INFOTEC-LaBD at PAN@CLEF21: Profiling Hate Speech Spreaders on Twitter through Emotion-based Representations

Notebook for PAN at CLEF 2021

Hiram Cabrera , Sabino Miranda-Jiménez and Eric S. Tellez

INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur No. 112, Fracc. Tecnopolo Pocitos Aguascalientes, Ags., México

Abstract

Nowadays, social media is perhaps one of the most powerful channels of communication among people worldwide. Despite the physical constraints, people in a social network communicate efficiently and instantaneously without restriction. While this can promote the interchange of ideas and information, in this scenario, people with a dangerous idiosyncrasy can achieve more people with low restrictions. Automatic *hate speech* identification in social networks is a Natural Language Processing task dedicated to pointing out users that have this kind of misconduct among its publication's content. In this work, we tackled the PAN21 Hate Speech identification task through Semantic Emotion-based models in both Spanish and English languages. We implement several approaches, one of them is designed to output explainable results based on the user's emotional charge.

Keywords

hate speech, author profiling, emotion-based classification

1. Introduction

Social media is perhaps one of the most powerful channels of communication among people worldwide. Despite the physical constraints, people in social networks such as Twitter interchange efficiently and instantaneously ideas and information without restriction. In this scenario, hate speech emerges as a problem when communication denigrates a person or a group based on some characteristics such as race, color, gender, or sexual orientation.

Automatic identification of hate speech has been popular because of the nature of social networks, and it has been tackled on several fronts. On the one hand, several competitions have been run contests at the message level. For example, the HatEval [1] challenge considers the identification of hate speech against immigrants and women in Twitter as a two-class classification problem, i.e., whether a tweet is hateful or not hateful. Also, OffensEval [2] challenge consists in determining whether a given message has offensive content. This event runs several tasks, such as identifying whether a message has offensive language and categorizing offense types. Among the offense types, OffensEval considers messages containing an insult or threat to someone, or a

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

hiram.cabrerap@infotec.mx (H. C.); sabino.miranda@infotec.mx (S. M.); eric.tellez@infotec.mx (E. S. Tellez)

tweet containing non-targeted profanity and swearing, and identifying the target, i.e., whether the offensive content is about an individual, a group, or others. On the other hand, author profiling has become a powerful tool for NLP applications offering multiple approaches to tackle several tasks such as author attribute identification [3, 4], sentiment analysis [5, 6], and text classification [7]. In this sense, the PAN @ CLEF21 challenge considers the profiling hate speech spreaders task [8, 9], identifying authors who have shared some hate speech in the past according to the tweets published. The competition determines whether a user is a hate speech spreader given a set of 200 tweets per author, for English and Spanish languages. Approaches to face this problem commonly use external information such as lexicons or datasets from related domains to enrich the knowledge database or encode semantic information generally using word embeddings. In the following sections, we introduce the tools used in our approach to profiling hate speech spreaders.

Sentiment and emotion lexicons

Using lexicons of labeled words is a fundamental tool in many text classification approaches. The main idea behind this method is to use curated information that associates words with some helpful information to solve the task.

Despite that dictionary-based classifiers are among the first approaches in the community, its simplicity and the inherent ability of explanation, these methods remain popular even today. For instance, Nielsen [10] introduces AFINN,[1] a lexicon associating a vocabulary of more than three thousand words in four languages with a degree of sentiment. It also includes sentiment scores for emojis. Bing Liu [11, 5] also provides a list of English words and their associated sentiment.[2] Currently, the lexicon contains close to 6800 entries. Mohammed and Turney [12] introduced the NRC Word-Emotion Association Lexicon (also called EmoLex);[3] the lexicon contains more than 14,000 English words associated with eight basic emotions: anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. EmoLex also provides the scoring for both negative and positive sentiments. All these lexicons have been automatically translated into several languages. The number of words has also been increased since their creations.

However, one of the main drawbacks of lexicons is that they need to be created by experts and have an inherent dependence on language and domain. Another critical issue is the lack of exhaustiveness due to the explosion of terms (e.g., neologisms, inflections, synonymy, hyponymy, hypernymy). Finally, depending on the domain, the lexical variations and errors also negatively affect the performance of the models.

