International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 591–603
DOI: https://doi.org/10.2991/ijcis.d.200519.003; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Machine Learning Techniques for the Detection of Inappropriate Erotic Content in Text

Gonzalo Molpeceres Barrientos[1], Rocío Alaiz-Rodríguez[1,*], Víctor González-Castro[1], Andrew C. Parnell[2]

[1] *Department of Electrical, Systems and Automation Engineering, Universidad de León Campus de Vegazana s/n, León, Spain*
[2] *Hamilton Institute, Maynooth University, Maynooth, Ireland*

ARTICLE INFO

Article History
Received 03 Mar 2020
Accepted 05 May 2020

Keywords
Inappropriate content
Machine learning
Text classification
Natural language processing
Text encoders

ABSTRACT

Nowadays, children have access to Internet on a regular basis. Just like the real world, the Internet has many unsafe locations where kids may be exposed to inappropriate content in the form of obscene, aggressive, erotic or rude comments. In this work, we address the problem of detecting erotic/sexual content on text documents using Natural Language Processing (NLP) techniques. Following an approach based on Machine Learning techniques, we have assessed twelve models resulting from the combination of three text encoders (Bag of Words, Term Frequency-Inverse Document Frequency and Word2vec) together with four classifiers (Support Vector Machines (SVMs), Logistic Regression, k-Nearest Neighbors and Random Forests). We evaluated these alternatives on a new created dataset extracted from public data on the Reddit Website. The best performance result was achieved by the combination of the text encoder TF-IDF and the SVM classifier with linear kernel with an accuracy of 0.97 and F-score 0.96 (precision 0.96/recall 0.95). This study demonstrates that it is possible to detect erotic content on text documents and therefore, develop filters for minors or according to user's preferences.

1. INTRODUCTION

People's digital habits have changed drastically in a very short time. The number of homes with Internet broadband connection has rapidly increased and the number of frequent Internet users (i.e., those who connect at least once a week) has also increased [1].

Nowadays, children have access to the Internet on a regular basis either for entertainment or for academic purposes. According to a Eurostat's recent survey about Internet access and use statistics—households and individuals [2], the Internet activities that most young people carry out are sending/receiving instant messages and emails, participating in social networks, listening to radio, watching video content from sharing services (e.g., Youtube) and finding information about goods and services.

Just like the real world, the Internet has many unsafe locations where kids may be exposed to harmful content. In a recent report [3], the online primary risks identified for young people are (1) interaction with strangers (i.e., unwanted friend requests and sexual, erotic or offensive messages) and (2) inappropriate content in the form of violence, hatred, sexual content and bullying.

As an attempt to prevent children to access inappropriate content, parents may use parental control applications, which include the possibility to manage the apps a minor is using, monitor his/her connection time and schedule, monitor the contents a minor is receiving and sending, etc. To the best of our knowledge, the protection implemented by these parental control tools are based on lists of categories or of specific websites whose access is forbidden or allowed to minors. These lists are created and kept up to date manually by the company that offers the tool, or by the adult managing it. In addition, they allow the adults managing the parental control tool to access the history of web browsing, watched videos, apps the minors have used, etc., in order to supervise what contents they have accessed to. Therefore, this kind of applications have some drawbacks:

1. Banning certain websites to be accessed does not guarantee that minors will not access inappropriate contents. For example, they may find unexpected ads with sexual content in a website whose access is permitted, or inappropriate ads in an app which is suitable for children [4].
2. Manually keeping up-to-date the blacklists or whitelists requires a huge amount of time and human resources, i.e., it is necessary to browse the websites and decide whether it contains inappropriate content or not. Moreover, this is a subjective decision and, therefore, might be prone to errors.
3. The parents or legal tutors can supervise the information about the usage of the minor's device, which is a huge help to detect some kinds of behaviors. However, the access to such information always takes place after the minor has accessed the content

[*] *Corresponding author. Email: rocio.alaiz@unileon.es*

(inappropriate or not). Moreover, it is not sure that the adult can carry out such supervision on a regular and frequent basis or, if he/she does, might not notice access to inappropriate content since, as well as any manual inspection, it is susceptible to errors.

Additionally, social networks, online forums and e-commerce sites also challenge the moderation of the content generated by the users. When it comes to user-generated text in these media, however, the daily amount of comments about any topic is so impressive that conducting human moderation successfully is not feasible. There is a high demand on tools to ease the repetitive, burdensome and time-consuming task of the human moderator in order to detect inappropriate, harmful or illegal content [5,6]. Natural Language Processing (NLP) techniques can be used for this purpose [7].

In this work, we explore the use of Machine Learning techniques applied to text classification to detect erotic content in text. We have assessed twelve models resulting from the combinations of three text encoders (Bag of Words (BoWs), Term Frequency-Inverse Document Frequency (TF-IDF) and Word2vec) together with four classifiers (Support Vector Machines (SVMs), Logistic Regression (LR), k-Nearest Neighbors and Random Forests (RFs)). We evaluated these methods creating a new dataset extracted from public data on the Reddit website.

The rest of the paper is organized as follows: Section 2 describes the text classification problem and related work to address the problem of detecting inappropriate texts. Next, we present the methodology used in this work in Section 3. Experimental evaluation is shown in Section 4.6. Finally, Section 5 summarizes the main conclusions and presents future lines of work.

2. TEXT CLASSIFICATION

The amount of information (i.e., text, images, audio and video) available on the Internet has grown exponentially over the last years, making the problem of information overload in multimedia environments to be now a fact. However, time and capabilities of users are not enough to process all this data. This has given rise to a high interest in the development of automatic text classifiers. These systems were mainly used at first for information retrieval [8,9] and later on for information filtering [10], text categorization [11,12], recommendation systems [13], sentiment analysis [14] and document summarization [15], among others. Likewise, these techniques have been used in many domains such as medicine, engineering, psychology, business, etc [16].

Most text classification processes can be divided into the following five stages (see Figure 1): (a) data acquisition and labelling, (b) preprocessing, (c) feature extraction or text encoding, (d) dimensionality reduction, (e) classifier training and (f) evaluation. Different studies tackle different problems that arise in this field. This study is centered on phases (c), (e) and (f), i.e., text encoding and classifier developing.

2.1. Detecting Inappropriate Text Content

Recently, the topic of detecting inappropriate content (i.e., in videos, images or text documents) has attracted an increasing interest in the machine learning field. In particular, there has been several attempts to filter inappropriate text content with NLP techniques.

In [17] text mining techniques based on combining simple (binominal) classifiers are proposed to block inappropriate web content. Some techniques of web sites analysis that provide information in different languages are also suggested.

The problem of identifying inappropriate query suggestions in search engines as well as detecting inappropriate language in users' conversations in messengers has been tackled in [18]. In this work, the authors propose deep learning architectures that, applied to real-world search queries and conversations, significantly outperform both pattern-based and other hand-crafted feature-based baselines.

Other proposals evaluate inappropriate comments to news [19]. In particular, the value of text distortion is assessed in this context resulting in an improvement in performance.

Inappropriate comments, however, may come in the form of aggressive, sexual, hate, gender, religious or racial offensive comments. In order to tackle the problem of inappropriateness of user-generated content on the Internet, more specialized models are being developed. Thus, there are several works that specifically address the detection of hate speech [20–26]. For instance, Hammer [22] used a LR model to detect violent content in threads about minorities, immigrants and homosexuals in 24840 manually labelled sentences from YouTube comments. The classifier only showed an approximate error rate of 10% within the violent text category whereas only 5% of nonviolent texts were classified as violent.

