A Semi-Supervised Algorithm for Detecting Extremism Propaganda Diffusion on Social Media

M. Francisco[1*], M.Á. Benítez-Castro[2], E. Hidalgo-Tenorio[3], Juan L. Castro[1]

[1]Department of Computer Science and Artificial Intelligence, University of Granada

[2]English and German Department, University of Zaragoza

[3]English and German Department, University of Granada

Abstract

Extremist online networks reportedly tend to use Twitter and other Social Networking Sites (SNS) in order to issue propaganda and recruitment statements. Traditional machine learning models may encounter problems when used in such a context, due to the peculiarities of microblogging sites and the manner in which these networks interact (both between themselves and with other networks). Moreover, state-of-the-art approaches have focused on non-transparent techniques that cannot be audited; so, despite the fact that they are top performing techniques, it is impossible to check if the models are actually fair. In this paper, we present a semi-supervised methodology that uses our *Discriminatory Expressions* algorithm for feature selection to detect expressions that are biased towards extremist content (Francisco and Castro 2020). With the help of human experts, the relevant expressions are filtered and used to

*Corresponding author: <francisco@decsai.ugr.es>

retrieve further extremist content in order to iteratively provide a set of relevant and accurate expressions. These discriminatory expressions have been proved to produce less complex models that are easier to comprehend, and thus improve model transparency. In the following, we present close to 70 expressions that were discovered by using this method alongside the validation test of the algorithm in several different contexts.

Keywords: feature selection, discriminatory expressions, extremism, propaganda, social media, interpretability, text mining, microblogging

Introduction

With the arrival of Social Networking Sites (SNS) to our lives, we have witnessed a significant increase in sources of information and communication. Social Media is now one of the most important ways of communication, where we can share our opinions and interact with each other in real time. Moreover, since we can observe the user's behaviour inside online platforms, we can analyse their actions and, eventually, even predict mass outcomes.

Microblogging sites such as Twitter are suitable for many data science applications. They are a hybrid between *blogging* and instant messaging, hence their popularity when dealing with real-time events like electoral debates or environmental issues such as earthquakes and so on. In fact, these platforms have already been used to study ongoing social issues and even to coordinate response to

catastrophic events (Periñán-Pascual and Arcas-Túnez 2019; Ashktorab et al. 2014; Wang et al. 2012).

In particular, Twitter has around 126 million daily-active users and almost 6000 *tweets* being sent per second (Twitter Inc. 2019; 'Twitter Usage Statistics - Internet Live Stats' 2013). Tweets are short messages that may contain media and refer to topics (hashtags), and/or other users and their tweets (mentions, responses…). These can be used to build automatic models that assist us in dealing with the aforementioned applications.

However, machine learning (ML) has its limitations[1] and Twitter has a few handicaps that need to be considered when training these models: use of contractions; misspelled words; lack of sufficient context, among others. Most ML algorithms work by trying to minimise the error, so they will focus on getting right the predictions for the majority of users while disregarding several minorities that, although not being irrelevant, have little impact on the optimization procedure.

Extremism Propaganda Diffusion networks (a case of Covert Networks) work in a manner that may be statistically irrelevant in comparison with the normal use cases of the Social Networking Site. Especially in Twitter, they try to stay hidden since the content they publish goes against Twitter's Terms and Conditions[2]. This results in

[1] Limitations inherent to theoretical models that are based on statistics, e.g., a misspelled word retweeted several times may be accepted as a valid classification feature since (1) it is relevant (it appears several times) and (2) it is accurate (since it is a misspelled word, it is likely that it only appears in the original tweet and its retweets, which would belong to the same class and therefore could be interpreted as a predictor for the class).
[2] Available in https://twitter.com/en/tos

unstable accounts that are eventually closed and migrate to other, similar ones. In order to model, analyse and keep track of this type of covert network, it is required to come up with new methods that are able to model minorities in a continuous flow of changing data.

Another issue that we need to consider when developing these mechanisms is that they need to be interpretable. When using automatic tools to make (or assist in making) decisions that may have a social impact, we need to be sure that the tools are fair and transparent, since otherwise we may be perpetuating social injustice (O'Dair and Fry 2019; Phillips 2018; Zhao et al. 2018). ML techniques require data to learn; if we happen to use biased data, the model will learn that bias. Since it is nearly impossible to avoid all sources of biases (the data itself is influenced in the same manner that societies are), we will need to audit the conclusions that an algorithm extract, so as to be sure that these conclusions respect certain principles (FAT/ML n.d.). For example, if we automatically detect an account that is supposedly disseminating extremist propaganda and we close it without checking the facts that the algorithm is uses to make this decision, we may be limiting free speech.

All in all, early detection of extremist propaganda is not a trivial task. First of all, we need to deal with networks of users that behave in an abnormal manner in order to stay hidden; since they are statistically irrelevant, we need to actively search for them and do it with transparency and fairness.

ML models work by extracting conclusions from facts. In the case of natural language problems, these facts are usually word or n-*gram* counts, and sometimes they are complemented with context-independent features such as mean word length and mean number of words per sentence, since these items may disclose information regarding the personality of the author. Hence, we can distinguish between different steps in the process of automatic classification and decision making: (1) pre-processing (removal of stop words, links, tokenisation, normalisation of text…); (2) feature selection (deciding on which facts are relevant for the model); (3) model training (this is when they actually learn and extract conclusions); and (4) model validation (checking that step 3 was performed correctly and measuring how good the model is).

For the present research, our hypotheses are as follows:

(1) We understand that there are certain words or expressions that can identify the class of a text within a certain degree of accuracy;
(2) We believe that it is possible to apply comparison methods between extremist content and general-purpose text (e.g., news, books, magazines…) to find these expressions, which will be later evaluated by experts to decide whether they are relevant or not;
(3) It is our contention that these expressions may be used to train automatic classifiers that would determine if a document contains extremist propaganda or not.

