

# Deep Modeling of Latent Representations for Twitter Profiles on Hate Speech Spreaders Identification Task

Notebook for PAN at CLEF 2021

Roberto Labadie Tamayo<sup>1</sup>, Daniel Castro Castro<sup>1</sup> and Reynier Ortega Bueno<sup>2</sup>

<sup>1</sup>Universidad de Oriente, Cuba

<sup>2</sup>PRHLT Research Center, Universitat Politècnica de València, Valencia Spain

## Abstract

In this paper, we describe the system proposed by UO-UPV team for addressing the task *Profiling Hate Speech Spreaders on Twitter* shared at PAN 2021. The system relies on a modular architecture, combining Deep Learning models with an introduced variant of the Impostor Method (IM). It receives a single profile composed of a fixed quantity of tweets. These posts are encoded as dense feature vectors using a fine-tuned transformer model and later combined to represent the whole profile. For classifying a new profile as hate speech spreader or not, it is compared by a similarity function with the Impostor Method with respect to random sampled prototypical profiles. In the final evaluation phase, our model achieved 74% and 82% of accuracy for English and Spanish languages respectively, ranking our team at 2<sup>nd</sup> position and giving a starting point for further improvements.

## Keywords

Deep Impostor Method, Spectral Graph Convolutional Neural Network, Prototypes, Transformers

## 1. Introduction

In the last decade, social media has gathered in the same place people with a wide diversity of interests and psychological characteristics, which is a great achievement from many points of view. However, due to the reluctance of some users to accept precisely this diversity, a huge amount of toxic content attacking immigrants and other minorities has raised, giving to the *hate speech spreading* phenomenon [1, 2] an influence that in the most extreme cases contributed to some kinds of social violence.

To face this, it is important to identify and censure, in the worst cases, users who can be considered as hate spreaders. To this end, social media platforms rely mainly on users' reports and content moderators, which are burdened by the amount or the content of posts. Also, Artificial Intelligence algorithms are employed, but in most cases, these algorithms are focused on identifying toxic content on isolated messages rather than handle and classifying a whole user profile based on its previous posts.

The real-time microblogging platform Twitter is one of the main scenarios where users share textual content mainly characterized for being short and informal. This platform has reached


---

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

✉ rlabadiet@gmail.com (R. Labadie Tamayo); danielcc@uo.edu.cu (D. Castro Castro); rortega@prhlt.upv.es (R. Ortega Bueno)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

192 million users<sup>1</sup>, which makes it reasonable to study the behavior of these users to avoid the spreading of toxic and hateful messages.

Traditional tasks as gender, age groups and native language detection are profiling tasks that have been well studied in the field of Author Profiling (AP) [3, 4]. Recently alongside the growth of misinformation and hate speech spreading phenomena in social media have stood out studies in the Author Profiling field related to identifying psycho-social characteristics of authors [5, 6]. This involves detecting highly subjective characteristics for instance; offensiveness, toxicity, or any psychological pattern of communication, which added to the unstructured kind of information from texts, increases the complexity w.r.t. traditional AP tasks.

The AP task proposed at PAN 2021 [7]: “Profiling Hate Speech Spreaders on Twitter” [8] aims to determine, given 200 tweets from a user profile, whether the author is a hate speech spreader or not. Also, it is deployed from a multilingual perspective, evaluating the detection of hateful speech in both English and Spanish languages.

Many efforts have been directed to the task of predicting computationally the existence of hateful or offensive forms of communications on isolated messages. The most noticeable results have been achieved employing Deep Learning (DL) methods like Recurrent Neural Networks [9] or more recently, fine-tuned transformer models [10] and conventional Machine Learning models like Random Forest for combining deep representations [11]. In the same way, modeling techniques for representing and classifying author profiles on AP tasks have employed representations ranging from single-dense or statistical-based vectors to sequences [12, 13].

In this working notes we introduce our model for participating in the “Profiling Hate Speech Spreaders on Twitter” task and we study the proposed classifier performance by exploring different ways for modeling the author profiles. The architecture combines DL techniques with traditional ones from Machine Learning, specifically, it consists of a Sentence Encoder Module, based on a transformer architecture for obtaining abstract tweets representations. These encodings are employed for modeling the user’s profile, which is fed into a Prediction Model based on the Impostor Method. The source code of our approach is available on GitHub<sup>2</sup>.