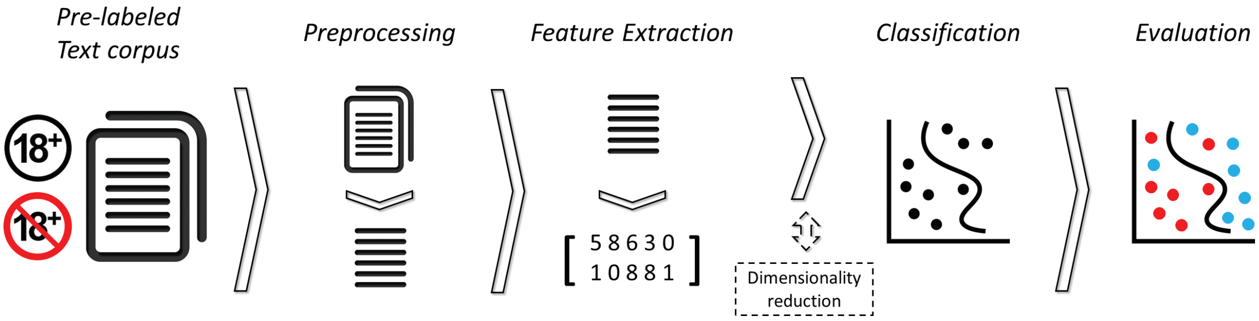

Figure 1 Typical text classification pipeline.

Others tackle the problem of cyberbullying detection on Twitter [27–32]. Thus, Chen *et al.* [29] compared two pre-trained word embedding techniques (i.e., Glove and Word2Vec) for text encoding as well as well-known techniques like TF-IDF. It turned out that the simple TF-IDF outperformed the more complex word embedding approaches. Murali *et al.* [30] evaluated the Naïve Bayes and RF classifiers using features derived from Twitter such as activity, user and tweet contents.

Regarding violent content, Merayo *et al.* [33] address this problem assessing several methods with two publicly available datasets from the Wikipedia Detox Project: Attack and Aggression. The authors evaluated six classic methods resulting from the combination of one out of two encoders (i.e., TF-IDF and BoWs) and one out of three classifiers (i.e., LR, SVMs and Naïve Bayes) that were additionally compared with the Deep Learning model StarSpace, developed by Facebook. Accuracy over 93% suggests that these models can be applied to develop real filters for social networks.

There is, however, little previous work specifically devoted to the detection of explicit sexual-erotic comments in text. Nonetheless, there is previous work that tackle other types of closely related problems. Thus, Narayanan *et al.* [34] propose a methodology to filter adult content in order to protect minors.

In this work, we tackle the problem of identifying user-generated sexual-erotic comments on social media with machine learning techniques. We will evaluate whether or not it is possible to train a classification model that automatically filters this kind of comments successfully. In that case, it would be possible to develop filters for minors or any user who would like to block this content.

3. PROPOSED METHODOLOGY

NLP is a large and multidisciplinary field that can be defined as the automatic processing of human languages. Basically, any practical application that makes use of text is a candidate for NLP and its success is largely driven by the advances in the machine learning field. These text processing tasks include classification, translation, structured prediction and sequential decision among others. This work focuses on text classification, i.e., the process of assigning categories to text according to its content. It applies to a wide variety of tasks like spam detection, sentiment analysis, detection of hate speech, cyberbullying or the detection of inappropriate content that we tackle in this paper.

The detection of inappropriate erotic/sexual content is a very challenging NLP problem. This process takes a considerable time to perform manually considering the vast amount of information published on social media. The development of fast and efficient tools for the automated discovery of this content becomes crucial to protect minors on the Internet. The goal of this work is to explore how the detection of sexual erotic content can benefit from the use of NLP and Machine Learning techniques.

In this section, we present the methodology used to build the proposed sexual/erotic text classifier. The typical scheme of a supervised text classifier has two main components: text representation to encode the input text samples into feature vectors and a classifier that categorizes these feature vectors into one of the two categories. Improvements and advances in computer hardware have made possible to use deep learning based classifiers with remarkable performance. Nonetheless, it has been seen that for some text classification problems classical machine learning techniques also give similar performance results with less training time and hardware requirements. In this paper we assess this latter approach.

We address the task of detecting sexual erotic text assessing different frameworks for text categorization. The classification models evaluated in this work result from the combination of different text encoders (see Section 3.1) together with some classical machine learning techniques have shown good performance on text classification tasks [35,36].

For any of the proposed schemes, we consider a preprocessing step that includes tokenization, lemmatization and removal of stop words, among others.

Consider a text dataset $\mathcal{D} = \{(\mathrm{x}_i, d_i), i = 1, \dots, N\}$ with N documents or text segments x and a class label d associated with each sample. Documents in $\mathcal{D}$ can be considered as a set of words dependent of one another, and they need to be represented as numerical vectors that, hopefully, reflect the relationships among words. Next, we present different text encoding techniques for this task.

3.1. Encoding Techniques

There are two main approaches to tackle the text enconding task: (a) The well known Vector Space Model (VSM) approach [37–40] and (b) the recent proposals based on word embedding [16,41–43]. The VSM approach converts a text document into a numerical vector whereas the word embedding approaches turn individual words into numerical vectors with arbitrary dimensionality.

Within the VSM framework, each vector component is a measure of the importance of the corresponding term in the represented document. Several techniques have been proposed to compute the weights (i.e., the vector components) for the different words in a given document. Term frequency (TF) (also known as BoWs) and TF-IDF are one of the most well known weighting techniques (see Sections 3.1.1 and 3.1.2). In this case, each instance x_i is represented as a p-dimensional vector $\mathrm{x}_i = [x_{i1}, x_{i2}, \dots x_{ip}]$ where each component x_{ij} represents the importance of a given word f_j for sample i.

Besides, over the last few years there has been a high interest in word embedding. Word2Vec [41], Glove [43] or FastText [44] are just some examples of popular word embedding-based approaches. The word vectors are obtained by training the algorithm with a text corpus. After such training has finished, each word in the corpus is represented by a numerical q-dimensional vector. The value of q is chosen at the time of training and it is smaller than the number of unique words in the corpus, i.e., $q << p$.

Since every word in a document has a representation in the same q-dimensional space, then a p-dimensional document vector (i.e., a set of words) can be represented as a vector in that same q-dimensional space. The original $p \times 1$ document vector x can be rewritten as a $q \times 1$ vector x^* according to

$$\underbrace{\mathbf{x}^*}_{q\times 1} = \underbrace{\mathbf{W}}_{q\times p}\underbrace{\mathbf{x}}_{p\times 1}, \tag{1}$$

where the matrix $\mathbf{W}$ contains the embeddings for the unique words in the text corpus.

In this work, we assess two VSM approaches, i.e., BOW and TF-IDF, and the word embedding Word2vec.

3.1.1. Bag of Words

The BoWs scheme, also known as TF [39] is one of the simplest feature weighting techniques. This technique is used in domains other than text classification such as computer vision [45] or document categorization.

BoW characterizes each text by means of a vector which represents the frequency with which a word in a dictionary appears on the text. That dictionary may be generated by means of a training set of texts. This vector does not reflect grammar, the word order in the text or the semantic relationship between words in the text.

An example of this technique is depicted in Figure 2, were a phrase is codified based on a previously generated dictionary.