Our work is based on the EmoLex dictionary along with semantic representations of the vocabulary to cope with many of the issues regarding plain lexicons.

Word embeddings

Several of these limiting issues can be solved using semantic word-embeddings, which are semantic lexicons associating words with a vector in a semantic space. Semantic spaces are

[1] https://github.com/fnielsen/afinn

[2] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[3] https://www.saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

learned from a huge non-annotated text corpus, based on the distributional hypothesis of semantics, i.e., words with similar meanings will tend to appear in similar contexts. Some examples of word embeddings are fastText [13] and Global vectors (GloVe) [14].

Semantic language models are sophisticated methods dedicated to understanding language, and they are created to predict the semantic of a sequence of words through very large non-annotated corpora. Our approach centers on word meanings instead of sentence meanings; therefore, language models are beyond the scope of this contribution. The interested reader is referred to the related literature [15, 16, 17, 18].

FastText[4] is a word-embedding with a fast construction. It learns word distributional semantic using small windows around words. In addition to other approaches, it is designed to tackle out-of-vocabulary words using *subwords* which are small substrings that compose words and are to compute word-vectors whenever a word is unknown. Using fastText models, we cope with the vocabulary diversity found in social networks.

Machine learning models

A popular way to tackle author profiling problems is supervised learning; in this approach, a set of labeled examples is given to an algorithm to create a model that can label never-seen examples. The PAN @ CLEF21 asks for profiling of *hate speech spreaders* using user's textual information retrieved from Twitter. Therefore, our dataset examples are a list of text messages and their associated label.

Classical supervised learning will receive a set of examples (X, y) of the form $X = x_1, x_2, \cdots, x_n$ and $y = y_1, y_2, \cdots, y_n$, where x_i is a vector of some dimension δ, i.e., $x_i \in \mathbb{R}^{\delta}$. On the other hand, y_i is a categorical value that represents a valid class. For this matter, profiling methods based on supervised learning need to transform input messages into a real-valued vector, y is often obtained by human annotators that assign a label to each example. From a general perspective, the vectorization can be based on how authors write, the actual content, or the meaning of their messages. More detailed, we can capture how authors write using stylometry features [19]. The content-based representations follow a generic procedure that relies on preprocessing the text, tokenize it with a variety of possible schemes to create bag-of-words, and then using a weighting scheme to vectorize [7, 20]. Other more sophisticated approaches use word embeddings or language models to vectorize using the semantic of the author's messages. Section 2 shows how our approach handles this step.

Once the dataset is in the (X, y) form, we need to learn from these examples to obtain a predicting model. There are different types of machine learning models used for the Author Profiling task [21]. For instance, we evaluate our approach with Naïve Bayes, K-nearest neighbors, Support Vector Machines, Logistic Regression, and Gradient Boosting, among others. The details about these methods are studied in [22, 23] and the precise implementation is documented in [24].

[4] https://fasttext.cc/

Our contribution

This notebook tackles the problem of profiling *Hate Speech Spreaders*, in Spanish and English languages, through its published messages in social media. In particular, we focus on the homonymous task in PAN @ CLEF21 for benchmarking our model. Each dataset, English, and Spanish, contains 200 cases for each of the two languages, each dataset contains 200 different authors, and each author is represented by two hundred messages. Furthermore, each author is labeled as a hate speech spreader or not.

Our approach is designed to be explainable through the projection of users into an Emotion-space induced by EmoLex. It can support variations of the same concepts and lexical variations of the same word based on semantic representations with the help of fastText. Multilingual support for Spanish and English languages is straightforward since translations of EmoLex, and existing pre-trained models of fastText for Spanish and English languages. We test our emotion-based encoding with seven different classifiers and provide a brief statistical analysis in the experimental results section. Section 2 details our modeling approach.

Roadmap

This section contextualizes our participation in PAN @ CLEF21. Section 2 details the construction and prediction stages of our model. The experimental setup and results are described in sections 3 and 4. Our final comments and conclusions are given in Section 5.