The paper is organized as follows: in Section 2 we briefly introduce a description of the datasets employed. Section 3 presents the system’s architecture and provides details about its modules. Section 4 describes the experiments and the achieved results. Finally, we present our conclusions and provide some directions that we plan to explore in future work.

## 2. Tasks and Datasets

In the classification task proposed as Profiling Hate Speech Spreaders, it is only provided a training set [14] composed by 200 examples of Twitter accounts for both English and Spanish languages. These training examples are uniformly distributed attending to hate spreader and no hate spreader classes and for each author’s profile a set of 200 tweets is provided. It is worth to notice that the individual tweets are not annotated regarding the presence of hateful content. In this work we also used an additional dataset from the training set of *SemEval-2019 Task 5*:

---

<sup>1</sup><https://www.oberlo.com/blog/twitter-statistics>

<sup>2</sup><https://github.com/labadier/hatespeechspread>

*Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter* [15] containing 9000 tweets for the English language with 3783 annotated as hateful, whereas for the Spanish language, it is composed by 4500 tweets with 1857 annotated as hateful messages. In this dataset for the annotation process the absence of hate speech or not on each example was considered only attending to the immigrants and women targets.

### 3. System Overview

For our system we decomposed the task of classifying a whole profile into a modular problem with three phases: i) constructing a representation that captures hate speech-related features from an individual tweet, ii) combining the representations of the messages from the profile, iii) determining whether the profile corresponds to a hate speech spreader or not. In this section, we describe the modules in charge of each phase of the problem.

#### 3.1. Tweets Encoder

For encoding the messages posted by the authors in such a way that explicit and long-term relations expressing hate could be detected and condensed into a single vector, we fine-tuned Transformers Models (TM) [16] by using the HuggingFace Transformers library<sup>3</sup>. Specifically, we employed the BERTweet [17] and BETO [18] models for the English and Spanish languages respectively. Their configuration is the same as the BERT-base [19] model and their training is based on the RoBERTa pre-training procedure [20] using a corpus of English tweets for BERTweet, and texts from other sources such as Spanish wikis, OpenSubtitles and ParaCrawl<sup>4</sup> for BETO.

We placed an intermediate classification task to fine-tune the Tweets Encoder which consists on predicting how likely to express hate a message is. For this, we stacked to the TM an intermediate layer as a bottleneck, for extracting a representation of the input message in a latent space. This intermediate layer is fed with the first token of the output sequence from the TM. Then, this encoding is passed to an output neuron which makes the prediction for the targeted task.

In previous works [21, 22] employing the strategy proposed in the Universal Language Model Fine-Tuning (ULMFiT) [23] for tuning pre-trained models in a gradual unfreezing-discriminative fashion, has outperformed the standard schema for fine-tuning TMs on sentiment analysis tasks. Taking this into account we decided to set a different learning rate for each encoder layer in the TM, increasing it while the neural network gets deeper, i.e  $\alpha_i = \lambda_i \alpha_0$  and  $\lambda_i = \lambda_{i-1} + 0.1$ , where  $\alpha_i$  is the learning rate of the  $i^{th}$  layer,  $\lambda_i$  is a multiplier to compute  $\alpha_i$  from  $\alpha_0$ . The shallower layer that receives the input message has a fixed  $\alpha_0$  and  $\lambda_0 = 1$ . This dynamic learning rate keeps most information from the pre-training at shallow layers and biases the deeper ones to learn about the hate detection task. The training data from *SemEval-2019* [15] was employed for this fine-tuning process.

---

<sup>3</sup><https://huggingface.co/transformers>

<sup>4</sup><https://github.com/josecannete/spanish-corpora>

## 3.2. Profile Modeling

The analysis of profiles' history containing such amount of textual information as the ones involved in this task, may difficult the use of DL models in terms of the sequence length resulting from treating all the textual data at once, this occurs independently if we are addressing a supervised or unsupervised task.

Therefore, we evaluate to model the profiles blending the dense representations obtained from the Tweets Encoder in three different ways. In the first one, we model the profile with a graph-based representation and combine the information from all the 200 tweets in the account by means of a Graph Convolutional Neural Network [24, 25].

