared two different schemes for text classification: one based on different Machine Learning methods, and the other one based on a Deep Learning technique. On the one hand, we combined two text encoders (see Subsection 3.1) and three different classifiers (see Subsection 3.2) representing the Machine Learning methods. On the other hand, Deep Learning, we chose a deep neuronal network classifier.

For any of the two schemes, we need to perform a preprocessing on the text of the datasets. First, we eliminated start-line words and tags from markup languages like HTML. Next, we separated punctuation symbols such as dots or commas to facilitate the recognition of text elements. After that, we transformed the text elements such as words, spaces and punctuation symbols into numerical values that can be used by a learning algorithm. Finally, we encoded the text into a vector of numbers, i.e. the process of *vectorization* or *encoding*.

3.1 Encoding Techniques

In *Term Frequency - Inverse Document Frequency (TF-IDF)*, each term has a weight given by the product of two factors [15]: TF and IDF as shown in (1). TF refers to the frequency of appearance of a word in the text, and IDF is a measure of the amount of information provided by the word, i.e., how common is the word in the considered text.

$$\text{TF-IDF} = \text{TF} \times \text{IDF} \tag{1}$$

Within the framework of *Bag of Words (BoW)*, the word codification is based on the use of a dictionary previously generated with the documents used for training. Given a text, the algorithm expresses it in a form of a vector where each of its components correspond with a word of the dictionary, and it records the number of times that this word appears on the text [16] disregarding the grammar and the word order of the original text. This process can be visualized on Fig. 2.

3.2 Classifiers

Logistic Regression (LR) was proposed by Cox [17] and it is used to classify binary data, i.e., belonging to a two-class scheme, using a linear combination of the variables used to characterize the samples. The basic idea behind the model consists in fitting a logistic function with the samples of the training data expressed as points in a bi-dimensional plane where the X coordinate expresses the result of a linear combination of the characteristics and the Y coordinate a "0" or "1" depending on the class of the sample. After the training, for each sample to be classified the same linear combination of characteristics as in training is calculated, then the fitted logistic function is used to calculate the corresponding y value from the linear combination and this y value is considered as the probability of belonging to the class labelled as "1".

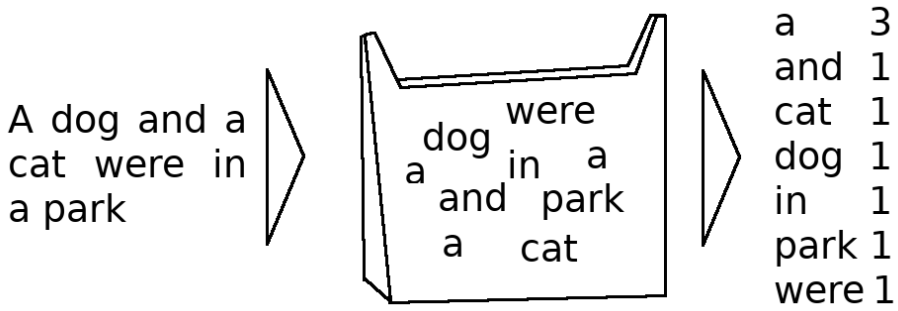

Fig. 2. Representation of the BoW encoding model

Support Vector Machine (SVM) classifier was proposed in 1995 by Cortes and Vapnik [18] and it consists on representing the data samples in a n-dimensional space where n is the number of characteristics used to describe the data, and then find the hyperplane that separates the two classes of the dataset with the largest margin, as shown in Fig. 3.

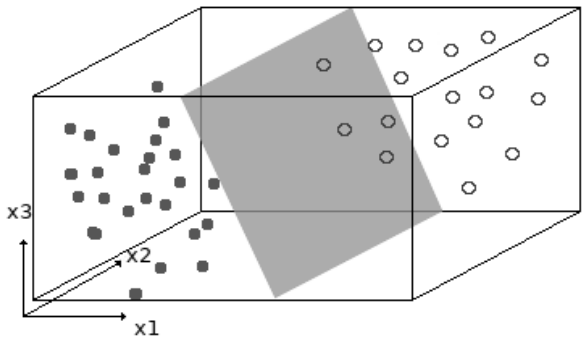

Fig. 3. Visualization of the hyperplane generated by a SVM model separating the samples of two classes represented as filled and empty dots, and assuming that their characteristics can be represented on a tridimensional space.

Finally, *Naïve Bayes (NB)* proposed by McCallum [19] is based on the Bayesian theorem: it considers that each of the characteristics of a sample contributes independently to the probability of the sample to belong to a class. Therefore, the presence or absence of a particular characteristic is not related to the presence or absence of any other characteristic.