We focused our work on the second step of the process: feature selection. We believe that it is possible to obtain a set of features that, while not affecting accuracy significantly, can generate models that are easier to comprehend (Rudin 2018). Since these features are the training base, they will necessarily affect the classification process. In this paper, we propose an algorithm that would look for expressions that are biased towards a certain class of expressions that are more frequent in a class of documents than in others. These biased expressions will facilitate the expert's audit, while also reducing the complexity of the resulting models.

From Section 2 and onwards, we will describe certain technical aspects that are necessary to understand the present paper, as well as its theoretical background. In Section 3, we address the particulars of our algorithm. Section 4 explains the proposed methodology, and Section 5 presents the settings and the target of the experiments. We discuss our results in Section 6, along with the limitations of this research. We conclude our paper and provide a view on our future work in sections 7 and 8, respectively.

Theoretical Background

In this section, we tackle the technical concepts the reader needs to be familiar with, and we also summarise the key points of other research related to our work.

What is a model and how do we train it?

When it comes to defining a model, we find that there is not one single definition. For the sake of simplicity, we will define a model as a pipeline of algorithms that perform different tasks with the purpose of predicting an outcome. Models are usually divided into classifiers and regressors, depending on the problem they try to solve (Kubat 2017). Regressors are used to predict the value of a dependent variable, given an independent value. The purpose of a classifier is to put each document in the category they belong to. This paper focuses on this latter kind of models.

Classifiers need to be trained, which means providing sufficient input data so that they can extract conclusions. According to how one trains the model, we can distinguish between several groups, although we are going to focus on only two of these: models of supervised and of unsupervised learning. The former consist in giving the expected output as an input, that is, a human expert will annotate each document with the class it presumably belongs to; the algorithm will then try to mimic the expert's decision-making process, using the available data. The latter process consists in providing the algorithm with the documents one wants to classify; without requiring human input, it will decide not only on the class they belong to, but also the classes themselves,. There is no optimal solution here, since all models have different drawbacks, and the choice would depend on their application. The semi-supervised approach is a mix of methods (in which there both are labelled instances and unlabelled ones) in order

to reduce the amount of resources required to annotate the full data set.

How can we check that the model *learned* correctly?

We have several performance metrics that we can use to validate a trained model. Most of them are based on the number of true positives vs. true negatives as well as false positives vs. false negatives that result when applying the model to test data.

Metric	Explanation	Formula
Accuracy	Rate of correctly classified instances between the total number of instances. It stands for how good a model is when correctly classifying instances, but gets easily distorted with imbalanced data.	$acc = \frac{tp + tn}{n}$
Precision	Fraction of positive instances between the number of instances predicted as positive. It helps us answer the question "How many positive items are actually positive?"	$pre = \frac{tp}{tp + fp}$
Recall	Fraction of predicted positive instances among the total number of positive	$rec = \frac{tp}{tp + fn}$

	instances. It helps us answer the question "How many positive items are detected?"	
$f1$-score	Alternative measure of the accuracy of a model. It deals with imbalanced data, but does not take *true negatives* into account.	$f1 = \frac{2}{\frac{1}{pre} + \frac{1}{rec}}$

Table 1: Accuracy measures explained. Although each measure is used to describe certain aspects, for our purpose here, we will rely on f1-score.

None of these measures are a panacea; each of them checks different aspects of a model, so they need to be interpreted as a whole. Although we could use additional metrics, we are going to focus on *f1-score* for the sake of simplicity, relying on quantifiable measures of the performance of a model.

However, these methods do not check if the model is fair. ML relies on statistical learning theory to model specific phenomena; if we train models with biased data, they are going to learn that bias. The implications of biased learning are clear: the resulting models may show outstanding performance, but since they do not have any critical capacity, they can be unfair (e.g. by catering to sexism, racism, homophobia, etc.) In the ideal case, humans should review model behaviour separately from any goodness metrics (FAT/ML n.d.).

Can models be interpreted by humans?

Not all models can be interpreted by humans. A subset can, but top-performing methods such as deep neural networks (popularly known under the label of *deep learning*) work as 'black-box' models; here, although we know how the models work, it is technically impossible to understand all the rules they follow to obtain an output.

Something similar happens in the case of interpretable models. We understand how they work, we can even list all the rules they follow, yet sometimes it is nearly impossible to understand the model holistically, given that the vast number of rules makes it impossible for us humans to manage such a quantity.

There is an ongoing question regarding how many different aspects can be handled by humans at the same time. The psychologist George A. Miller once famously postulated that this number would be 7 ± 2; in other words, any human should only be able to retain between five and nine elements for a short time (Miller 1956). However, further studies addressing the same problem conclude that this number could be even smaller (Cowan 2001).

How do models deal with Natural Language documents?

When working with natural language, a machine would need to transform a document's text[3] into some representation that can be managed by a program.

[3] We refer to a document as a distinct text or a minimum piece of information. For example, when processing newspapers, documents can be articles, paragraphs or even sentences. The kind of *documents* you choose depend on the nature of your study.

Classically, such programs are vectorisations of the words/n-*grams* we can find in the document set as a whole. In other words, given all unique terms (V) in the corpus, they programs turn each document X into a $|V|$-dimensional vector $\vec{X}$ where each dimension i stands for the presence or absence of the i-th element of the vocabulary V in document X. These dimensions (each of them standing for a word or a sequence of them) are called *features*.

Arguably, the most common document representations are Bag of Words (BoW) and TF-IDF (Harris 1954; Sparck Jones 1972). Although commonly used in the literature, these representations result in complex models that are difficult to interpret for several reasons, such as the high dimensionality of feature vectors, the loss of the word order, and the sparsity of the vectors themselves (H.-T. Zheng et al. 2018).