3.1.2. Term Frequency-Inverse Document Frequency

TF-IDF is one of the most commonly used term weighting schemes in nowadays information retrieval systems [46,47]. The term TF refers to the frequency of appearance of a word in the text, whereas IDF is a term weighting function that measures how important each word is to the text document. IDF was first proposed by Karen Spärck Jones [40].

Let the vector of weights for document D be $\mathbf{w}_D = (w_{1,D}, w_{2,D}, \dots, w_{p,D})$. The weight of the i^{th} term in document D is defined as

$$w_{i,D} = tf_{i,D} \log \frac{N}{df_i}, \tag{2}$$

where $tf_{i,D}$ is the frequency of appearance of term i in D, df_i is the number of documents in which that term appears and N the total number of documents in the document set.

The intuition behind this weighting technique is that (a) the more frequently a word appears in a document, the more representative it is for that document but (b) the more documents the word appears in, the less informative it is [48].

3.1.3. Word2vec

Word2vec is a method, proposed in 2013 by T. Mikolov and colleagues [42,49], to efficiently create word embeddings that has gained a lot of attention in the past few years. Basically, the Word2vec model creates a vocabulary from the training text data and learns dense word embeddings, i.e., the representation of a word, i.e., useful for prediction of other words in the sentence. This vector representation can be subsequently used in machine learning applications like text categorization. The word vectors generated with Word2vec seem to capture many linguistic regularities as long as the models are trained with large enough datasets [42]. Next, we briefly describe this method and refer the interested reader to [41,50] for details about the implementation.

Word2vec generates word embeddings using a shallow neural network, i.e., a fully connected neural network with a single hidden layer. The input layer has as many neurons as words in the training vocabulary. The size of the hidden layer is the dimensionality of the feature space and the weights adjusted during the learning stage are then used as the embeddings. The size of the output layer is the same as that of the input layer.

There are two main learning algorithms in word2vec: Continuous Bag-Of-Words (CBOW) and Skip-Gram (SG). CBOW tries to predict the target word (i.e., the center word) given some context words (i.e., surrounding words), whereas SG predicts context words based on the target word. Some studies [41,42,49,50] suggest that SG is slower but better for infrequent words while CBOW is faster. An illustration of the architecture of both techniques can be seen in Figure 3.

For instance, considering a simple sentence, “the greedy dog story writing,” there can be pairs of (context, target word) where if we consider a context window of size 1, we have examples like ([greedy, story], dog), ([the, dog], greedy), ([dog, writing], story) and so on. Thus, with CBOW, the model is trained to predict the target word based on the surrounding context words and the other way round in the case of SG.

The training algorithms could be either hierarchical softmax or negative sampling. The former tends to be better for infrequent words while the latter is better for frequent words and also better with low dimensionality of the feature space [41,42,49,50]. This leads to a dense word embedding with lower dimensionality than the traditional sparse vector models.

4. EXPERIMENTAL RESULTS

In this work, we have built models to detect written inappropriate—i.e., specifically erotic—content. We have assessed several encoding techniques together with different classifiers. The evaluation is conducted in terms of the classification performance.

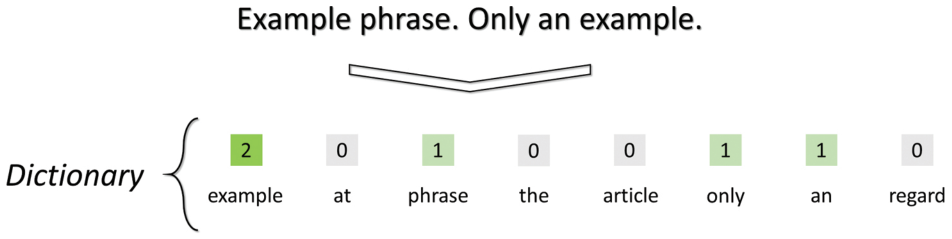

Figure 2 Example of feature extraction using Bag of Word (BoW).

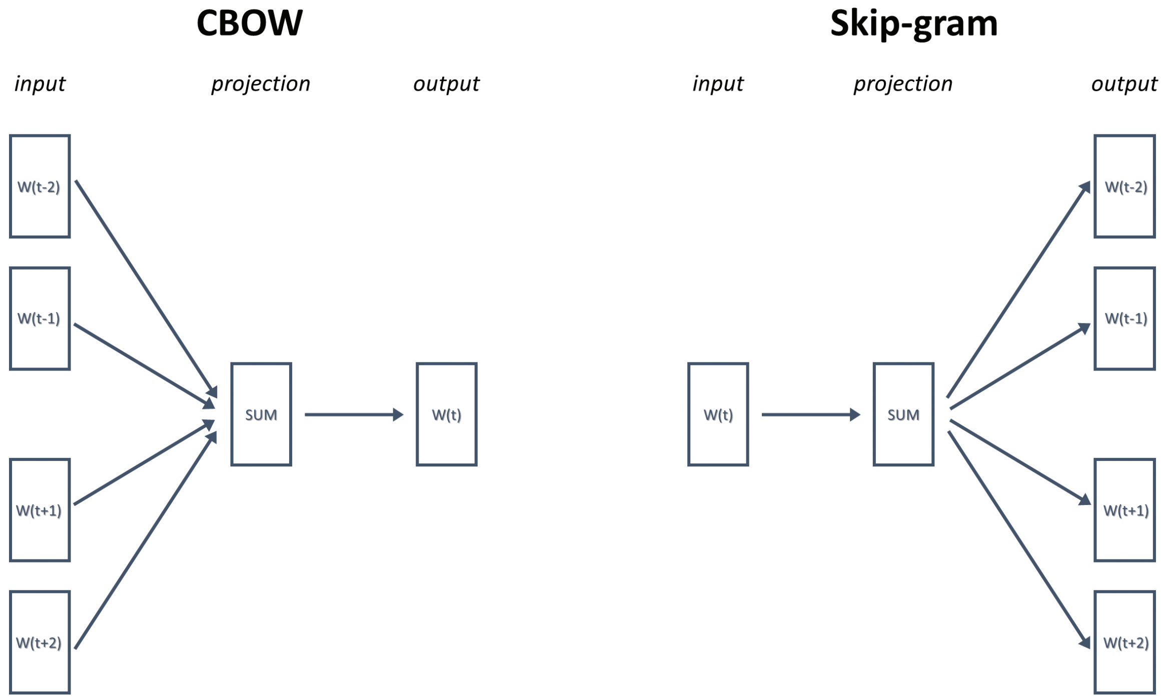

Figure 3 Architecture of Word2Vec models: CBOW and Skip-Gram. The window size in both cases (i.e., the number of surrounding words before and after the target word is set to 2).

In the next subsections we present the dataset used in the assessment, the text preprocessing and feature extraction techniques and the classifiers employed. Finally we explain the performance metrics we have used (i.e., precision, recall, F-score and accuracy) and the obtained results.

4.1. Dataset

The experiments were carried out using a filtered version of the Reddit public dataset.[1]

The experiments were carried out using a filtered version of the Reddit public dataset.[2]

The original dataset contained 98,753,936 files, with comments posted to the Reddit website that covered different topics, i.e., Subreddits. After filtering by topic the texts published during the months of January and July of the years 2015–2018, we obtained a dataset with 111,834 files: 50,921 erotic and the remaining 60,913 neutral. The selection of Subreddits that were chosen to divide the documents as erotic or neutral is shown in Table 1.

Table 1 List containing the number of files for each subreddit.