3.3 StarSpace

StarSpace is a general-purpose deep neural model proposed by Wu et al. [20]. The model learns how to represent different encoded entities in a common linear space and then compares them against each other. Because this is a general scheme, it can be used in a wide variety of tasks[4]:

1. *PageSpace user / page embedding*: Used to recommend which Facebook pages may interest someone to follow based on the tracking of others.
2. *DocSpace document recommendation*: Recommendation of web pages based on the "like" history and clicks of a user.
3. *GraphSpace: Link Prediction in Knowledge Bases*: Map between entities and relationships in Freebase[5].
4. *SentenceSpace: Learning Sentence Embedding*: Given the coding of a sentence, it tries to find other semantically similar ones.
5. *ArticleSpace: Learning Sentence and Article Embedding*: Given the coding of a sentence, it tries to search relevant articles.
6. *ImageSpace: Learning Image and Label Embedding*: Learn the relationship between images and other entities.
7. *TagSpace word / tag embedding*: Classic classification of short texts.

In this work, we use the algorithm to embed the text and classify its content automatically. The algorithm can be configured with a Hit @ n accuracy, i.e., in the classification of a sample the model indicates the n most probable classes. In this work we use a Hit @ 1 accuracy, i.e., the model only indicates the most probable class in the classification of a sample.

4 Experimental Results

4.1 Experimental Settings

This work has been carried out in a computer environment with an Intel Core i5 CPU (1.3 GHz) and 8 GB of RAM. The software was developed on Python using the modules Anaconda[6], Scikit-Learn[7], Pandas[8] and Numpy[9]. The classic Machine Learning classifiers provided by the used modules were configured as follows:

1. The LR and SVM classifiers were configured to automatically adjust the weights of the classes as inversely proportional to class frequencies in the training data.

[4] https://github.com/facebookresearch/StarSpace
[5] https://developers.google.com/freebase/
[6] https://www.anaconda.com/distribution/
[7] https://scikit-learn.org/stable/
[8] https://pandas.pydata.org/
[9] http://www.numpy.org/

2. In SVM, we chose a linear kernel, as well as it was done in similar works of the related literature.
3. We chose a multinomial NB classifier since the input data contained discrete features, such as word counts.

4.2 Datasets

In our experiments we used two public datasets manually tagged by ten human collaborators from the Wikipedia Detox Project[10]: the Aggression content and the Attack content ones. Both datasets contain approximately 150,000 comments dual tagged by each operator with an integer value from 3 (i.e., peaceful comment) to -3 (i.e., very aggressive comment), and with a binary value indicating if the comment is inappropriate or not, as shown on Table 1. In the case of the Attack dataset, instead of the binary tag it presented a label indicating if the comment was classified as mentioned, received, attack to third parties or another attack; and we simplified this information into a binary tag informing whether the comment was an attack or not. Table 2 shows examples of violent and non-violent comments of both datasets.

Table 1. Description of the Attack and Aggression datasets. Amount of comments (violent vs. non violent.) in both datasets

Dataset	Violent	Not Violent
Attack	13590	102274
Aggression	7498	58452

Table 2. Examples of violent and non-violent comments on the Attack and Aggression datasets

	Attack	Aggression
Violent	- People as stupid as you should not edit Wikipedia! - NO! im not gunna sign my posts you ass! - Fuck you and Fuck your mom. And her dog.	- Charles, you are a real fag aren't you? siding with chinks?? - You are a raging faggot. Kill yourself. - Please take time today to kill yourself. We would all enjoy it.
Non-violent	- Thank you for your contribution, you did a great job! - I think Mac mini is just a ordinary desktop in a small case. - For your own safety, please do not post personal information.	- Thank you for reminding me about my signature - I think you're a bit late with that last post. - Correct, and noted in article

[10] https://meta.wikimedia.org/wiki/Research:Detox

The datasets are already divided into a training subset and a testing subset with 75% and 25% of the data respectively, and we discarded the comments where the standard deviation of the operators was higher than 1.25 since we consider that a larger deviation in a system of scoring between -3 and 3 indicates a misleading judgment of the character of the comment. After this, in our experiments, we only consider the binary tag, i.e., if the comment is appropriate or not.

4.3 Results

For each of the two Wikipedia detox datasets, we trained the seven evaluated methods with the training subset, i.e., 75% of the data, and we assessed them on the test subset. We presented the achieved accuracy of each model on Table 3.

Table 3. Accuracies achieved by the seven tested methods on the Attack and the Aggression datasets of the Wikipedia Detox Project

Method	Attack	Aggression
TF-IDF + LR	0.922	0.923
TF-IDF + SVM	0.907	0.905
TF-IDF + NB	0.931	0.931
BoW + LR	0.919	0.917
BoW + SVM	0.898	0.899
BoW + NB	0.926	0.927
StarSpace	**0.938**	**0.937**

The results show that the achieved accuracy of each classifier is similar in both datasets. StarSpace achieved the best scores on the Attack and Aggression datasets, with an accuracy of 0.938 and 0.937, respectively. Among the tested Machine Learning methods, the best performance was achieved by the combination of TF-IDF and NB with an accuracy of 0.931 on each of the two datasets, only behind of the Deep Learning classifier by less than 1% in both datasets.

We can also observe that considering the six combination methods, our results are higher when using TF-IDF than when using BoW for each of the three classifiers (LR, SVM and NB). We consider two reasons behind this event. First, BoW relies on a dictionary, which means that if there are offensive words present in the comments but not in the dictionary then this method will fail to encode those violent words. However, TF-IDF does not rely on a previous knowledge on the words and thus is capable to considerate new terms. Second, TF-IDF gives each word a score based on the information that it provides through the IDF term, unlike BoW which only accounts the number of times that each word in the predefined dictionary appears in the comment, and therefore cannot differentiate the importance of different words.