In recent years, other document representations, such as word and context *embeddings*, have come to be used. Here, we are looking at complex, abstract representations of word meanings that are computed using artificial neural networks. Despite the fact that such embeddings are being used in a wide range of applications, the interpretability loss is huge, since the vectors in question merely reflect the internal state of the neurons constituting the network.

Is it possible to reduce the dimensionality of the vector representation?

Given that any given set of documents can result in thousands of different words, it is possible to think that

vector representations are not efficient and should be simplified; that we can reduce dimensionality of the vectors representing the set of documents; that, in fact, it is better to carefully select the features before feeding the classifier, since we will be aiding it by reducing the complexity of deciding (1) which features are more important and (2) how they influence the outcome.

There are many mechanisms that can be used to evaluate how useful a feature is, and they are often classified within three categories (Xue, Zhang, and Browne 2013): filtering methods, wrapper methods and hybrid ones. We are going to focus on filtering methods; these score each feature by applying an evaluation function that considers the correlation between each feature (in this case, each word) and the document labels, and then selects the *k* bests of them. By contrast, wrappers consider the correlation between words and the actual results of the selected classifier.

What are the reference filtering methods?

As we stated earlier, there are a lot of feature selection methods backed by the scientific community and widely used in real-life scenarios. Since we are focusing on filtering methods, we selected a few evaluation functions (presented below), basing ourselves on the relevance, performance and popularity of these methods. Once the evaluation function has been established, the filter is built upon it.

CHI2 (chi-square). This is one of the most popular correlation functions used to build filters for feature

selection problems. It uses the statistical test to check if two events are independent or not, that is, $p(AB) = p(A)p(B)$, where A and B respectively are a feature and a class. The chi-square function is defined by the equation below, where D is the total number of documents, t is a feature and c is a class (Z. Zheng, Wu, and Srihari 2004; Rutkowski et al. 2008; Senthil Kumar B and Bhavitha Varma E 2016):

$$\chi^2_{(t,c)} = \frac{D \times \left[p(t,c)p(\overline{t},\overline{c}) - p(\overline{t},c)p(t,\overline{c})\right]^2}{p(t)p(\overline{t})p(c)p(\overline{c})}$$

Information Gain (IG). This function measures the gain of a feature with respect to a class (Z. Zheng, Wu, and Srihari 2004; Caropreso, Matwin, and Sebastiani 2001; Forman 2003; Largeron, Moulin, and Géry 2011; Ding and Fu 2018; Deng et al. 2019).

$$IG_{(t,c)} = \sum_{c' \in \{c,\overline{c}\}} \sum_{t' \in \{t,\overline{t}\}} p(\mathrm{t}', \mathrm{c}')\, log \frac{p(\mathrm{t}', \mathrm{c}')}{p(\mathrm{t}')p(\mathrm{c}')}$$

Mutual Information (MI). MI measures the level of shared information between a feature and a class (Xu et al. 2007; Al-Salemi, Mohd Noah, and Ab Aziz 2016; Senthil Kumar B and Bhavitha Varma E 2016; Deng et al. 2019).

$$MI_{(t,c)} = \log \frac{p(t,c)}{p(t)p(c)}$$

Odds Ratio (OR). OR measures the co-occurrence probability of a feature and a class, normalised by the

probability of t occurring in other classes (Al-Salemi, Mohd Noah, and Ab Aziz 2016; Z. Zheng, Wu, and Srihari 2004).

$$OR_{(t,c)} = \log \frac{p(t|c)\left(1 - p(t|\overline{c})\right)}{\left(1 - p(t|c)\right)p(t|\overline{c})}$$

Expected Cross Entropy (ECE). ECE ranks the distance between the class distribution co-occurring with the feature t and the class distribution (Wu et al. 2015).

$$ECE_{(t,c)} = p(t) \times \left(p(c|t) \log \frac{p(c|t)}{p(c)} + p(\overline{c}|t) \log \frac{p(\overline{c}|t)}{p(\overline{c})} \right)$$

ANOVA F-value. This checks if there is a significant difference between the variances of two variables (Misangyi et al. 2016).

$$F\text{-}statistic = \frac{\text{variance between groups/}}{\text{variance within groups}}$$

Galavotti-Sebastiani-Simi coefficient (GSS). This is a simplified version of chi-square (Galavotti, Sebastiani, and Simi 2000; Largeron, Moulin, and Géry 2011).

$$\mathrm{GS}S_{(t,c)} = \mathrm{p}(t,c)\mathrm{p}\left(\overline{t},\overline{c}\right) - \mathrm{p}(t,\overline{c})\mathrm{p}\left(\overline{t},c\right)$$

In general, the filtering methods have similar disadvantages. They select features based on the correlation between a feature and a document class but not all evaluation functions yield the optimum feature set for each classifier. By contrast, wrappers, on the other hand, are capable of determining the best subset of functions for each classifier, since the former receive feedback from the latter: they score each feature by testing

if it improves the accuracy of an underlying classifier. Wrappers are slower than filters, since the time of evaluating each subset depends on the training time of the classifier. All in all, in our work we stick to filters, as they may be described using a formula, hence can be interpreted. Also, tailoring features (or facts) to the specific needs of a classifier operates on a less general scale than do correlation functions. Therefore, in an interpretable context, filtering methods make more sense. since facts should remain intact, regardless of the classifier 'judge'.

Are filters going to help us comprehend models?

Filters may facilitate the training step, by reducing the dimensionality and making models more comprehensible. However, our preliminary study of the actual comprehensibility of models trained with the features selected by classical filtering methods led us to conclude that the results were not good enough. All of the methods presented above required between 20 and 100 features to achieve a median value of 0.55 in *f1-score*; the resulting models would have a mean complexity of 366 rules with 12 clauses each. We believe there is room for improvement here, and that it is possible to obtain a set of features that, while not affecting accuracy significantly, can generate models that are easier to comprehend (Rudin 2018).