Category	Subreddit	Number in Files
Erotic	sex	25640
	gonewildstories	12943
	sluttyconfessions	1828
	eroticliterature	528
	sexstories	426
	erotica	332
	hotpast	195
	eroticwriting	29
Neutral	explainlikeimfive	18760
	pets	11053
	paranormal	10176
	outoftheloop	8133
	getmotivated	8064
	hfy	1865
	ask	1619
	stories	545
	literature	440
	write	199
	kitchen	24
	series	21
	workstories	14

4.2. Preprocessing

Prior to classification, the documents must be preprocessed—i.e., the original text is transformed into a set of tokens. Afterward, a set of features must be extracted from such preprocessed text (see Section 3), which are thereafter passed as input to the classification model.

In this work, we have carried out the following preproceessing steps:

1. Transformation of all text to lower case (e.g., “Rodrigo” → “rodrigo”).
2. Removal of everything, i.e., not a word and leave only one space between words (e.g., “this 5is incredible!!!!” → “this is incredible”).
3. Removal of stop-words, i.e., words that are so common in a language that they do not provide any meaning (e.g., “the,” “is,” “a,” etc.).

[1] https://files.pushshift.io/reddit/submissions/

[2] https://files.pushshift.io/reddit/submissions/

4. Reduction of the different lexical forms of the words to its root or verbs into their infinitive, i.e., lemmatization (e.g., "am," "are," "is" → "be," "different" → "differ")
5. Division of the phrases into tokens, i.e., the words.

The stop-words removal and tokenization processes were carried out with the help of the Natural Language Toolkit (NLTK)[3] library [51] while the lemmatization was carried out with the Spacy[4] library.

4.3. Feature Extraction

The feature extractor is the responsible to convert the tokens into numerical values that can be used by a learning algorithm. In this work, we chose three feature extraction methods to encode the preprocessed text (i.e., the tokens) into the data which can be processed by the classification model: BOW, TF-IDF and Word2Vec.

All of them were executed using n-grams, which means that the words (i.e., the tokens) were analyzed in groups of n. In this work, we used unigram (i.e., $n = 1$) and bigram (i.e., $n = 2$).

BOW and TF-IDF as well as the mentioned preprocessing methods (see Section 4.2) were implemented through the Scikit-learn[5] library. The minimum document frequency has been set to 3 (i.e., the documents with a document frequency less than 3 have been ignored).

Word2Vec was implemented using the Gensim[6] library. More specifically, we evaluated the CBOW model architecture with the following parameters: the dimensionality of the word embeddings was set to 500, the context window size (i.e., how many words before and after a given target word would considered as context) was set to 5, the number of epochs was set to 30 and the minimum document frequency was set to 3. For feature learning we used the training algorithm negative sampling, setting to 5 the number of negative samples. Finally, the threshold for randomly downsampling higher-frequency words was set to 0.001.

4.4. Classification Model

We have assessed different classifiers for the task of detecting inappropriate text: LR, k-Nearest Neighbors (kNNs), SVMs and RF. All of them have been implemented by means of the Scikit-learn library.

The performance of the *LR* classifier has been assessed for different values of the inverse of regularization strength. Such values have been 0.1, 1, 10 and 100.

Concerning *SVMs* [52], we have evaluated it with both a linear kernel and a Radial Basis Function (RBF) kernel. In the case of SVM with linear kernel, only the value of the inverse of the regularization strength, i.e., C, has been adjusted, testing the values of $C = \{0.1, 1, 10, 100\}$. On the other hand, when the RBF kernel has been used, we have combined the values of $C = \{0.1, 1, 10\}$ and $\gamma = \{0.1, 1\}$.

In the case of *KNNs*, we have assessed different numbers of nearest neighbors k, specifically $k = \{3, 5, 7\}$, and we have used the Euclidean distance.

Finally, the *RF* [53] model has been evaluated with 3, 10 and 30 trees. It has been trained with a maximum depth of the tree set to 5. Moreover, the RF has also been trained expanding the nodes until all leaves were pure or until all leaves contained less than two samples (i.e., maximum number not limited).

For the not specified parameters, we have taken the Scikit-Learn default values. Classifier generalization ability has been estimated using k-fold cross-validation technique, with $k = 5$.

4.5. Performance Metrics

We have assessed the performance of the models by means of accuracy, precision, recall and F-score. We have considered the erotic-sexual class as the positive class and the neutral class as the negative one.

The classification accuracy, i.e., the number of correctly classified comments over the total number of comments seen by the system, can be computed as shown in Equation (3).

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}, \tag{3}$$

where tp (true positives) represents the number comments with erotic-sexual content classified as sexual whereas tn (true negatives) is the number of neutral comments classified as neutral. The false positives (fp) refer to the comments classified as erotic-sexual that are actually neutral and fn (false negatives) are comments with erotic-sexual content categorized as neutral.

Precision is the fraction of correctly classified erotic comments over the number of items classified as erotic, as indicated by (4)

$$precision = \frac{tp}{tp + fp}. \tag{4}$$

Recall refers to the fraction of correctly classified erotic comments over the total number of erotic items (5)

$$recall = \frac{tp}{tp + fn}. \tag{5}$$

Finally, the F-score metric is defined as the harmonic mean of precision and recall and it can be computed as shown in (6)

$$F\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{6}$$

4.6. Results

The results of the experiments using the classifiers SVM with linear kernel, SVM with RBF kernel, LR, kNN and RF are presented in

[3] Library specialized in NLP in English. Source: https://www.nltk.org/

[4] Library specialized in advanced NLP. Source: https://spacy.io/

[5] Library specialized in Machine Learning for Python. Source: https://scikit-learn.org

[6] Library with topic modeling and NLP functions for Pyhton. Source: https://radimrehurek.com/gensim/index.html

Tables A.1–A.5, respectively (Appendix). On each table, the results achieved with the different feature extractors (see Section 4.3) are shown.

For the sake of clarity, we have summarized the best results of accuracy and the erotic F-score for each model, which are shown in Tables 2 and 3, respectively.

Tables 2 and 3 show the accuracy and F-score, respectively, achieved by each of the assessed combinations of classifier and feature extractor. Specifically, these are the highest among all the results obtained with different combinations of feature extractor options and classifier parameters (see Appendix). The three best results have been highlighted with a gradual green color.

It is noteworthy that TF-IDF is the text encoder that leads to the best performance results both in terms of accuracy and F-score.

Analyzing the results shown in Tables 2 and 3, we can conclude that the feature extractor that offers the best results is TF-IDF, except in the case of RF, in which Word2Vec shows the best performance.

Concerning the assessed classifiers, SVM with linear kernel shows the highest accuracy, i.e., 0.9712 with TF-IDF and 2-gram. Moreover, looking at Table A.1, not only its lowest accuracy (i.e., 0.9474 with Word2vec and 1-gram) is already quite good, but also it can be noticed that an improvement in the results can be appreciated as the regularization become stricter (i.e., the value of C is increased).

The performance of SVM with RBF kernel is quite remarkable. According to Table A.2, results obtained using TF-IDF are quite good (except with $C = 0.1$ and $\gamma = 0.1$), whereas with BOW its performance is always quite low. Using Word2Vec, however, the accuracy is very dependent of the parameters C and γ (e.g., 0.6385 with $C = 0.1$ and $\gamma = 1$ or 0.9659 with $C = 1$ and $\gamma = 1$). Since the kernel is a transformation of the set of features, a possible explanation might be that both BOW and Word2Vec have less compatible data with such transformation. An instance with generally negative results is SVM with RBF kernel, since although acceptable results can be seen in an extractor such as TF-IDF, the results are very negative in the case of using it together with BOW or Word2Vec, such as when adopts the values C = 0.1 and γ = 1. Since the kernel is a transformation of the set of the characteristics returned by the classifier, it can be con In the case of LR, the results are quite acceptable in all cases, although not as much as with SVM with linear kernel, being the best TF-IDF extractor closely followed by Word2Vec. As in SVM, the higher the parameter C, the better the performance.