Moreover, either using TF-IDF or BoW, NB outperforms other approaches while the lowest performance results are achieved with the SVM classifier. The advantage of the NB algorithm is explainable due to the nature of the features provided by the encoders, such as the number of times that a word appears or the information that a single word provides, i.e., the features are mostly independent from each other which is the basic idea behind the NB classifier.

5 Conclusions and Future Work

The increase of inappropriate content on the Internet over the last years is forcing the development of new tools to filter it. In this work, we evaluated the accuracy of the combination of two encoders with three Machine Learning classifiers, and a Deep Learning model to detect violent content on two datasets from the Wikipedia Detox Project: the Attack and the Aggression ones. We tested six classic combination methods resulting from selection one of two encoders, i.e. TF-IDF and BoW, and one out of three classifiers, i.e. Logistic Regression, Support Vector Machine and Naive Bayes. Additionally, these six methods were compared with the Deep Learning classifier StarSpace, developed by Facebook.

The highest scores were achieved by StarSpace in both datasets, with an accuracy of 0.938 on the Attack dataset and of 0.937 on the Aggression one. However, these results are only less than 1% higher than the achieved by the combination of the simple algorithms TF-IDF and Naive Bayes, which obtained an accuracy of 0.931 in both datasets.

The experimental results achieved in this work suggest that we can apply these models to develop real filters for social networks which have minors as potential users, and other areas where this type of comments is not relevant for the reader, such as YouTube where we can find a large number of comments without regulation.

In future works, we will try to improve the obtained results by developing new Deep Learning models. In addition, we will explore the use of these techniques to the detection of other kinds of inappropriate contents in text, e.g. sexual, terrorism-related, hatred contents, etc.

References

1. Hussainalsaid, A., Azami, B. Z., and Abhari, A.: Automatic classification of the emotional content of URL documents using NLP algorithms. Proceedings of the 18th Symposium on Communications & Networking, 56–59 (2015)
2. Chin, H., Kim, J., Kim, Y., Shin, J., and Yi, M. Y.: Explicit Content Detection in Music Lyrics Using Machine Learning. IEEE International Conference on Big Data and Smart Computing, 517–521 (2018)
3. Duarte, N., Llanso, E., and Loup, A.: Mixed Messages? The Limits of Automated Social Media Content Analysis. FAT, 106 (2018)
4. Mironczuk, M. and Protasiewicz, J.: A Recent Overview of the State-of-the-Art Elements of Text Classification. Expert Systems with Applications, 106 (2016)

5. Bui, D. D. A., Del Fiol, G. and Jonnalagadda, S.: PDF text classification to leverage information extraction from publication reports. Journal of Biomedical Informatics, 61, 141–148 (2016)
6. Chen, J., Huang, H., Tian, S. and Qu, Y.: Feature Selection for Text Classification with NaïVe Bayes. Expert Syst. Appl., 36(3), 5432–5435 (2009)
7. Rogati, M. and Yang, Y.: High-performing Feature Selection for Text Classification. Proceedings of the Eleventh International Conference on Information and Knowledge Management, 659–661 (2002)
8. Diab, D. M. and Hindi, K.: Using Differential Evolution for Fine Tuning Nave Bayesian Classifiers and its Application for Text Classification. Applied Soft Computing, 54 (2016)
9. Chavan, V. and Shylaja, S.: Machine learning approach for detection of cyber-aggressive comments by peers on social media network. International Conference on Advances in Computing, Communications and Informatics, 2354–2358 (2015)
10. Hammer, H.: Automatic Detection of Hateful Comments in Online Discussion. Industrial Networks and Intelligent Systems, 164–173 (2017)
11. Eshan, S. and Hasan, M.: An application of machine learning to detect abusive Bengali text. International Conference of Computer and Information Technology, 1–6 (2017)
12. LeCun, Y., Bengio, Y. and Hinton, G.: Deep Learning. Nature, 521, 436–444 (2015)
13. Chu, T., Jue, K., and Wang, M.: Comment abuse classification with deep learning. Stanford University (2016)
14. Badjatiya, P., Gupta, S., Gupta, M., and Varma, V.: Deep learning for hate speech detection in tweets. International Conference on World Wide Web Companion, 759–760 (2017)
15. Aizawa, A.: An information-theoretic perspective of tfidf measures. Information Processing & Management, 39(1), 45–65 (2003)
16. Harris, Z.: Distributional structure. Word, 10(2-3), 146–162 (1954)
17. Cox, D.: The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B, 20(2), 215–232 (1958)
18. Cortes, C., and Vapnik, V.: Support-vector networks. Machine learning, 20(3), 273–297 (1995)
19. McCallum, A., and Nigam, K.: A comparison of event models for naive bayes text classification. AAAI-98 workshop on learning for text categorization, 752(1), 41–48 (1998)
20. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., and Weston, J.: Starspace: Embed all the things!. AAAI Conference on Artificial Intelligence, 5569–5577 (2018)