2. Emotion-based modeling of users

Our model is pretty general and straightforward. However, its construction needs a set of Emotion-prototypes based on the emotion lexicon and word embeddings. We use the EmoLex and pre-trained FastText embeddings for both Spanish and English languages. Figure 1 illustrates the general flow of the prototype's computation.

First, the procedure segments words per emotion; in the case of the EmoLex, it associates each word with ten emotions and sentiments: anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, and trust. Note that words linked with several emotions will be linked with such emotions. The emotion sub-lexicons are then used to create a unique vector prototype that summarizes the emotion using word embeddings of individual words, using the companion fastText word-embedding model. These emotion-prototypes are stored as $P = \{p_1, p_2, \cdots, p_{10}\}$, and will be used to encode user's messages and be able to train the models and predict new instances.

Once emotion-prototypes are created, we can train and predict. Figure 2 shows the flow that transforms an author into an emotion vector. Each author is represented as the sequence of its messages; these messages are plain text normalized and partitioned into a list of tokens. This procedure removes diacritic symbols, punctuation signs, duplicate letters, stop words, URLs, and emojis in this step. Along with text normalization, all user mentions are normalized to *USER*. Messages are then tokenized as unigrams. Finally, these unigrams are used to create a sequence that will be used to create a 300-dimensional vector.

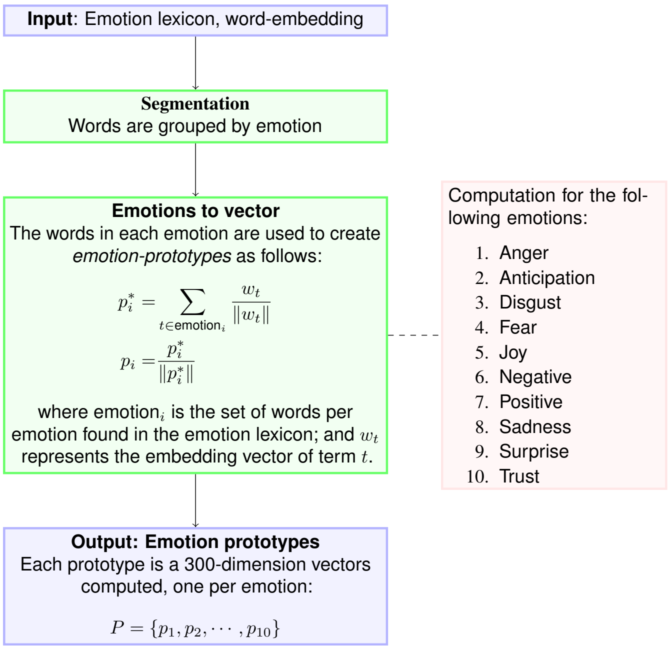

Input: Emotion lexicon, word-embedding
Segmentation
Words are grouped by emotion
Emotions to vector
The words in each emotion are used to create *emotion-prototypes* as follows:
$p_i^* = \sum_{t \in \text{emotion}_i} \frac{w_t}{\|w_t\|}$
$p_i = \frac{p_i^*}{\|p_i^*\|}$
where emotion$_i$ is the set of words per emotion found in the emotion lexicon; and w_t represents the embedding vector of term t.
Output: Emotion prototypes
Each prototype is a 300-dimension vectors computed, one per emotion:
$P = \{p_1, p_2, \cdots, p_{10}\}$
Computation for the following emotions:
1. Anger
2. Anticipation
3. Disgust
4. Fear
5. Joy
6. Negative
7. Positive
8. Sadness
9. Surprise
10. Trust

Figure 1: Computing emotion-prototypes for our modeling.

The emotion-prototypes are used to map user's messages to an emotion-space, using the cosine similarity among prototypes and user vectors. First, we need to transform the sequence of words into a single vector using the companion word-embedding; then, we will use this vector and vector prototypes P to create a 10-dimensional vector used by a classifier at training and prediction stages.

3. Experimental setup

Our implementation of the emotion-space above detailed in §2 is based on a number of well-known libraries in the Python language. NLTK [25] for preprocessing of text messages, Scikit-Learn [24] is used to create features and machine learning algorithms, fastText [26] for mapping Tweets to semantic space. In particular, we use the model pre-trained on 600 billion tokens on Common Crawl for the English task and the one from [27] to tackle the Spanish task. We use the NRC Lexicon [12] (EmoLex) to create lists of representative keywords; it contains a multilanguage pack with support for English and Spanish languages.