How can we be sure that this is the way to go?

Benigni et al. suggest that ISIS continues to use social media as an essential element of propaganda (Benigni, Joseph, and Carley 2017). They also discovered that detecting users whose activity supports ISIS propaganda

diffusion is especially complex, given that there are different roles and degrees (unaffiliated sympathizers, propagandists, fighters and recruiters). Having explored a large community of Twitter users by using the computational technique of *Iterative Vertex Clustering and Classification (IVCC)*, these authors claim that the system outperforms previous approaches; still, they highlight that it is unlikely that a sufficient number of labelled cases will always be available, and therefore they suggest applying semi-supervised algorithms or active learning to improve this type of systems.

Alvari, Sarkar, and Shakarian (2019) likewise present an automatic detection scheme for Violent Extremists in Social Media by using three groups of information respectively related to user names, user profiles and textual content. These authors also claim that a valuable research direction would be to deploy iterative supervised learning in order to improve the system performance.

Most of the recent papers focus on deep learning models (Alharbi and de Doncker 2019). Such models are not interpretable by humans; hence we need to trust the models themselves, alternatively rely on surrogate models to explain them. Deep neural networks can extract features directly from the text, but they are also capable of interacting with traditional feature vectors. In this sense, the main opportunity for human interaction is in the feature selection process.

Automatic feature extraction is an important research field for text mining in Social Media, due to the extreme

conditions these media present (i.e. short messages, misspelled words, emoticons...). In this connection, we have developed an algorithm for detecting and ranking *Discriminatory Expressions* (*DE*, i.e. expressions with a significant difference in the statistical frequency between classes) as an extraction feature especially suitable for Social Media (Francisco and Castro 2020).

In the context of terrorism propaganda and its diffusion, we think that our DE algorithm will produce relevant expressions for detecting this kind of documents. However, given the small number of labelled cases, other expressions may arise; hence in order to obtain a high-performance system, human supervision is still required to polish the feature subset generated by DE.

Discriminatory Expressions (DE)

In this section, we present our proposed algorithm to select features that are relevant to identify extremist propaganda. Here, we will not develop all technical aspects of the algorithm, since we prefer to focus on the application of the algorithm (the reader may wish to refer to our original paper (Francisco and Castro 2020).

Definition (Expression). Given a document $d = (t_1, t_2, \dots, t_n)$ as a sequence of words, e is an expression of the document if, and only if, (1) there is at least one word in e that is not a stop word, and (2) all words of the expression can be found in the document d and the order is preserved.

Definition (Discriminatory Expression). Given the minimum relevance (or recall) r and a minimum precision p, an expression e is said to be (r, p)-discriminatory for a given class C, if, and only if, (1) e is an expression, (2) the recall of the expression e for the class C is above the given threshold r, and (3) the precision of the expression e for the class C is at least p.

Taking r and p as its *hyperparameters*, the proposed algorithm will compute a set of discriminatory expressions biased towards class C; it will turn each document d into a vector $\vec{d}$ where each component d_i stands for the occurrence of the i-th discriminatory expression in the document.

In order to do this, the proposed feature selection technique will arrange candidates in order of importance (or their potential ability to become a discriminatory expression) by using the CF-ICF ranking method (Francisco and Castro 2020). CF-ICF evaluates the importance of each word as the ratio between the number of times that the word appears in a class and the number of documents that contain it, thus allowing us to prioritise expressions by maximising their relevance and performance.

Algorithm DE for feature selection

Require: Training set of documents X, vector of labels y, p and r

Ensure: Set of discriminatory expressions

1. For each document:

---

a. Tokenize
b. Remove stop words
c. Apply stemming
d. Sort elements by CF-ICF
e. Add candidates to the candidate list

2. For each candidate:
    a. Compute precision and recall of the candidate
    b. If candidate recall is greater or equal than r
        i. If candidate precision is greater or equal than p, and there is at least one word that is not a stop word, accept the candidate as a discriminatory expression
        ii. If (i) does not apply, expand the expression with the remaining candidates (the Cartesian product of the current candidate with the rest of them), and go to step 2
    c. In all other cases, discard the current candidate
3. Go to step 2 until the candidate list is empty or the accepted features list reach the size limit
4. For each document, apply its transformation into a vector of DE features

---

Words require context. Given a word, depending on its co-text, the meaning of the sentence can vary. Our algorithm relies on the surrounding words and the order in which they are presented in the document, so as to find combinations that are helpful towards determining if a document belongs to a class. Since we take into

consideration both occurrence and order, the resulting features are easily comprehended by humans.

Methodology

In this section, we present our proposed methodology for building a model that, using Discriminatory Expressions (DE), can automatically determine if a document contains extremist propaganda.

1. First of all, we need to gather an initial data set of documents that contain extremist propaganda. Even if small in size, the set needs to be carefully curated by experts for the algorithm to work correctly. We will also require a set of topic-related documents that do not contain any propaganda, as well as a further set of documents unrelated to the topic. This is necessary for the algorithm to compare the sets in question, and check that recall and precision for the positive class are good enough.
2. Subsequently, we apply our DE algorithm to find all relevant expressions that can differentiate between classes. The resulting set of features needs to be revised by experts in the field, to determine if the expressions are (1) relevant, (2) meaningful, and (3) accurate.
3. Following this, we will train a classifier with the chosen expressions in order to automatically determine if a document contains propaganda.
4. Additionally, we will retrieve more content and use the trained model to classify the new documents into the positive class (i.e. containing extremist propaganda) or the negative class (i.e. not containing extremism).

5. Finally, we will return to step 2 as many times as necessary.

The approach sketched here is expected to help us maintain a set of expressions that are relevant in the literature of extremist propaganda.