Table 2 Accuracy comparison.

	BOW	TF-IDF	Word2Vec
SVM-linear	0.9642	0.9712	0.9601
SVM-RBF	0.8019	0.9707	0.9659
LR	0.9639	0.9687	0.9617
KNN	0.8211	0.9294	0.9159
RF	0.9218	0.9375	0.9493

BOW, Bag of Words; TF-IDF, Term Frequency–Inverse Document Frequency; LR, Logistic Regression; KNN, k-Nearest Neighbors (kNN); SVM, Support Vector Machine; RF, Random Forest; RBF, Radial Basis Function.

Table 3 Erotic F-score comparison.

	BOW	TF-IDF	Word2Vec
SVM-linear	0.9527	0.9621	0.9474
SVM-RBF	0.7402	0.9614	0.955
LR	0.9522	0.9587	0.9494
KNN	0.7213	0.9065	0.896
RF	0.8978	0.9166	0.9323

BOW, Bag of Words; TF-IDF, Term Frequency–Inverse Document Frequency; LR, Logistic Regression; KNN, k-Nearest Neighbors (kNN); SVM, Support Vector Machine; RF, Random Forest; RBF, Radial Basis Function.

The KNN classifier has the worst results in general, being especially negative in the case of BOW, specifically with 2-gram (see Table A.4), and improving with TF-IDF and Word2Vec, emphasizing that the more the number of neighbors k is increased, the better results.

Regarding RF, we can appreciate in Table A.5 that when the depth of the trees is equal to 5, its performance is quite bad when using BOW and TF-IDF, while Word2Vec performs quite acceptably, reaching a maximum Accuracy of 0.9351. As can be seen, results improve when no maximum depth is established and the number of trees is increased, being the best combination with Word2Vec and reaching a maximum of 0.9493 in Accuracy.

Finally, considering the use of n-grams, except in specific cases, there is not a significant difference when using 1-gram or 2-gram, although a general small improvement can be appreciated in the case of 2-gram. This could be expected, since when talking about natural language in a non-formal context, there could be different expressions or a jargon of its own that are better understood with 2-grams.

5. CONCLUSIONS AND FUTURE WORK

The vast amount of user-generated content on the Internet over the last few years makes the manual moderation of these texts, images and videos an unachievable task. There is a clear need of techniques to automate the detection of inappropriate content generated by users. In this work, we address the problem of detection of erotic-sexual comments or text posts on social media using machine learning techniques.

The aim of this study is the assessment of several text encoders (either based on VSM or word embeddings) together with different classification models to detect erotic-sexual content in texts. The experimental results were conducted with a dataset extracted from public data on the Reddit Website. The best performance result is achieved with a SVM classifier with a linear kernel using the TF-IDF technique as text encoder. The classification error is only 3% and precision and recall reach the values of 0.96 and 0.95, respectively. It is noteworthy that the simple TF-IDF feature weighting approach outperforms more complex ones based on word embeddings.

The experimental results achieved in this work suggest that applying machine learning techniques to this problem is a reliable approach that enables automatic moderation for this form of inappropriate content. These models can be used to develop real filters for social networks. Minors are potential users as well as other users for whom these comments are not relevant, such as YouTube where the user can find a large number of comments without regulation.

Exploring the performance of Deep Learning models as well as classifier ensembles becomes part of our future work. Additionally, given the high dimensionality of the feature space, we would like to analyze if these classification models benefit from the use the feature reduction techniques.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

All authors contributed to the work. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This research has been funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the authors, and the European Commission cannot be held responsible for any use which may be made of the information contained therein. Andrew Parnell's work was supported by a Science Foundation Ireland Career Development Award grant 17/CDA/4695 and an SFI centre grant 12/RC/2289_P2.

REFERENCES

[1] K. Pte, Digital 2019: Global Digital Overview, 2019. https://datareportal.com/reports/digital-2019-global-digital-overview

[2] Eurostat, Internet Access and Use Statistics – Households and Individuals, 2016. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Internet_access_and_use_statistics_-_households_and_individuals

[3] NSPCC, Net Aware Report 2017: "Freedom to Express Myself Safely": Exploring How Young People Navigate Opportunities and Risks in Their Online Lives, NSPCC, London, England, 2017. https://learning.nspcc.org.uk/research-resources/2017/net-aware-report-2017-freedom-toexpress-myself-safely

[4] Y. Chen, H. Xu, Y. Zhou, S. Zhu, Is this app safe for children?: a comparison study of maturity ratings on android and ios applications, in Proceedings of the 22nd international conference on World Wide Web, ACM, Rio de Janeiro, Brazil, 2013, pp. 201–212.

[5] C.-S. Li, G. Xiong, E.M. Tapia, New frontiers in cognitive content curation and moderation, APSIPA Trans. Signal Inf. Process. 7 (2018).

[6] M. Yar, A failure to regulate? The demands and dilemmas of tackling illegal content and behaviour on social media, Int. J. Cybersecur. Intell. Cyber. 1 (2018), 5–20.

[7] N. Duarte, E. Llanso, A. Loup, Mixed messages? The limits of automated social media content analysis, in FAT, New York City, NY, USA, 2018.

[8] S.K. Dwiv, C. Arya, Automatic text classification in information retrieval: asurvey, in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, ACM, Udaipur, India, 2016, p. 131.

[9] C. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Nat. Lang. Eng. 16 (2010), 100–103.

[10] W. Bruce Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice, vol. 520, Addison-Wesley Reading, 2010.

[11] H. Bunke, A. Kandel, A. Schenker, M. Last, Classification of web documents using graph matching, Int. J. Pattern Recognit. Artif. Intell. 18 (2004), 475–496.

[12] Z. Li, J. Huang, A text classification algorithm based on improved multidimensional–multiresolution topological pattern recognition, Int. J. Pattern Recognit. Artif. Intell. 13 (2019).

[13] C.C. Aggarwal, Content-based recommender systems, in Recommender systems, Springer, Cham, Switzerland, 2016, pp. 139–166.

[14] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: C. Aggarwal, C. Zhai (Eds.), Mining Text Data, Springer, Boston, MA, USA, 2012, pp. 415–463.

[15] A. Joshi, E. Fidalgo, E. Alegre, L. Fernández-Robles, SummCoder: an unsupervised framework for extractive text summarization based on deep auto-encoders, Expert Syst. Appl. 129 (2019), 200–215.

[16] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text classification algorithms: a survey, Information. 10 (2019), 150.

[17] I. Kotenko, A. Chechulin, D. Komashinsky, Evaluation of text classification techniques for inappropriate web content blocking, in 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IEEE, Warsaw, Poland, 2015, vol. 1, pp. 412–417.

[18] H. Yenala, A. Jhanwar, M.K. Chinnakotla, J. Goyal, Deep learning for detecting inappropriate content in text, Int. J. Data Sci. Anal. 6 (2018), 273–286.

[19] P. Bellan, C. Strapparava, Detecting inappropriate comments to news, in: C. Ghidini, B. Magnini, A. Passerini, P. Traverso (Eds.), International Conference of the Italian Association for Artificial Intelligence, Springer, Cham, Switzerland, 2018, pp. 403–414.