The second one treats the set of vectors computed by the Tweets Encoder from the profile's posts as a sequence and the information is condensed employing an attention-based model. Finally, the third one averages the encoding of the 200 tweets in the profile (AVG-Method).

### 3.2.1. Graph Based Profile Modeling

In social media, hate sometimes is spread in a scattered manner i.e., the hateful content of an idea can be constructed by relating the information from different posts. Also, the profiles within the data are not structured following any temporal order, for this, we may be interested in sharing the information from one tweet to the others while its individual information is being transformed to express how it belongs to its context (i.e., the profile taking into account the targeted task). Graph Neural Networks are specialized in learning patterns from this kind of unstructured representation of relations.

We model the account with graph-based representation, where every node is a tweet, and each of them is connected to the others. Then, this graph is processed by a Spectral Graph Convolutional Neural Network (SGN).

The fact that every node in the modeled graph is connected to all the other, makes that after one step of message passing, an individual node has knowledge about every node in the graph. We employ the convolution operator proposed in [25] defined as:

$$X' = ReLU(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X \Theta) \quad (1)$$

Where  $X$  is the matrix of vector representations of the nodes,  $\hat{A} = A + I$  is the adjacency matrix  $A$  added to the identity matrix  $I$  what involves that self-loops are introduced to our graph-based representation. And  $D$  is a diagonal matrix containing the degree of the  $i^{th}$  node (i.e.  $D_{ii} = 200$  in our case due to the graph completeness). Finally  $\Theta$  is the matrix of learnable parameters for the message passing function. In the node-wise formulation this is given by:

$$x'_i = ReLU \left( \Theta \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} x_j \right) \quad (2)$$

Here,  $x_i$  represents the encoding of the  $i^{th}$  node,  $d_i$  the degree of the  $i^{th}$  node,  $\mathcal{N}(i)$  is the set of neighbor nodes to  $i$  and  $\Theta$  is the matrix of learnable parameters. As we can see, in this convolutional schema the information is shared simply by a normalized sum before computing

the new  $x'_i$  encoding by  $\Theta$ .

For our convolutional net we repeat this message passing and updating processes through two convolutional layers. After the graph is convolved the nodes information are combined by a mean-pool layer and fed into a dense layer which condense the graph information. The outcome of this layer is considered the new profile’s modeling.

The model was trained by using the annotation provided in the competition for trying to predict whether the user is a hate spreader.

### 3.2.2. Sequence Based Profile Modeling

For this approach, we consider the whole profile as a sort of sequence, where each encoded tweet is a token whose information is contextualized by means of an Additive Attention-based Fully Connected Neural Network (Att-FCNN). As in this sequence the order of the elements is not relevant, we get rid of temporal dependencies possibly miscaptured with sequence-specialized models, by avoiding the sum of a positional encoding [16] to the tokens vector. For training this model, as in SGN, we employ the classification task of predicting if the account corresponds to a hate spreader.

At first, the tokens of the input sequence are related and weighted through the Attention Mechanism proposed in [26] to learn which representation has a stronger impact regarding determining if the user is a hate spreader. The output of this attention module is reduced to a single vector by adding together all the tokens’ weighted-vector, then it is fed into a dense layer, whose outcome is our new profile modeling. The output of the model is computed by feeding this encoding into a neuron, which predicts the odds of the user spreading a hateful speech.

### 3.3. Deep Impostor Method

An interesting fact about DL approaches is despite they are powerful at detecting abstract features that allow partitioning the space of representations into different classes, when the available data in the training stage is not enough their performance is affected. This is precisely the reason why we decided to employ the above-described models (i.e., SGC and Att-FCNN models) just to find a representation in a latent space of the profiles. Instead, for make predictions, we propose our Deep Impostor Method (DIM) based on the Impostor Method [27].

As in the original method, for making a prediction about an unknown object (in this case a author’s profile) we must have previously defined a set  $H$  and  $K$  of positive and negative examples respectively and the unknown object  $u$ .