Experiments

In this section, we present three experiments we have run in order to check the accuracy and suitability of our proposal.

Performance and Comprehensibility Tests

We used several popular data sets to test our proposal within different contexts and topics. All of these data are publicly available or can be requested from their authors: US Airlines Sentiment[4], Twitter User Gender[2], Sentiment140[5] (Go, Bhayani, and Huang 2009), SLS IMDB subset[6] (Kotzias et al. 2015), and TASS[7] (Villena-Román et al. 2013). We compared our algorithm with several of the filtering methods presented earlier (CHI2, IG, MI, OR, ECE, f-ANOVA and GSS), and used four interpretable classifiers (k-nearest neighbours (kNN), decision tree (DT), random forest (RF), and logistic regression (LR)).

Tests were conducted using a 5-fold cross validation scheme (employing python 3.8.1, pandas 0.25.3, NLTK 3.4.5 and sklearn 0.22). We used nine features to test our models (unless specified otherwise), since this number has

[4] https://www.figure-eight.com/data-for-everyone
[5] https://www.kaggle.com/kazanova/sentiment140
[6] https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences
[7] http://tass.sepln.org/2017

traditionally been assumed to be the maximum number of elements in working memory (Miller 1956).

The results of these experiments were measured using several performance metrics (namely, accuracy, precision, recall, *f1-score*, and area-under-the-curve ROC); for the sake of simplicity, below we are going to focus on the *f1-score* metric.

Application-related Tests

More than 250 articles from several extremist magazines (i.e. *Azan, Dabiq, Gaidi Mtaani*, *Inspire*, *Jihadi Recollections* and *Rumiyah*) were collected and subsequently coded by human experts on the grounds of author gender, date of publication, and the text's function (recruitment, propaganda, indoctrination, radicalization, and instructions). They were compared against 18k news articles (topic-related texts and abstracts) plus emails, all from REUTERS[8] and 20-Newsgroups[9] data sets.

Results and Discussion

In this section, we present the results yielded by our tests. For each (kind of) text, we only explore the most important result. (Full results are available in Francisco and Castro 2020).

We conducted 5-fold cross validation using 4 different unrelated data sets with 4 different interpretable classifiers. As mentioned earlier, in order to simplify interpretation and because of its capacity to deal with

[8] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection
[9] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

imbalanced data, here we are going to focus on the *f1-score* to measure classification performance.

Figure 1 shows classification performance measured with *f1-score* against the number of features used. DE features were proved to be more useful than the rest (owing to their limited dimensionality; other methods may show better performance for over 20 features). This result indicates that we can obtain a neat baseline performance using fewer, but more meaningful, features. Hence, it is possible to maintain a good accuracy while reducing dimensionality; this again directly affects model complexity.

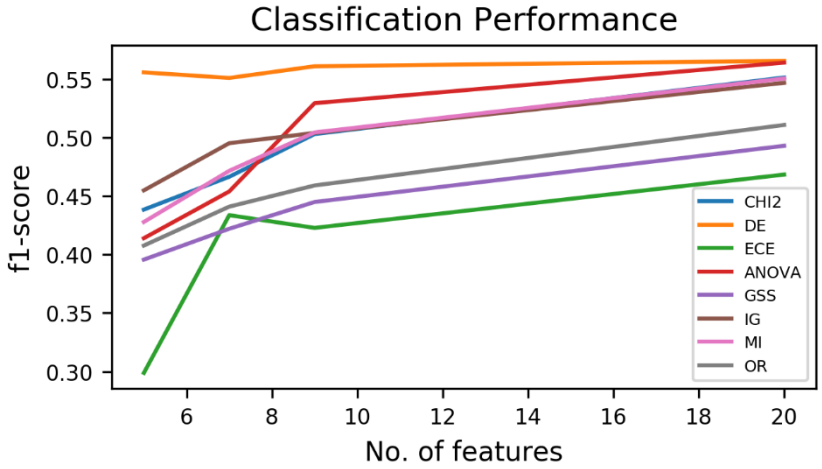

Figure 1

Mean f1-score of each feature selection algorithm against number of features. Whilst most mechanisms start under 0.45 and increase their performance with the number of features, our proposal (DE) starts over 0.55, giving a neat baseline performance for 5-9 features.

Figure 2 exhibits centrality and statistical dispersion of each fold. DE features are not only better (for under 20 features)

but also more consistent. This indicates that DE is less sensitive to (1) training sets (since there are several datasets), (2) topic (since each dataset has a specific topic), and (3) lack of context, at least with a reduced feature space (as when the datasets are tweets, which have an extension limit, currently of 280 characters).

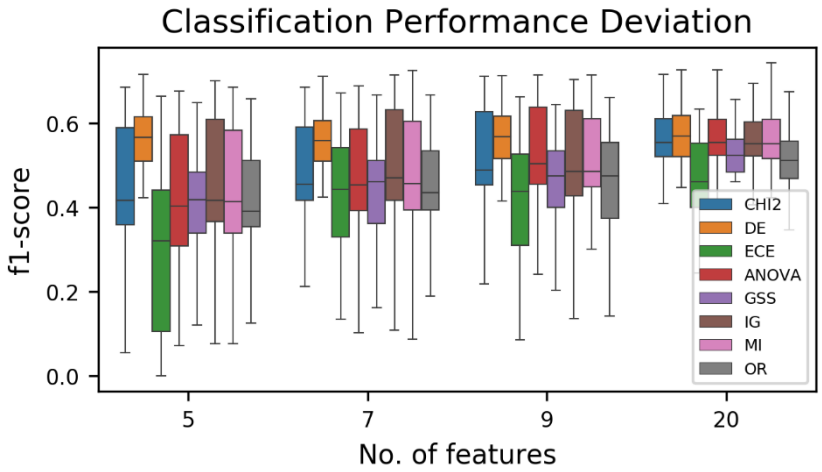

Figure 2

Distribution of f1-score over number of features for each feature selection method.