[20] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in Eleventh International AAAI Conference on Web and Social Media, Montreal, Quebec, Canada, 2017.

[21] L. Gao, R. Huang, Detecting online hate speech using context aware models, in Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2017), Varna, Bulgaria, 2017, pp. 260–266.

[22] H.L. Hammer, Automatic detection of hateful comments in online discussion, in: L. Maglaras, H. Janicke, K. Jones (Eds.), Industrial Networks and Intelligent Systems, International Conference on Industrial Networks and Intelligent Systems, Springer, Cham, Switzerland, 2016, pp. 164–173.

[23] D. Robinson, Z. Zhang, J. Tepper, Hate speech detection on twitter: feature engineering vs feature selection, in: A. Gangemi, *et al.* (Eds.), European Semantic Web Conference, Springer, Cham, Switzerland, 2018, pp. 46–49.

[24] N.D.T. Ruwandika, A.R. Weerasinghe, Identification of hate speech in social media, in 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, Colombo, Sri Lanka, 2018, pp. 273–278.

[25] H. Watanabe, M. Bouazizi, T. Ohtsuki, Hate speech on tTwitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection, IEEE Access. 6 (2018), 13825–13835.

[26] Z. Zhang, D. Robinson, J. Tepper, Detecting hate speech on twitter using a convolution-gru based deep neural network, in: A. Gangemi, *et al*. (Eds.), European Semantic Web Conference, Springer, Cham, Switzerland, 2018, pp. 745–760.

[27] S. Agrawal, A. Awekar, Deep learning for detecting cyberbullying across multiple social media platforms, in: G. Pasi, B. Piwowarski, L. Azzopardi, A. Hanbury (Eds.), Advances in Information Retrieval, European Conference on Information Retrieval, Springer, Cham, Switzerland, 2018, pp. 141–153.

[28] M.A. Al-garadi, K.D. Varathan, S.D. Ravana, Cybercrime detection in online communications: the experimental case of cyberbullying detection in the twitter network, Comput. Hum. Behav. 63 (2016), 433–443.

[29] J. Chen, S. Yan, K.-C. Wong, Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis, Neural Comput. Appl. (2018), 1–10.

[30] S.P. Murali, *et al*., Detecting cyber bullies on twitter using machine learning techniques, Int. J. Inf. Secur. Cyber. 6 (2017), 63–66.

[31] K. Nalini, L.J. Sheela, Classification of tweets using text classifier to detect cyber bullying, in: S. Satapathy, A. Govardhan, K. Raju, J. Mandal (Eds.), Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI, vol. 2, Springer, Cham, Switzerland, 2015, pp. 637–645.

[32] R. Zhao, A. Zhou, K. Mao, Automatic detection of cyberbullying on social networks based on bullying features, in Proceedings of the 17th international conference on distributed computing and networking, ACM, Singapore, 2016, p. 43.

[33] S. Merayo-Alba, E. Fidalgo, V. González-Castro, R. Alaiz-Rodríguez, J. Velasco-Mata, Use of natural language processing to identify inappropriate content in text, in: H. Pérez García, L. Sánchez González, M. Castejón Limas, H. Quintián Pardo, E. Corchado Rodríguez (Eds.), Hybrid Artificial Intelligent Systems, International Conference on Hybrid Artificial Intelligence Systems, Springer, Cham, Switzerland, 2019, pp. 254–263.

[34] B.K. Narayanan, S. Moses, M. Nirmala, *et al*., Adult content filtering: restricting minor audience from accessing inappropriate internet content, Educ. Inf. Technol. 23 (2018), 2719–2735.

[35] A. Genkin, D.D. Lewis, D. Madigan, Large-scale bayesian logistic regression for text categorization, Technometrics. 49 (2007), 291–304.

[36] G. Schohn, D. Cohn, Less is more: active learning with support vector machines, in Proceedings of the Seventeenth International Conference on Machine Learning, Citeseer, Stanford, CA, USA, 2000, p. 6.

[37] R. Baeza-Yates, B. Ribeiro-Neto, *et al*., Modern Information Retrieval, vol. 463, ACM Press, New York, NY, USA, 1999.

[38] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (1990), 391–407.

[39] Z.S. Harris, Distributional structure, Word. 10 (1954), 146–162.

[40] K.S. Jones, A Statistical Interpretation of Term Specificity and its Application in Retrieval, Taylor Graham Publishing, London, UK, 1988.

[41] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, arXiv preprint arXiv:1301.3781, 2013.

[42] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, 2013, pp. 746–751.

[43] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

[44] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017), 135–146.

[45] E. Fidalgo, E. Alegre, L. Fernández-Robles, V. González-Castro, Fusión temprana de descriptores extraídos de mapas de prominencia multi-nivel para clasificar imágenes, Rev. Iberoam. Autom. Inf. ind. 16 (2019), 358–368.

[46] S.M.H. Dadgar, M.S. Araghi, M.M. Farahani, A novel text mining approach based on tf-idf and support vector machine for news classification, in 2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE, Coimbatore, India, 2016, pp. 112–116.

[47] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in Advances in Neural Information Processing Systems, Montreal, Quebec, Canada, 2015, pp. 649–657.

[48] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (2002), 1–47.

[49] T. Mikolov, Q.V. Le, I. Sutskever, Exploiting Similarities among Languages for Machine Translation, arXiv preprint arXiv:1309.4168, 2013.

[50] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in Advances in Neural Information Processing Systems, Nevada, USA, 2013, pp. 3111–3119.

[51] X. Shu, R. Cohen, *et al*., Natural Language Toolkit (NLTK), 2010.

[52] V. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.

[53] L. Breiman, Random forests, Mach. Learn. 45 (2001), 5–32.

APPENDIX

Table A.1 Results of the SVM with linear kernel.

SVM with Linear Kernel			BOW		TF-IDF		Word2Vec	
			1-gram	**2-gram**	**1-gram**	**2-gram**	**1-gram**	**2-gram**
C = 0.1	Precision	Erotic	0.9635	0.9697	0.979	0.9701	0.9476	0.9494
		Neutral	0.9628	0.9576	0.9505	0.9498	0.9659	0.9643
	Recall	Erotic	0.9422	0.9335	0.9212	0.9207	0.9468	0.9441
		Neutral	0.9781	0.982	0.988	0.9826	0.9675	0.9685
	F-score	Erotic	0.9527	0.9512	0.9491	0.9447	0.9472	0.9467
		Neutral	0.9703	0.9696	0.9687	0.9658	0.9667	0.9664
	Accuracy		0.9642	0.9632	0.9619	0.9585	0.9599	0.9595
C = 1	Precision	Erotic	0.9398	0.967	0.9718	0.9687	0.9463	0.9477
		Neutral	0.9626	0.9589	0.9647	0.9693	0.9655	0.9645
	Recall	Erotic	0.9428	0.9359	0.945	0.9526	0.9485	0.9451
		Neutral	0.9634	0.9803	0.9835	0.9812	0.9683	0.9677
	F-score	Erotic	0.9413	0.9512	0.9582	0.9606	0.9474	0.9464
		Neutral	0.963	0.9695	0.974	0.9752	0.9669	0.9661
	Accuracy		0.9554	0.9631	0.9685	0.9701	0.9601	0.9592
C = 10	Precision	Erotic	0.9277	0.9668	0.9568	0.9668	0.9433	0.9406
		Neutral	0.9603	0.959	0.963	0.9722	0.9667	0.9699
	Recall	Erotic	0.9391	0.9359	0.9427	0.957	0.9485	0.9526
		Neutral	0.956	0.9801	0.9745	0.9797	0.9644	0.9616
	F-score	Erotic	0.9333	0.951	0.9496	0.9618	0.9459	0.9465
		Neutral	0.9581	0.9694	0.9686	0.976	0.9655	0.9657
	Accuracy		0.9493	0.963	0.962	0.971	0.9587	0.959
C = 100	Precision	Erotic	0.9259	0.9668	0.9492	0.9666	0.9117	0.9253
		Neutral	0.9596	0.959	0.9617	0.9727	0.9678	0.9656
	Recall	Erotic	0.9379	0.9359	0.941	0.9577	0.9541	0.9449
		Neutral	0.9549	0.9801	0.9698	0.9796	0.9451	0.9506
	F-score	Erotic	0.9317	0.951	0.945	0.9621	0.9319	0.9342
		Neutral	0.9572	0.9694	0.9657	0.9761	0.9561	0.9577
	Accuracy		0.9482	0.963	0.9584	0.9712	0.9474	0.9495

Table A.2 Results of the SVM with RBF kernel classifier.