We ran our tests on a computer with a 3.4 GHz Quad-core Intel Core i7, 32 GB RAM, and operating system macOS Catalina 10.15.7.

Preprocessing

The data provided by PAN organizers were 200 XML documents for both English and Spanish, each document corresponds to a user and includes 200 tweets written by him.

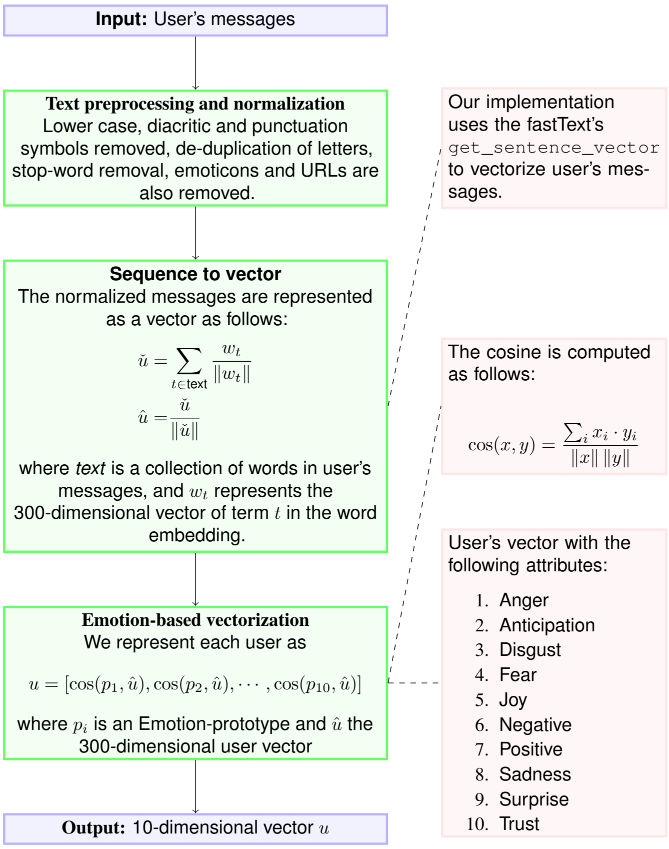

Input: User's messages
Text preprocessing and normalization
Lower case, diacritic and punctuation symbols removed, de-duplication of letters, stop-word removal, emoticons and URLs are also removed.
Sequence to vector
The normalized messages are represented as a vector as follows:
$\check{u} = \sum_{t \in \text{text}} \frac{w_t}{\|w_t\|}$
$\hat{u} = \frac{\check{u}}{\|\check{u}\|}$
where *text* is a collection of words in user's messages, and w_t represents the 300-dimensional vector of term t in the word embedding.
Emotion-based vectorization
We represent each user as
$u = [\cos(p_1, \hat{u}), \cos(p_2, \hat{u}), \cdots, \cos(p_{10}, \hat{u})]$
where p_i is an Emotion-prototype and $\hat{u}$ the 300-dimensional user vector
Output: 10-dimensional vector u
Our implementation uses the fastText's `get_sentence_vector` to vectorize user's messages.
The cosine is computed as follows:
$\cos(x, y) = \frac{\sum_i x_i \cdot y_i}{\|x\| \, \|y\|}$
User's vector with the following attributes:
1. Anger
2. Anticipation
3. Disgust
4. Fear
5. Joy
6. Negative
7. Positive
8. Sadness
9. Surprise
10. Trust

Figure 2: Projecting users into the emotion-space.

We create the emotion-prototypes and project our datasets as described in §2. The preprocessing of texts is performed with the help of NLTK[5] package. Therefore, we characterize every user with a 10-dimensional vector, and this representation is the input for the machine learning models at training and prediction stages.

Model selection

In order to participate in the PAN contest, we develop several models using the training data provided by the organizers. We split the original training dataset into two subsets to test our models. We randomly assign 70%/30% from the training dataset to training/test splits, each of which forms a balanced subset.