Not only does DE have good results under 10 features but they are also the most consistent. Inter-quartile range is the lowest among the rest of methods and the distribution is almost balanced. Although f1-score increases with the number of features, DE manages to yield approximately the same results when using 5 rather than 20 features, facilitating model interpretability.

As for comprehensibility, apart from reducing the number of features, the resulting models are less complex and therefore more likely to be interpretable in real-life

scenarios. When we measured the complexity of kNN, DT and RF with different feature sets, the DE set was the one that produced less complex models, especially when working with decision trees. In our tests, classifiers trained with the gender data set are less comprehensible, whereas DE features produced reasonable classifiers.

Table 1 shows complexity and comprehensibility scores for Decision Tree models with different feature sets. In terms of complexity and *f1-score*, the DE models are the best. We computed complexity and model comprehensibility following the equations in Francisco and Castro (2020).

	f1		leaves		length		complexity	comprehensibility
	mean	std	mean	std	mean	std	mean	mean
CHI2	0.5595	**0.0854**	95.2500	142.7127	8.6858	3.4138	28.7440	2.0857
DE	0.5711	0.1416	**27.2500**	**10.4363**	**6.8067**	**0.3266**	**5.0500**	**11.3083**
ECE	0.4623	0.1393	346.2500	398.5736	11.7405	2.5386	190.9065	0.2421
F-ANOVA	**0.5939**	0.0992	78.7500	104.2605	8.5890	3.0212	23.2377	2.5557
GSS	0.4486	0.1921	37.5000	14.8885	7.8571	0.7002	9.2602	4.8447

IG	0.6051	0.0860	99.7500	141.4340	8.6970	3.1512	30.1796	2.0051
MI	0.5927	0.0994	95.5000	138.4883	8.2910	3.3720	26.2592	2.2573
OR	0.5167	0.1246	323.7500	363.7530	10.8829	2.7581	153.3770	0.3369

Table 2

Performance metrics of the algorithm and comprehensibility measures of the resulting models when trained with each feature selection mechanism. DE manages to have the highest score in comprehensibility.

Application-specific Results

After applying our algorithm to propaganda and recruitment magazines while looking for 3-word discriminatory expressions (with recall percentile of 75 and a precision of 0.9), we obtained the expressions listed in Table 2.

We display the expression in the form of sequences of words, with wildcards in between each word and at both ends. Each wildcard stands for any given sequence of characters. Consequently, each sentence is going to yield a positive result when tested for an expression if the words that confirm these expressions' presence in the sentence in their exact order, independently of what comes before,

after, and between them. Highlighted words are the ones selected by our experts after filtering for the relevant ones.

rāfidah	*obama*	*āt*
kufr	*brother*islam*	*last*islam*
abū	*sahwah*	*crusad*war*
shām	*rasūlullāh*	*mujāhid*
allah	*islam*state*one*	*came*islam*
hijrah		*syria*iraq*
tawāghīt	*releas*islam*	*alayhis-salām*
jihād	*halab*	*islam*militari*
mujāhidīn	*mujahidin*	*ramadān*
khilāfah	*airstrik*	*sharī*
*mani*islam*state*	*american*islam*	*kuffār*
shaytān	*state*apost*	*prophet*muhammad*
*crusad*one*	*support*crusad*	*crusad*nation*
wah		*word*crusad*
ibn	*ar-raqqah*	*muslim*us*
ummah	*murtadd*	*war*islam*state*
dābiq	*tāghūt*	*at-tawbah*
wilayat	*al-islām*	*attack*crusad*
alayhi	*wilāyāt*	*shirk*
āl	*qur*	*kāfir*
nusayrī	*muwahhid*	*one*crusad*
*islam*front*	*islam*state*includ*	*prophet*g*
*word*enemi*	*2014*	*shaykh*
	*said*repor*muslim*	

wilāyat

Table 3

*A list of relevant expressions detected by our algorithm when applying our methodology to several propaganda magazines. Highlighted expressions are those that passed our experts' filter. Due to the fact that some magazine issues are from different dates than the texts used in the negative class, certain expressions, such as *2014*, achieve relevance and precision targets despite being irrelevant. This hitch can be avoided by extending the negative class, or by manually removing those that are irrelevant, which has been the case here.*

Limitations

We applied our algorithm to propaganda and recruitment magazines whose content may not be the same as that occurring in the microblogging accounts. For this reason, rather than using whole articles as documents, we tokenized the appropriate sentences and fed them to the algorithm one by one, simulating tweets. This is not an ideal approach, since full-length articles rely on context to transmit ideas to the reader, whereas tweets have very little context or none at all. We are actively working on obtaining a data set of propaganda tweets so that we can improve the results presented here.

Moreover, since all of the expressions examined in our research are language-dependent, we would need data sets for any given language we may want to work with. It Possibly, this problem may be overcome by using

embedding, but doing that would affect the very interpretability of the features discovered. And even so, we would need to test the method's performance for different languages, given that any earlier defined performance of expressions may vary significantly, in particular if one is working with Arabic texts.

Conclusions

Arguably, Social Media is currently the principal and most important way of communication. Microblogging sites such as Twitter can be used for many research applications, and their characteristics (promptness and short messages) enable them to be used widely.

Reportedly, extremist networks using Twitter to spread recruitment and propaganda statements have tried to stay hidden, since this kind of content goes against Twitter Terms and Conditions. Direct consequences are not only the instability of the accounts; several other disadvantages appear when one is tracking and studying the networks' online behaviour.

Current Machine Learning (ML) techniques are not relevant in this context, inasmuch most covert networks are not statistically relevant; so new methods were necessary in order to deal with the issues at hand. In the present paper, we present a semi-supervised methodology to detect certain words in a fixed order, specifically expressions that can be used to determine a document's relevance for the study of online extremism.