We assume that our  $u$  is a hate spreader and from the set  $H$  we sample with a uniform fashion a subset  $\bar{H}$  as prototypes of the positive class and analyze for each  $\bar{H}_i$  if  $u$  is more similar to this prototype than to the elements of a set  $\bar{K}_i$  (i.e.,  $\mathcal{F}(u, \bar{H}_i) > \mathcal{F}(u, \bar{K}_{ij})$  where  $\mathcal{F}$  is a similarity function; in our case the cosine similarity). After this, we say that  $u$  taking into account  $\bar{H}_i$  is a candidate to be a hate spreader by majority voting, this is:

$$P_i(u, \bar{H}_i) = \begin{cases} 1 & \text{if } \sum_j^{|\bar{K}_i|} [\mathcal{F}(u, \bar{H}_i) > \mathcal{F}(u, \bar{K}_{ij})] > \frac{|\bar{K}_i|}{2} \\ 0 & \text{otherwise} \end{cases}$$

Since the features of any object is learned by means of a DL model, we skip the feature selection step exposed in the original method, because removing indiscriminately some of them, may result in destroying the similarity relations between elements from the same or different classes learned by the DL model.

After computed each  $P_i$ , we define the profile  $u$  as a hate spreader following the rule:

$$\hat{y}(u) = \begin{cases} 1 & \text{if } \sum_i^{|\bar{H}|} P_i(u, \bar{H}_i) > \frac{|\bar{H}|}{2} \\ 0 & \text{otherwise} \end{cases}$$

We set empirically the size of sampled sets  $\bar{H}$  and  $\bar{K}_i$ , in relation to the original size of  $H$  and  $K$  in the training set provided by the organizers. That is, for the Spanish language  $|\bar{H}| = 0.45|H|$  and  $|\bar{K}_i| = 0.45|K|$ , whereas for the English language they were set as  $|\bar{H}| = 0.4|H|$  and  $|\bar{K}_i| = 0.4|K|$ .

## 4. Experiments and Results

In this section, we present the experiments conducted in the tuning process of each module, as well as the results obtained by the best strategies on the test dataset. Results regarding the official test set were obtained by running our models on TIRA environment [28]. The metric employed for evaluating the models was accuracy, proposed by the task organizers.

### Tweets Encoder

By attaining a good performance on the intermediate task for training the Tweets Encoders, we can obtain a better set of hate-related features for each tweet and increase the quality of their aggregation when the profile is modeled. We explored some strategies directed to improve the Encoder performance. The first one was employing Adapters [29] on the TMs as a way to unify the Tweets Encoders into a Multilingual model (*Adapter*). Also, we explored using the main ideas from ULMFiT without Adapters (*Dynamic*), and finally, we tried to use just a straightforward fine-tuning (*Standard*).

**Table 1**

Fine-Tuning strategies for hate speech detection on SemEval 2019 dataset

Strategy	Language		Average
	EN	ES	
Standard	0.794	0.783	0.789
Dynamic	0.848	0.872	0.860
Adapters	0.617	0.691	0.654

As can be observed in Table 1 showing the results of validating with 20% of the data, the best-performed strategy was employing the ideas from ULMFiT. On this model, we tuned the size of the stacked intermediate layers, as well as the learning rate ( $lr$ ), resulting in the

best-performed combination setting 64 neurons for the intermediate layer and an initial lr of 1e-5. The parameters of these models were optimized with the RMSprop method.

## Profile Modeling

As described in Section 3.2, we employed two main paradigms based on learning methods to relate all tweets’ extracted features and represent the whole profiles. We made some variations for these methods and we evaluated them with 5-folds cross-validation (CV) on every language. Also, we compared the performance of these models with the one proposed in [13] employing an Attention BiLSTM based model (Att-BiLSTM). In table Table 2 is shown the performance of these models for the cross-validation evaluation and the official test set.

**Table 2**

Performance of Profile Modeling modules under Profiling Hate Speech Spreader on Twitter task

Data	Language	Deep Model			
		SGCN-2	SGCN-3	Att-FCNN	Att-BiLSTM
CV	English	0.76	0.76	0.75	0.77
	Spanish	0.83	0.75	0.88	0.82
	AVG	0.795	0.755	0.815	0.795
Test	English	0.49	0.51	0.73	0.79
	Spanish	0.59	0.51	0.81	0.74
	AVG	0.54	0.51	0.77	0.765

Here, SGCN-2 and SGCN-3 refer to the Spectral Graph Networks described in Section 3.2, staking two and three convolutional layers respectively. As can be observed, some of these models, especially the graph-based one, had a poor performance for the test set in comparison with the CV evaluation, resulting in over-fitting. This is possibly caused because the density of the constructed graphs makes that in the message passing from one neighbor to another some important features vanish, especially if the profile belongs to a hate speech spreader but the hateful messages are just a few in the account. Therefore, may result convenient performing a more sophisticated construction of these graphs where only be joint the more similar tweets belonging to the same class according to the Tweets Encoder predictions. Nevertheless, we also must consider that in terms of profiles, the amount of examples for each class is not too large, which makes it difficult for the generalization capability gain in the training process of these DL models.