We consider several classifiers for creating our emotion-based models. More precisely, we use Naive Bayes (NB), K-Nearest Neighbor (KNN), both linear and non-linear Support Vector Machine (SVM), Nearest Centroid (NC), Logistic Regression (LR), and Gradient Boosting (GB) from the Scikit-Learn package. We ran a hyperparameter optimization process to select those

[5] https://www.nltk.org/

Table 1
Grid-searched hyper-parameters for the used machine learning models

Model	**Model Hyperparameters**	
	Name (*Scikit-learn parameter name*)	**Values**
LinearSVM	Iterations (*max_iter*)	5000
	Random number generator (*random_state*)	42
KNN	Number of neighbors (*n_neighbors*)	{1, 2, 3, 4, 5, 6, 7}
	Power (*p*)	{1, 2, 5}
	Weight function (*weights*)	{uniform, distance}
	Algorithm to compute NN (*algorithm*)	{auto, ball_tree, kd_tree, brute}
	Leaf size (*leaf_size*)	{10, 20, 30, 50}
SVM	Regularization coefficient (*C*)	{0.1, 1, 10, 100, 1000}
	Kernel type (*kernel*)	{linear, rbf}
	Kernel coefficient for rbf (*gamma*)	{1, 0.1, 0.01, 0.001, 0.0001}
NB	Stability (*var_smoothing*)	$1 \dots 10^{-9}$ with 100 equally spaced points
NC	Threshold for shrinking centroids (*shrink_threshold*)	$0 \dots 1$ with linear steps of 0.01
	Distance (*metric*)	{euclidean, manhattan}
GB	Number of estimators (*n_estimators*)	{10,100,1000}
	Learning rate (*learning_rate*)	{0.001, 0.01, 0.1}
	Subsample ratio (*subsample*)	{0.5, 0.7, 1.0}
	Maximum depth of a tree (*max_depth*)	{3, 7, 9}

models that perform the best. Nonetheless, we used default parameters for Logistic Regression. Table 1 lists the parameter grid used for the model selection.

We used grid search on the mentioned space for the search process and weighted each model using a 5-fold, 3-repetition stratified k-fold cross-validation. We chose the parameter combination in each language that had the highest accuracy during the cross-validation for the final selection. Thus, the final models were fitted on the entire training set. The best hyperparameters are summarized in Table 2.

After every learning algorithm fit a model for each language, we evaluate them using the test subset to predict the class labels, obtaining an estimate of how well these models perform on unseen data. The accuracies achieved are shown later in Table 3. Finally, based on the test results, the ML models selected to participate in the Hate Speech Spreader task were chosen.

4. Experimental results

This section presents the experimental results of our approach for the PAN @ CLEF21 Hate Speech Spreader Profiling task. As commented, we divided the dataset to perform model selection, and therefore we show the results for this internal process and the results in the official gold standard.

Figure 3 shows the performance of model evaluation using 5-fold with 3-repetition stratified k-fold cross-validation for both English and Spanish. The classifiers SVM for English and LSVM for Spanish evidence that their data points consistently hover around the center values; thus, the predictions will have less variation. Likewise, these two models have competitive medians