One key issue in the present paper is the methodology employed to iteratively build a set of discriminatory expressions by (1) retrieving relevant and non-related content, (2) applying the algorithm to the set of documents, (3) filtering the selected features (using human experts) and (4) use the filtered features to retrieve more relevant content and start again, until necessary.

When applying the methodology to the 250 documents extracted here from extremist propaganda magazines, almost 45% of the expressions found were judged to be relevant by human experts, thereby proving that the algorithm may be fruitfully used for this particular purpose. In addition, the machine learning models that were built including these feature sets are expected to generate significantly less complex models, thereby improving human comprehensibility and transparency.

Despite the fact that we tested our algorithm with several contexts and sources of information, there is still room for improvement. The current work is part of a larger project aiming to build a set of tools for keeping track of online covert networks, in order to enhance research capabilities in this particular area.

Future Work

We are actively working on building several supervised data sets within different topics (mainly politics and extremist propaganda) and languages (English, Spanish and Arabic), which will help us test our algorithm more thoroughly at the application level.

Moreover, feature selection is just one chunk of the machine learning pipeline effort. In order to build a fully comprehensible model that can be evaluated and approved by human experts, it is necessary to extend our study to comprise classifiers.

Funding Information

This research has been financially supported by several entities: the European Social Fund, the Spanish Ministry of Economy and Competitiveness (Project Reference: FFI2016-79748-R), and the Junta de Andalucía (Project References: P18-FR-5020 and A-HUM-250-UGR18). Furthermore, Manuel Francisco Aparicio has also been funded by the Spanish Ministry of Economy and Competitiveness 2017 FPI Predoctoral Programme (Grant Number: BES-2017-081202).

References

Alharbi, Ahmed S.M., and Elise de Doncker. 2019. 'Twitter Sentiment Analysis with a Deep Neural Network: An Enhanced Approach Using User Behavioral Information'. *Cognitive Systems Research* 54: 50–61. https://doi.org/10.1016/j.cogsys.2018.10.001.

Al-Salemi, Bassam, Shahrul Azman Mohd Noah, and Mohd Juzaiddin Ab Aziz. 2016. 'RFBoost: An Improved Multi-Label Boosting Algorithm and Its Application to Text Categorisation'. *Knowledge-Based Systems* 103 (July): 104–17. https://doi.org/10.1016/j.knosys.2016.03.029.

Alvari, Hamidreza, Soumajyoti Sarkar, and Paulo Shakarian. 2019. 'Detection of Violent Extremists in Social Media'. *ArXiv:1902.01577 [Cs]*, February. http://arxiv.org/abs/1902.01577.

Ashktorab, Zahra, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. 'Tweedr: Mining Twitter to Inform Disaster Response.' In *ISCRAM*.

Benigni, Matthew C., Kenneth Joseph, and Kathleen M. Carley. 2017. 'Online Extremism and the Communities That Sustain It: Detecting the ISIS Supporting Community on Twitter'. *PLOS ONE* 12 (12): e0181405. https://doi.org/10.1371/journal.pone.0181405.

Caropreso, Maria Fernanda, Stan Matwin, and Fabrizio Sebastiani. 2001. 'A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization', 15.

Cowan, Nelson. 2001. 'The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity'. *The Behavioral and Brain Sciences* 24 (1): 87–114; discussion 114-185.

Deng, Xuelian, Yuqing Li, Jian Weng, and Jilian Zhang. 2019. 'Feature Selection for Text Classification: A Review'. *Multimedia Tools and Applications* 78 (3): 3797–3816. https://doi.org/10.1007/s11042-018-6083-5.

Ding, Jianli, and Liyang Fu. 2018. 'A Hybrid Feature Selection Algorithm Based on Information Gain and Sequential Forward Floating Search'. *Journal of Intelligent Computing* 9 (3): 93. https://doi.org/10.6025/jic/2018/9/3/93-101.

FAT/ML. n.d. ‘Principles for Accountable Algorithms and a Social Impact Statement for Algorithms’. Accessed 8 January 2019. http://www.fatml.org/resources/principles-for-accountable-algorithms.

Forman, George. 2003. ‘An Extensive Empirical Study of Feature Selection Metrics for Text Classification [J]’. *Journal of Machine Learning Research - JMLR* 3 (March).

Francisco, Manuel, and Juan Luis Castro. 2020. ‘Discriminatory Expressions to Produce Interpretable Models in Microblogging Context’. *ArXiv:2012.02104 [Cs]*, November. http://arxiv.org/abs/2012.02104.

Galavotti, Luigi, Fabrizio Sebastiani, and Maria Simi. 2000. ‘Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization’. In *Research and Advanced Technology for Digital Libraries*, edited by José Borbinha and Thomas Baker, 59–68. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-45268-0_6.

Go, Alec, Richa Bhayani, and Lei Huang. 2009. ‘Twitter Sentiment Classification Using Distant Supervision’. *Processing* 150 (January).

Harris, Zellig S. 1954. ‘Distributional Structure’. Word 10 (2–3): 146–62. https://doi.org/10.1080/00437956.1954.11659520.

Kotzias, Dimitrios, Misha Denil, Nando de Freitas, and Padhraic Smyth. 2015. ‘From Group to Individual

Labels Using Deep Features'. In *KDD '15*. https://doi.org/10.1145/2783258.2783380.

Kubat, Miroslav. 2017. *An Introduction to Machine Learning*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-63913-0.

Largeron, Christine, Christophe Moulin, and Mathias Géry. 2011. 'Entropy Based Feature Selection for Text Categorization'. In *ACM Symposium on Applied Computing*, edited by William C. Chu, W. Eric Wong, Mathew J. Palakal, and Chih-Cheng Hung, 924–28. TaiChung, Taiwan: ACM. https://doi.org/10.1145/1982185.1982389.