In this module the learnable parameters of every tested model were optimized with the Adam Optimization Method.

## Classification Method

For the DIM, naturally, we tuned the profile modeling to be employed, and in exhaustive fashion for each of these modelings, the proportion of the positive and negative classes represented by  $\bar{H}$  and  $\bar{K}_i$  respectively. The best results for each model regarding this experimental process are shown in Table 3.

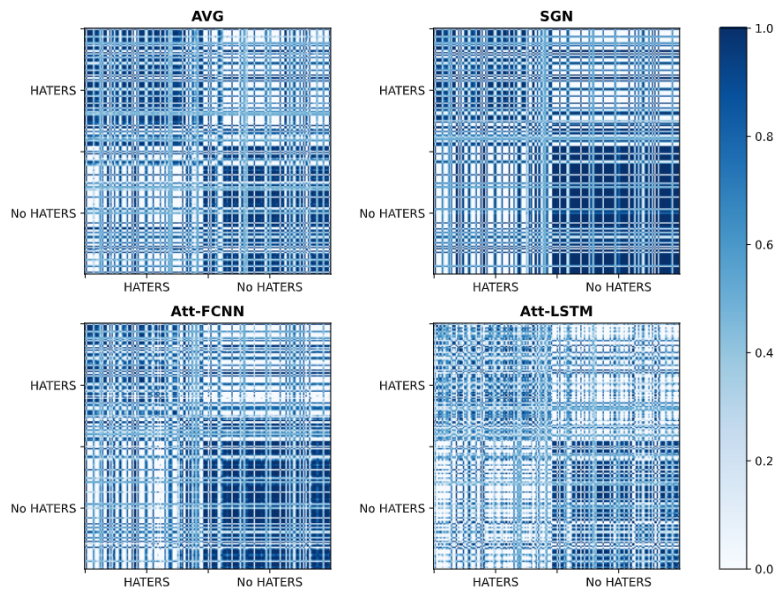


**Table 3**  
Deep Impostor Method Performance

Data	Language	Profiling Model			
		SGCN	Att-FCNN	Att-BiLSTM	AVG Method
CV	English	0.73	0.72	0.74	0.73
	Spanish	0.76	0.76	0.82	0.78
	AVG	0.745	0.74	0.78	0.755
Test	English	0.72	0.73	0.73	<b>0.74</b>
	Spanish	0.80	<b>0.85</b>	0.79	0.82
	AVG	0.76	<b>0.79</b>	0.76	<b>0.78</b>

At first glance, it can be observed how this machine learning method yields more encouraging and stable results with respect to the individual modules described above. Also, more simplistic methods for representing the whole profile, like AVG-Method, reported better performance. Taking into account that DIM relies on similarity comparisons, this implies that such approaches gave a better representation in terms of intra-profile similarity relations, which can be observed in Figure 1, where on each matrix the first 100 columns and rows correspond to the hate spreader profiles.

Here the AVG-Method and Att-LSTM show the most symmetric behavior. This means that the



**Figure 1:** Similarity Relation Between Profiles for English Language in the Training Set

representation of profiles within the same class are more similar. Nevertheless for the first one the intraclass similarity is more strongly defined.

We also tried to use fixed  $\bar{H}$  and  $\bar{K}_i$  employing the prototype selection method proposed in



[30]. In this method, at first the whole training data is clustered independently of the objects' class. Then, from homogeneous clusters (i.e., all the objects belong to the same class) is defined as prototype the object more similar to the *mean prototype*. For non-homogeneous clusters is defined a major class (i.e., the one most represented within the cluster) then, the *border prototypes* in the cluster are those not in the major class and more similar to elements from the major class. Conversely, the elements of the major class more similar to the computed border prototypes are also considered as border prototypes.