SVM with RBF Kernel			BOW		TF-IDF		Word2Vec	
			1-gram	**2-gram**	**1-gram**	**2-gram**	**1-gram**	**2-gram**
C = 0.1	Precision	Erotic	1	0	0.9824	0.9802	0.9759	0.9743
		Neutral	0.621	0.6207	0.8692	0.7765	0.938	0.9405
$\gamma = 0.1$	Recall	Erotic	0.0016	0	0.7653	0.543	0.8993	0.9037
		Neutral	1	1	0.9914	0.9934	0.9862	0.9853
	F-score	Erotic	0.0031	0	0.8601	0.6971	0.936	0.9376
		Neutral	0.7643	0.764	0.9256	0.8699	0.9614	0.9622
	Accuracy		0.6213	0.6207	0.9042	0.8194	0.9526	0.9537
C = 0.1	Precision	Erotic	0	0	0.9847	0.973	1	1
		Neutral	0.6207	0.6207	0.8987	0.8442	0.632	0.6376
$\gamma = 1$	Recall	Erotic	0	0	0.8251	0.7118	0.0484	0.0718
		Neutral	1	1	0.992	0.9883	1	1
	F-score	Erotic	0	0	0.8978	0.8215	0.0922	0.1336
		Neutral	0.764	0.764	0.9427	0.9096	0.7725	0.7766
	Accuracy		0.6207	0.6207	0.9276	0.8813	0.6385	0.647
C = 1	Precision	Erotic	0.7234	1	0.9816	0.9759	0.9716	0.9608
		Neutral	0.8799	0.6212	0.9465	0.9423	0.9608	0.9607
$\gamma = 0.1$	Recall	Erotic	0.8097	0.0022	0.9141	0.9069	0.9389	0.9384
		Neutral	0.7932	1	0.9897	0.9864	0.9829	0.9824
	F-score	Erotic	0.7174	0.0044	0.9466	0.9401	0.955	0.9542
		Neutral	0.8162	0.7644	0.9674	0.9637	0.9717	0.9714
	Accuracy		0.7786	0.6215	0.9602	0.9555	0.9659	0.9654

(*Continued*)

Table A.2 Results of the SVM with RBF kernel classifier. *(Continued)*

SVM with RBF Kernel			BOW		TF-IDF		Word2Vec	
			1-gram	**2-gram**	**1-gram**	**2-gram**	**1-gram**	**2-gram**
C = 1	Precision	Erotic	0.4	0.4	0.9798	0.9718	0.9948	0.9932
		Neutral	0.6207	0.6207	0.956	0.9568	0.682	0.6912
$\gamma = 1$	Recall	Erotic	0.0002	0.0002	0.9306	0.9327	0.2436	0.2764
		Neutral	1	1	0.9884	0.9835	0.9992	0.9988
	F-score	Erotic	0.0005	0.0005	0.9545	0.9517	0.3895	0.4303
		Neutral	0.7641	0.7641	0.9718	0.9699	0.8085	0.8149
	Accuracy		0.6208	0.6208	0.9658	0.9636	0.7102	0.722
C = 10	Precision	Erotic	0.757	1	0.9705	0.9667	0.9698	0.9679
		Neutral	0.8853	0.6235	0.9658	0.9717	0.9602	0.9615
$\gamma = 0.1$	Recall	Erotic	0.8148	0.0122	0.9464	0.9561	0.9384	0.9403
		Neutral	0.8267	1	0.9826	0.9797	0.9819	0.9808
	F-score	Erotic	0.7402	0.024	0.9583	0.9614	0.9538	0.9538
		Neutral	0.8386	0.7661	0.9741	0.9757	0.9709	0.971
	Accuracy		0.8019	0.6251	0.9686	0.9707	0.9649	0.965
C = 10	Precision	Erotic	0.4	0.4	0.9785	0.9597	0.9939	0.993
		Neutral	0.6207	0.6207	0.9597	0.9606	0.6911	0.7015
$\gamma = 1$	Recall	Erotic	0.0002	0.0002	0.9366	0.9386	0.2758	0.3118
		Neutral	1	1	0.9876	0.9825	0.999	0.9986
	F-score	Erotic	0.0005	0.0005	0.957	0.9542	0.4297	0.4724
		Neutral	0.7641	0.7641	0.9733	0.9714	0.8148	0.8219
	Accuracy		0.6208	0.6208	0.9677	0.9654	0.722	0.7353

Table A.3 Results of the LR classifier.

LR			BOW		TF-IDF		Word2Vec	
			1-gram	**2-gram**	**1-gram**	**2-gram**	**1-gram**	**2-gram**
C = 0.1	Precision	Erotic	0.9748	0.9749	0.972	0.9531	0.9505	0.9487
		Neutral	0.9476	0.9455	0.9004	0.8783	0.9659	0.9641
	Recall	Erotic	0.9168	0.9132	0.8303	0.7895	0.9468	0.9438
		Neutral	0.9855	0.9854	0.9852	0.9766	0.9694	0.9681
	F-score	Erotic	0.9449	0.943	0.8953	0.8629	0.9486	0.9462
		Neutral	0.9661	0.9649	0.9405	0.924	0.9677	0.9661
	Accuracy		0.9587	0.9573	0.9253	0.9034	0.961	0.9592
C = 1	Precision	Erotic	0.9672	0.9722	0.9764	0.9619	0.9505	0.9485
		Neutral	0.9599	0.9543	0.9474	0.9473	0.9671	0.9656
	Recall	Erotic	0.9377	0.9282	0.9165	0.9172	0.9483	0.9465
		Neutral	0.9805	0.9835	0.9866	0.9778	0.9693	0.968
	F-score	Erotic	0.9522	0.9497	0.9455	0.939	0.9494	0.9475
		Neutral	0.9701	0.9687	0.9666	0.9623	0.9682	0.9668
	Accuracy		0.9639	0.9621	0.9593	0.9542	0.9617	0.9601
C = 10	Precision	Erotic	0.9558	0.9714	0.974	0.9664	0.9463	0.9472
		Neutral	0.962	0.9563	0.961	0.9636	0.9662	0.9653
	Recall	Erotic	0.941	0.9312	0.9388	0.9435	0.9474	0.9457
		Neutral	0.9736	0.9829	0.9847	0.9796	0.9668	0.967
	F-score	Erotic	0.9483	0.9508	0.956	0.9548	0.9468	0.9464
		Neutral	0.9677	0.9694	0.9726	0.9715	0.9665	0.9662
	Accuracy		0.9609	0.9629	0.9668	0.9657	0.9597	0.9593
C = 100	Precision	Erotic	0.9459	0.9708	0.9668	0.9672	0.9452	0.9496
		Neutral	0.9619	0.9551	0.9628	0.9681	0.9662	0.9657
	Recall	Erotic	0.9411	0.9291	0.9421	0.9504	0.9475	0.9467
		Neutral	0.9674	0.9826	0.9804	0.9802	0.9661	0.9683
	F-score	Erotic	0.9434	0.9495	0.9542	0.9587	0.9464	0.9482
		Neutral	0.9646	0.9686	0.9715	0.9741	0.9661	0.967
	Accuracy		0.9572	0.9619	0.9655	0.9687	0.9592	0.9604

Table A.4 Results of the KNN classifier.