Table 2
The best hyperparameters for the machine learning models

Language	Model	Hyperparameters

EN	LinearSVM	max_iter = 5000
		random_state = 42

	KNN	n_neighbors = 6
		p = 5
		leaf_size = 10

	SVM	C = 1000
		kernel = rbf
		gamma = 1

	NB	var_smoothing = 0.8111308307

	NC	shrink_threshold = 0.0
		metric = manhattan

	GB	n_estimators = 1000
		learning_rate = 0.001
		subsample = 0.5

ES	LinearSVM	max_iter = 5000
		random_state = 42

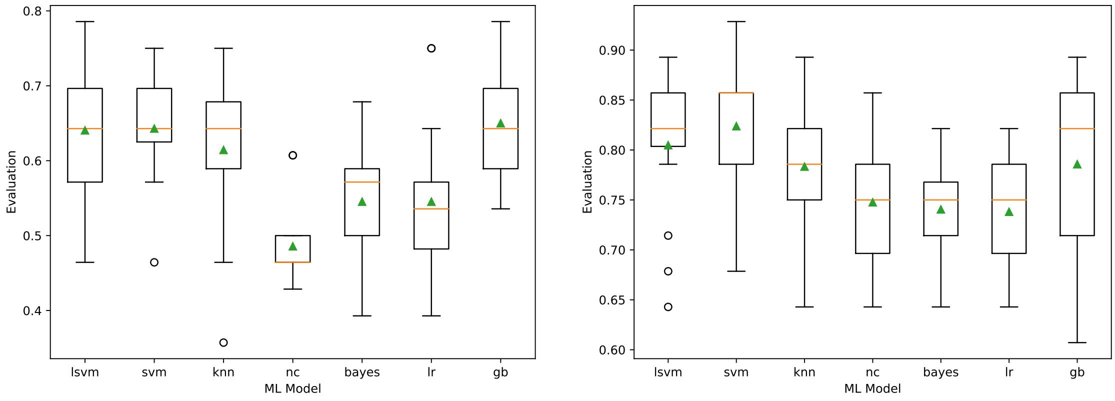
	KNN	n_neighbors = 7
		p = 1
		leaf_size = 10

	SVM	C = 100
		gamma = 1

	NB	var_smoothing = 0.2310129700
	NC	shrink_threshold = 0.93

	GB	learning_rate = 0.001
		subsample = 0.5

comparing to the others.



(a) English language

(b) Spanish language

Figure 3: Accuracy distribution of our models using the cross-validation partitions. The higher, the better.

The performance for the machine learning models is shown in Table 3. Note that the highest accuracy during the cross-validation is SVM with RBF kernel for English and LinearSVC for Spanish. Subsequently, we choose these two models based on the model evaluation (see Fig. 3) and the accuracy from Table 3.

Table 3
Accuracy for the different classifiers applied to our emotion-based author encodings. The higher, the better.

Model	**Accuracy**	
	EN	**ES**
Linear SVC	71%	**73%**
SVM rbf	**78%**	70%
KNN	73%	66%
NC	60%	71%
NB	61%	65%
LR	60%	72%
GB	75%	67%

Table 4
Accuracies achieved during the Cross-Validation process and on the test set

Model (language)	**C-V (training set)**	**PAN@CLEF21 (test set)**
SVM RBF (EN)	**78%**	62%
LSVM (ES)	73%	**78%**
Average	75%	70%

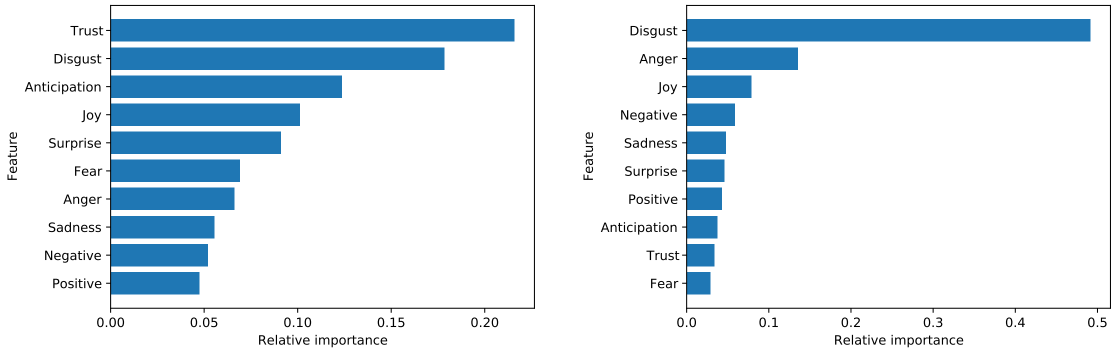
The accuracy of our models was 5% lower on the official test set compared to the cross-validation results as shown in Table 4.

Analysis of emotion-based hate speech spreader models

Gradient boosting is an ensemble of decision trees that, instead of accessing data through a kernel function, accesses attributes directly. Decision trees require the computation of the importance of each attribute as part of its construction algorithm; it is feasible to use it outside of the decision tree context to obtain insight into the problem. In particular, we use the Gini importance [24] to measure the influence of each attribute on the final decision.