We changed the criterion for comparing the objects from a distance-based to a similarity-based to make the method compatible with our DIM, but the performance did not improve the one achieved by the random strategy. This clearly means that representative elements are not being well selected by this method in terms of similarity relationships and we hypothesize this fact is related to the definition of clusters, which relies on fuzzy c-means clustering algorithm [31]. This one is based on distance relations rather than similarity relations.

Regarding the official Results, our best submission to the “*Profiling Hate Speech Spreaders on Twitter*” shared task achieved 74% and 82% of accuracy for English and Spanish languages respectively and it was performed by combining the TM fine-tuned with the ideas from ULMFiT and the AVG-Method for modeling the profile, setting the proportion represented by  $\bar{H}$  and  $\bar{K}_i$  of the hate spreader and no hate spreader classes to 0.4 and 0.45 for English and Spanish languages respectively. Nevertheless, employing the representation learned by the Att-FCNN yield a better performance as shown on Table 3.

## 5. Conclusion and Future Works

In this paper, we presented our system for addressing the task “*Profiling Hate Speech Spreaders on Twitter*” proposed at PAN 2021, consisting of given a Twitter author profile, determining whether it corresponds to a hate speech spreader or not. We addressed this task analyzing it from a modular standpoint, at first we explored different ways for encoding individual tweets by employing BERT-based models, which have placed the state of the art in many downstream tasks from NLP, specifically in sentiment analysis and hate speech detection. Afterward, we analyzed some techniques for representing a user profile, using the encodings of textual information from the profile's posts. Finally, we presented a classification module based on slight modifications to the Impostor Method. We achieved the best performance of our system in competition by applying fine-tuning with an intermediate task to the Transformer Models, employing the main ideas from the ULMFiT and modeling the whole profile by combining the tweets with a linear operation such as average, reporting a 74% and 82% of accuracy on English and Spanish languages respectively.

To sum up, the main issues of our system are related to obtaining good representations for the entire profiles, taking into account that the graph modeling technique seems to be a promising way to relate such unstructured relations like the ones existing among the posts, we plan to analyze and construct a graph representation that captures stronger relations considering the predictions made by the Twitter Encoder module. Also, for the Deep Impostor Method it would be interesting to explore a more steady prototype selection technique and replace the similarity function based on cosine similarity by an automatic metric learning model, which could be

trained introducing external data for an intermediate task due to the limited amount of data provided in the competition in terms of profiles.

## Acknowledgments

The work of the third author was in the framework of the research project MISMIS-FAKEHATE on MISinformation and MIScommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31), funded by Spanish Ministry of Science and Innovation, and DeepPattern (PROMETEO/2019/121), funded by the Generalitat Valenciana.

## References

- [1] A. Matamoros-Fernández, J. Farkas, Racism, Hate Speech, and Social Media: A Systematic Review and Critique, *Television & New Media* 22 (2021) 205–224. URL: <https://doi.org/10.1177/1527476420982230>. doi:10.1177/1527476420982230. arXiv:<https://doi.org/10.1177/1527476420982230>.
- [2] C. E. Ring, Hate speech in social media: An exploration of the problem and its proposed solutions, Ph.D. thesis, University of Colorado at Boulder, 2013.
- [3] P. Rosso, M. Potthast, B. Stein, E. Stamatatos, F. Rangel, W. Daelemans, Evolution of the pan lab on digital text forensics, in: N. Ferro, C. Peters (Eds.), *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, Springer International Publishing, Cham, 2019, pp. 461–485. URL: [https://doi.org/10.1007/978-3-030-22948-1\\_19](https://doi.org/10.1007/978-3-030-22948-1_19).
- [4] P. Rosso, F. Rangel Pardo, Author Profiling Tracks at FIRE, FIRE 10th Anniversary, *SN Computer Science* 1 (2020). doi:10.1007/s42979-020-0073-1.
- [5] F. Rangel, A. Giachanou, B. Ghanem, P. Rosso, Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névóel (Eds.), *CLEF 2020 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [6] F. Rangel, P. Rosso, Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), *CLEF 2019 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2019. URL: <http://ceur-ws.org/Vol-2380/>.
- [7] 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.
- [8] 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: *CLEF 2021 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2021.
- [9] G. H. Paetzold, M. Zampieri, S. Malmasi, UTFPR at SemEval-2019 Task 5: Hate Speech Identification with Recurrent Neural Networks, in: *Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapo-*