KNNs			BOW		TF-IDF		Word2Vec	
			1-gram	2-gram	1-gram	2-gram	1-gram	2-gram
K = 3	Precision	Erotic	0.8893	0.9179	0.8792	0.8829	0.8292	0.8275
		Neutral	0.7959	0.7226	0.9353	0.9261	0.9784	0.9772
	Recall	Erotic	0.6092	0.392	0.8989	0.8836	0.9699	0.9685
		Neutral	0.9551	0.9795	0.9268	0.9302	0.8794	0.8781
	F-score	Erotic	0.7213	0.5456	0.8887	0.883	0.8939	0.8922
		Neutral	0.8667	0.8293	0.9309	0.9279	0.9262	0.925
	Accuracy		0.8211	0.7531	0.9159	0.912	0.9143	0.9128
K = 5	Precision	Erotic	0.9165	0.9326	0.8898	0.8932	0.8316	0.8324
		Neutral	0.7852	0.7112	0.9417	0.938	0.9781	0.9781
	Recall	Erotic	0.5766	0.3522	0.9099	0.9037	0.9717	0.9695
		Neutral	0.9686	0.9852	0.9331	0.9359	0.8809	0.8819
	F-score	Erotic	0.7061	0.5084	0.8995	0.8981	0.896	0.8956
		Neutral	0.8656	0.8268	0.9373	0.9367	0.9276	0.9275
	Accuracy		0.8171	0.742	0.9238	0.923	0.9159	0.9157
K = 7	Precision	Erotic	0.9263	0.9407	0.8929	0.8985	0.8308	0.8322
		Neutral	0.7798	0.707	0.9469	0.9453	0.9799	0.9785
	Recall	Erotic	0.5606	0.3371	0.9186	0.9151	0.9719	0.9701
		Neutral	0.9734	0.9876	0.9347	0.9391	0.8804	0.8816
	F-score	Erotic	0.694	0.4936	0.9053	0.9065	0.8956	0.8957
		Neutral	0.8641	0.8218	0.9406	0.9421	0.9275	0.9275
	Accuracy		0.8137	0.7379	0.928	0.9294	0.9157	0.9157

Table A.5 Results of the random forest classifier.

Random Forest			BOW		TF-IDF		Word2Vec	
			1-gram	2-gram	1-gram	2-gram	1-gram	2-gram
n_est = 3	Precision	Erotic	0.8315	0.8002	0.8682	0.8466	0.9077	0.8959
		Neutral	0.7133	0.6511	0.7211	0.6783	0.9168	0.9154
depth = 5	Recall	Erotic	0.378	0.1459	0.3957	0.2519	0.8651	0.862
		Neutral	0.9535	0.9766	0.9658	0.9699	0.9462	0.9379
	F-score	Erotic	0.5127	0.2405	0.527	0.3768	0.8858	0.8786
		Neutral	0.8132	0.7788	0.8214	0.7949	0.9311	0.9265
	Accuracy		0.7313	0.6594	0.7415	0.6924	0.9154	0.9099
n_est = 3	Precision	Erotic	0.8171	0.8256	0.8416	0.8148	0.8852	0.8947
		Neutral	0.8746	0.8352	0.88	0.8399	0.922	0.9213
depth = None	Recall	Erotic	0.8001	0.7162	0.8051	0.7297	0.8775	0.8755
		Neutral	0.8917	0.9083	0.9087	0.8995	0.9313	0.9372
	F-score	Erotic	0.8075	0.7659	0.8223	0.7679	0.8812	0.8848
		Neutral	0.8825	0.8693	0.8937	0.8676	0.9265	0.929
	Accuracy		0.8558	0.8341	0.8686	0.8331	0.9105	0.9134
n_est = 10	Precision	Erotic	0.8983	0.9947	0.9248	0.9093	0.9318	0.9268
		Neutral	0.7209	0.6574	0.7232	0.6582	0.928	0.9241
depth = 5	Recall	Erotic	0.3918	0.1556	0.3934	0.1625	0.8833	0.8767
		Neutral	0.9728	0.9945	0.9812	0.9894	0.9606	0.9582
	F-score	Erotic	0.5336	0.26	0.5469	0.2723	0.9068	0.9009
		Neutral	0.8243	0.7888	0.8299	0.7883	0.9439	0.9407
	Accuracy		0.7454	0.6727	0.7538	0.6739	0.9311	0.927
n_est = 10	Precision	Erotic	0.846	0.8681	0.8886	0.8753	0.92	0.9184
		Neutral	0.9319	0.8982	0.9401	0.8931	0.9447	0.9424
depth=None	Recall	Erotic	0.8992	0.8371	0.9087	0.8293	0.9137	0.91
		Neutral	0.901	0.9242	0.9308	0.9285	0.9512	0.9498
	F-score	Erotic	0.8709	0.851	0.8982	0.8506	0.9168	0.9141
		Neutral	0.9159	0.9103	0.9353	0.9098	0.9479	0.9461
	Accuracy		0.8996	0.8893	0.9221	0.8889	0.937	0.9349
n_est = 30	Precision	Erotic	0.9193	0.9388	0.9361	0.9409	0.9377	0.9366
		Neutral	0.7093	0.6486	0.7143	0.6522	0.931	0.9283
depth = 5	Recall	Erotic	0.3496	0.1211	0.3619	0.1361	0.8885	0.8837
		Neutral	0.9807	0.995	0.9846	0.9938	0.964	0.9633
	F-score	Erotic	0.4988	0.2115	0.5167	0.2321	0.9123	0.9093
		Neutral	0.8202	0.7829	0.8256	0.785	0.9471	0.9454
	Accuracy		0.7365	0.6612	0.7454	0.6657	0.9351	0.9329

(*Continued*)

Table A.5 Results of the random forest classifier. *(Continued)*

Random Forest			**BOW**		**TF-IDF**		**Word2Vec**	
			1-gram	**2-gram**	**1-gram**	**2-gram**	**1-gram**	**2-gram**
n_est = 30	Precision	Erotic	0.8962	0.9007	0.9327	0.9165	0.9498	0.9522
		Neutral	0.9342	0.8958	0.9374	0.9075	0.9452	0.9454
depth = None	Recall	Erotic	0.9008	0.8324	0.9016	0.8486	0.9125	0.9134
		Neutral	0.9366	0.9447	0.9609	0.9539	0.9703	0.9718
	F-score	Erotic	0.8978	0.8635	0.9166	0.8804	0.9307	0.9323
		Neutral	0.9351	0.9186	0.9488	0.9296	0.9575	0.9583
	Accuracy		0.9218	0.8992	0.9375	0.9125	0.9482	0.9493