Figure 4 shows the importance of each attribute, as seen by our models. For example, in the case of English, Fig. 4a, there is some remarkable difference between the fourth most important and the rest of them. Trust, Disgust, Anticipation, and Joy are the most critical predictors in English, highlighting Trust and Disgust. On the other hand, Fig. 4b shows the attribute importance for the Spanish language; Disgust dominates the other features, clearly standing out for determining hate speech spreaders. Our Spanish modeling closely mimics Plutchik's classification of emotions [28] that link hatred to three primary emotions: Disgust, Anger, and Fear.

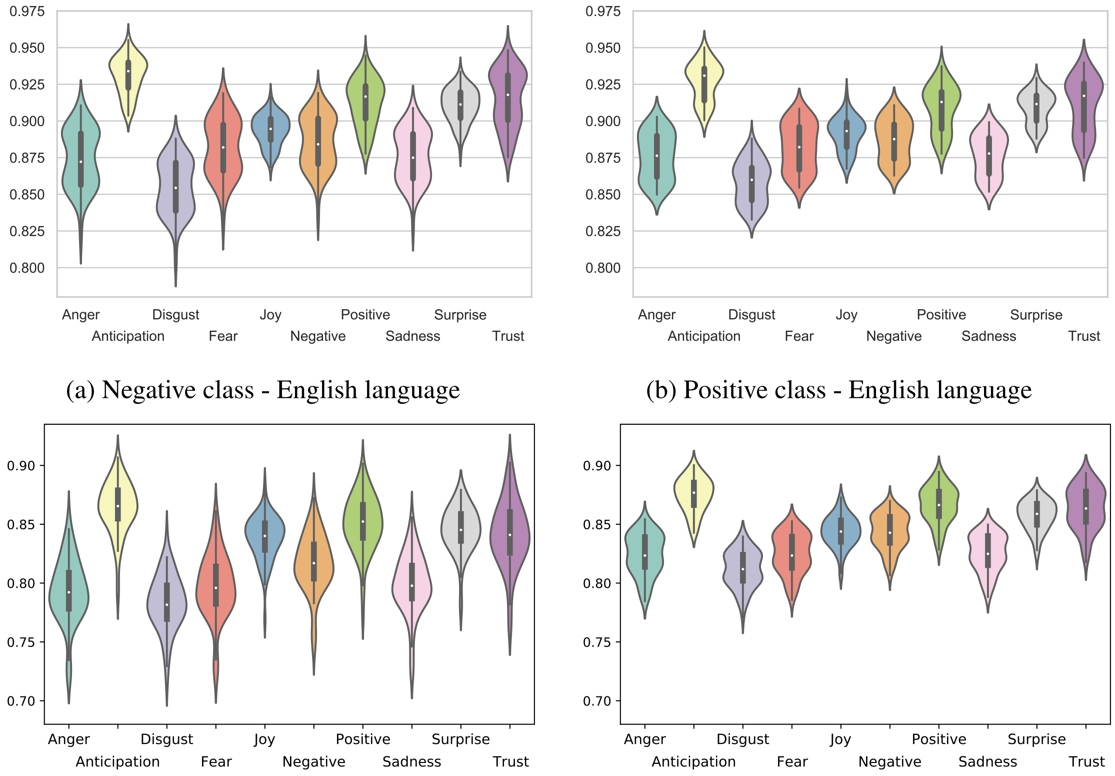
Figure 5 shows the emotion and sentiment distributions per class, computed on the PAN @ CLEF21 training set. The top row shows distributions for the English language. Here we observe that *no hate speech spreaders* (negative examples) have large variations in their emotions as



(a) English language

(b) Spanish language

Figure 4: Feature importance of the gradient boosting decision tree as applied to our emotion-based author's encoding. The higher, the most important.