- lis, Minnesota, USA, 2019, pp. 519–523. URL: <https://www.aclweb.org/anthology/S19-2093>. doi:10.18653/v1/S19-2093.
- [10] Y. Zhang, B. Xu, T. Zhao, CN-HIT-MI. T at SemEval-2019 Task 6: Offensive Language Identification Based on BiLSTM with Double Attention, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 564–570.
- [11] A. Seganti, H. Sobol, I. Orlova, H. Kim, J. Staniszewski, T. Krumholz, K. Koziel, NLPR@SRPOL at SemEval-2019 Task 6 and Task 5: Linguistically enhanced deep learning offensive sentence classifier, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 712–721. doi:10.18653/v1/S19-2126.
- [12] R. Dias, I. Paraboni, Combined CNN+RNN Bot and Gender Profiling, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), CLEF 2019 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2019. URL: <http://ceur-ws.org/Vol-2380/>.
- [13] R. Labadie-Tamayo, D. Castro-Castro, R. Ortega-Bueno, Fusing Stylistic Features with Deep-learning Methods for Profiling Fake News Spreader—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névél (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [14] F. Rangel, B. Chulvi, G. L. D. L. Peña, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter, 2021. URL: <https://doi.org/10.5281/zenodo.4603578>. doi:10.5281/zenodo.4603578.
- [15] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, 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, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 54–63. URL: <https://www.aclweb.org/anthology/S19-2007>. doi:10.18653/v1/S19-2007.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, CoRR abs/1706.03762 (2017). URL: <http://arxiv.org/abs/1706.03762>. arXiv:1706.03762.
- [17] D. Q. Nguyen, T. Vu, A. T. Nguyen, BERTweet: A pre-trained language model for English Tweets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020.
- [18] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish Pre-Trained BERT Model and Evaluation Data, in: PML4DC at ICLR 2020, 2020.
- [19] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, CoRR abs/1810.04805 (2018). URL: <http://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, CoRR abs/1907.11692 (2019). URL: <http://arxiv.org/abs/1907.11692>. arXiv:1907.11692.
- [21] D. Palomino, J. Ochoa-Luna, Palomino-Ochoa at SemEval-2020 Task 9: Robust System Based on Transformer for Code-Mixed Sentiment Classification, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 963–967. URL: <https://www.aclweb.org/anthology/>

2020.semeval-1.124.

- [22] R. L. Tamayo, M. J. R. Cisneros, R. O. Bueno, P. Rosso, A Transformer-based Approach for Detecting and Rating Humor and Offense, in: Proceedings of the 15th International Workshop on Semantic Evaluation, 2021.
- [23] J. Howard, S. Ruder, Universal Language Model Fine-tuning for Text Classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. URL: <https://www.aclweb.org/anthology/P18-1031>. doi:10.18653/v1/P18-1031.
- [24] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R. P. Adams, Convolutional Networks on Graphs for Learning Molecular Fingerprints, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 28, Curran Associates, Inc., 2015. URL: <https://proceedings.neurips.cc/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf>.
- [25] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: <https://openreview.net/forum?id=SJU4ayYgl>.
- [26] G. Zheng, S. Mukherjee, X. L. Dong, F. Li, OpenTag: Open Attribute Value Extraction from Product Profiles, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2018). URL: <http://dx.doi.org/10.1145/3219819.3219839>. doi:10.1145/3219819.3219839.
- [27] S. Seidman, Authorship Verification Using the Impostors Method—Notebook for PAN at CLEF 2013, in: P. Forner, R. Navigli, D. Tufis (Eds.), CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain, CEUR-WS.org, 2013. URL: <http://ceur-ws.org/Vol-1179>.
- [28] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1\_5.
- [29] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, I. Gurevych, AdapterHub: A Framework for Adapting Transformers, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 46–54. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.7>.
- [30] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, A New Fast Prototype Selection Method Based on Cluste, Pattern Anal. Appl. 13 (2010) 131–141. URL: <https://doi.org/10.1007/s10044-008-0142-x>. doi:10.1007/s10044-008-0142-x.
- [31] J. Bezdek, R. Ehrlich, W. Full, FCM—the Fuzzy C-Means clustering-algorithm, Computers and Geosciences 10 (1984) 191–203. doi:10.1016/0098-3004(84)90020-7.