Miller, George A. 1956. 'The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information'. *Psychological Review* 63 (2): 81–97. https://doi.org/10.1037/h0043158.

Misangyi, Vilmos F., Jeffery A. LePine, James Algina, and Jr Francis Goeddeke. 2016. 'The Adequacy of Repeated-Measures Regression for Multilevel Research: Comparisons With Repeated-Measures ANOVA, Multivariate Repeated-Measures ANOVA, and Multilevel Modeling Across Various Multilevel Research Designs'. *Organizational Research Methods*, June. https://doi.org/10.1177/109442810528319O.

O'Dair, M., and A. Fry. 2019. 'Beyond the Black Box in Music Streaming: The Impact of Recommendation Systems upon Artists'. *Popular Communication*. https://doi.org/10.1080/15405702.2019.1627548.

Periñán-Pascual, Carlos, and Francisco Arcas-Túnez. 2019. 'Detecting Environmentally-Related Problems on

Twitter'. *Biosystems Engineering*, Intelligent Systems for Environmental Applications, 177 (January): 31–48. https://doi.org/10.1016/j.biosystemseng.2018.10.001.

Phillips, Avery. 2018. 'The Moral Dilemma of Algorithmic Censorship'. Becoming Human: Artificial Intelligence Magazine. 27 August 2018. https://becominghuman.ai/the-moral-dilemma-of-algorithmic-censorship-6d7b6faefe7.

Rudin, Cynthia. 2018. 'Please Stop Explaining Black Box Models for High Stakes Decisions'. *ArXiv:1811.10154 [Cs, Stat]*, November. http://arxiv.org/abs/1811.10154.

Rutkowski, Leszek, Ryszard Tadeusiewicz, Lofti A. Zadeh, and Jacek M. Zurada. 2008. *Artificial Intelligence and Soft Computing – ICAISC 2008: 9th International Conference Zakopane, Poland, June 22-26, 2008, Proceedings*. Springer Science & Business Media.

Senthil, Kumar B. and Bhavitha Varma E. 2016. 'A Different Type of Feature Selection Methods for Text Categorization on Imbalanced Data' 5 (9): 7.

Sparck-Jones, Karen. 1972. 'A Statistical Interpretation of Term Specificity and Its Application in Retrieval'. *Journal of Documentation* 28 (1): 11–21. https://doi.org/10.1108/eb026526.

Twitter Inc. 2019. 'Q1 2019 Earning Report'. https://s22.q4cdn.com/826641620/files/doc_financials/2019/q1/Q1-2019-Slide-Presentation.pdf.

'Twitter Usage Statistics - Internet Live Stats'. 2013. 2013. http://www.internetlivestats.com/twitter-statistics/.

Villena-Román, Julio, Sara Lana-Serrano, Eugenio Martínez-Cámara, and José Carlos González-Cristóbal. 2013. 'TASS - Workshop on Sentiment Analysis at SEPLN'. *Procesamiento del Lenguaje Natural* 50 (0): 37–44.

Wang, Hao, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. 'A System for Real-Time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle'. In *Proceedings of the ACL 2012 System Demonstrations*, 115–20. ACL '12. Stroudsburg, Penn., USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=2390470.2390490.

Wu, Guohua, Liuyang Wang, Nailiang Zhao, and Hairong Lin. 2015. 'Improved Expected Cross Entropy Method for Text Feature Selection'. In *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, 49–54. https://doi.org/10.1109/CSMA.2015.17.

Xu, Yan, Gareth Jones, Jintao Li, Bin Wang, and Chunming Sun. 2007. 'A Study on Mutual Information-Based Feature Selection for Text Categorization'. *Journal of Computational Information Systems* 3 (March).

Xue, Bing, Mengjie Zhang, and Will Browne. 2013. 'Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach'. *IEEE Transactions on Cybernetics* 43 (December): 1656–71. https://doi.org/10.1109/TSMCB.2012.2227469.

Zhao, Z., M. Gao, J. Yu, Y. Song, X. Wang, and M. Zhang. 2018. 'Impact of the Important Users on Social Recommendation System'. *Lecture Notes of the*

Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST 252: 425–34. https://doi.org/10.1007/978-3-030-00916-8_40.

Zheng, Hai-Tao, Zhe Wang, Wei Wang, Arun Kumar Sangaiah, Xi Xiao, and Congzhi Zhao. 2018. 'Learning-Based Topic Detection Using Multiple Features'. *Concurrency and Computation-Practice & Experience* 30 (15): e4444. https://doi.org/10.1002/cpe.4444.

Zheng, Zhaohui, Xiaoyun Wu, and Rohini Srihari. 2004. 'Feature Selection for Text Categorization on Imbalanced Data'. *ACM SIGKDD Explorations Newsletter* 6 (1): 80–89. https://doi.org/10.1145/1007730.1007741.

Manuel Francisco (corr. author)

Rafael Gómez Montero, 2

CITIC-Univ. Granada

Granada 18071, Spain

francisco@decsai.ugr.es

Miguel-Ángel Benítez-Castro

Universidad de Zaragoza

Facultad de Ciencias Sociales y Humana

Filología Inglesa y Alemana

C/ Ciudad Escolar, S/N

Teruel 4403, Spain

mbenitez@unizar.es

Encarnación Hidalgo-Tenorio

Universidad de Granada

Facultad de Filosofía y Letras

Departamento de Filologías Inglesa y Alemana

Campus de Cartuja S/N

Granada 18071, Spain

ehidalgo@ugr.es

Juan-Luis Castro-Peña

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

Granada 18071, Spain

jcastro@ugr.es

To Be Added for each author

Info about funding

Bio Notes (max. 150 words)

Addresses for Correspondence (postal & email)

Publication history

Date received: 26 January 2021

Date accepted: 16 November 2021