(a) Negative class - English language

(b) Positive class - English language

(c) Negative class - Spanish language

(d) Positive class - Spanish language

Figure 5: Distribution of the emotion-space (i.e., computed with the emotion projection procedure, see Fig. 2) for both English and Spanish languages.

compared with those in the positive class. Note that several median values (white point in the box inside the violin shape) also dramatically moves. This effect is easily noticeable by those features with the highest Gini importance, see Fig. 4a. Figures 5c and 5d illustrates how emotions distributes for the Spanish language writers for negative and positive classes. Again, we observe that mass concentrates around the median for positive class; we can also observe a noticeable difference, the median in some attributes like Anger, disgust, and fear; this could be associated with a higher emotional charge in the Spanish language Hate Speech spreaders.

5. Conclusions

This paper proposes Semantic Emotion-based models on both Spanish and English languages to cope with the Profiling Hate Speech Spreaders challenge at PAN @ CLEF21. Our approach was designed to be explainable through the projection of users into an Emotion-space induced by EmoLex, supporting lexical variations of the same word based on semantic representations with the help of word embeddings such as fastText.

We conducted a broad model selection study to get the best performing algorithms for our approach. In this sense, we selected SVM with RBF kernel for English and Linear SVC for Spanish as our emotional-based models. Unfortunately, the accuracy of our models was 5% lower on the test set compared to the cross-validation results.

There is still room to improve our work in the future; our next steps in the research include exploring the use of different word embeddings and lexicons with different emotion classifications to improve the overall effectiveness of the classification process. Due to our emotion-centered modeling, we noticed that the Spanish model closely mimics Plutchik's classification of emotions for hatred from our experiments. This behavior requires a more profound exploration to describe its effect on a more fine scale.

References

[1] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. Rangel, P. Rosso, M. Sanguinetti, Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), Association for Computational Linguistics, 2019.

[2] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, c. Çöltekin, SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020), in: Proceedings of SemEval, 2020.

[3] F. Rangel, Author profile in social media: Identifying information about gender, age, emotions and beyond, in: Proceedings of the 5th BCS IRSG Symposium on Future Directions in Information Access, 2013, p. 58–60.

[4] F. M. R. Pardo, P. Rosso, Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter., in: CLEF (Working Notes), volume 2380 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019.

[5] L. Bing, Sentiment analysis: mining opinions, sentiments, and emotions, Cambridge University Press, 2015.

[6] M. E. Aragón, A. P. López-Monroy, L. C. González, M. Montes-y-Gómez, Attention to emotions: Detecting mental disorders in social media, in: TSD: International Conference on Text, Speech, and Dialogue, volume 12284, Springer, Cham, 2020, pp. 231–239.

[7] E. S. Tellez, D. Moctezuma, S. Miranda-Jiménez, M. Graff, An automated text categorization framework based on hyperparameter optimization, Knowledge-Based Systems 149 (2018) 110–123.

[8] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[9] F. Rangel, G. L. D. L. P. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[10] F. A. Nielsen, A new anew: Evaluation of a word list for sentiment analysis in microblogs., in: M. Rowe, M. Stankovic, A.-S. Dadzie, M. Hardey (Eds.), MSM, volume 718 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011, pp. 93–98.

[11] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 168–177.

[12] S. M. Mohammad, P. D. Turney, Crowdsourcing a word-emotion association lexicon, Computational Intelligence 29 (2013) 436–465.

[13] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[14] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv:1706.03762 (2017).

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[18] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).

[19] S. Ashraf, O. Javed, M. Adeel, H. Iqbal, R. M. A. Nawab, Bots and gender prediction using language independent stylometry-based approach., in: CLEF (Working Notes), 2019.

[20] F. Rangel, P. Rosso, M. Potthast, B. Stein, Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter, Working notes papers of the CLEF (2017) 1613–0073.

[21] J. Bevendorff, B. Ghanem, A. Giachanou, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. M. R. Pardo, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Wiegmann,

E. Zangerle, Overview of PAN 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on twitter, and style change detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22-25, 2020, Proceedings, volume 12260 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 372–383.

[22] G. James, D. Witten, T. Hastie, R. Tibshirani, An introduction to statistical learning, volume 112, Springer, 2013.

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

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.

[25] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.

[26] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, Fasttext.zip: Compressing text classification models, CoRR abs/1612.03651 (2016). URL: http://arxiv.org/abs/1612.03651.

[27] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[28] R. Plutchik, The emotions: Facts, theories and a new model., American Journal of Psychology 77 (1964) 518.