Fusing Stylistic Features with Deep-learning Methods for Profiling Fake News Spreaders

Notebook for PAN at CLEF 2020

Roberto Labadie-Tamayo, Daniel Castro-Castro, and Reynier Ortega-Bueno

Universidad de Oriente, Santiago de Cuba, Cuba
roberto.labadie@estudiantes.uo.edu.cu , danielbaldauf@gmail.com ,
reynier.ortega@uo.edu.cu

Abstract False or unverified information spreads just like accurate information on social media platforms, thus possibly going viral and influencing the public opinion and its decisions. Fake news represents one of the most popular forms of false and unverified information, and should be identified as soon as possible for minimizing their dramatic effects. In order to face this challenge, in this paper we describe our system developed for participating in the Author Profiling task: "*Profiling Fake News Spreaders on Twitter*" proposed at the PAN 2020 Forum. Our proposal learns two representations for each tweet in an account's profile. The first one is based on CNN and LSTM nets analyzing the tweets at word level, the second one is learned by using the same architecture without sharing the weights, but at this time, the tweets are analyzed at character-level. These representations are used for modeling the accounts' profiles. Also is conceived for the whole account's profile a general representation based on stylistic features, which contains information about the writing style. Finally, the profiles' representations are given as inputs to our classification model which is based on LSTM neural nets with attention mechanism. Experimental results show that our model achieves encourage results.

1 Introduction

In this actual modern society, a big part of the communication between peoples is established by digital information, using videos, photos, texts, among others. Everyday the amount of digital information generated by each person is growing very fast and the analysis of these data to satisfy the users' needs is also a challenging task. Specifically the analysis of textual data introduces several challenges because texts are unstructured information written in Natural Language (NL), also there are a lot of textual genre and diversity of languages.

Computational applications for e-marketing and Forensic Analysis based on textual information deal with the mentioned diversity of challenges, but also it is important for them to known social demographic characteristics of author's texts in order to direct the extraction of valuable information from textual sources. Author Profiling (AP) task aims at discovering different marks or patterns (linguistic or not) from texts, that allows to identify some characteristics of the authors, e.g. age, sexual gender, personality, etc.

One of the most important evaluation forums to share recent research and ideas for tasks such as Author Profiling and Authorship Detection (AD), can be found in the platform PAN [1]. Specifically, AP task has focused in the last editions on analyzing micro-blogging textual genre due to the importance to known what and how information is spread among users and communities, and which are the profile characteristics of such users.

PAN 2016 profiling task [24] addressed the prediction of age and gender from a cross-genre perspective, using tweets as training samples and a different textual genre for evaluation. The main goal of PAN 2017 profiling [23] was to identify the gender and language variety from Twitter messages. In PAN 2018 profiling task [22] the goal was to address gender identification from a multi-modal perspective considering text and images. The last edition PAN 2019 profiling task [19] focused on determining whether the author of a Twitter feed is a bot or human and to profile the gender for human authors.

There are others efforts in the research community for AP analysis, such as profiling authors from Arabic [21][20][28], Russian [14] and the Mexican variant of the Spanish language [2]. Generally, the most studied languages had been English and Spanish, whereas gender and age are the social demographic characteristics most explored. In addition, Twitter has raising as the most salience textual genre, due to its popularity as social platform.

A bad phenomenon that is not new, but in last years have gained special significance is the spreading of Fake News, due to its dramatic growing in social media and news sources. That is the reason why a lot of recent works try to discover when a text is fake or at least it contains false or unverified information. However, as important as knowing when a text is fake or not, is to verify when a source generates and spreads fake information or misinformation, for example, a company that promotes its product without the quality promoted, false data published in an election campaign, etc. These last scenarios could be analyzed with the use of Author Profiling technique.

In this direction, the PAN Profiling task of this year "*Profiling Fake News Spreaders on Twitter*", focus on "Given a Twitter feed, determine whether its author is keen to be a spreader of fake news" [18].

The working note is organized as follow: in the next section a brief description about the main aspects of the related works presented in the last three profiling evaluation forum. Next, we present our proposal. Specifically, we describe the data preprocessing as well as the deep-learning method used as classification model. Finally, we describe the experimental setting, the experiments conducted and the results achieved.

2 Related Works

Different aspects would be considered as part of the AP algorithms presented by the research community and highlighted in the overviews published by the organizers on

[1] https://pan.webis.de/

different PAN profiling editions. An important issue is the preprocessing step applied to the texts, considering that genre dependent characteristics can be eliminated or homogenized, such as in tweets, the use of hashtag, mentions, URL, etc. Generally, previous approaches have used content and style-based features and in recent years words and characters embeddings [23][22][19]. Regarding the machine learning algorithms, the most of studied works have employed traditional methods like Logistic Regression (LR) [31], Support Vector Machine (SVM) [26] and Random Forest (RF) [11], among others.

The textual information has been represented through lexical and character features, by using of n-gram models and bag of words (BoW) representation [26][31]. Several approaches have employed style features like the analysis of capital letters, function words, punctuation marks, etc. [6][8]. In recent PAN profiling task, words and documents embedding was introduced [23][22][19][15][12].

Shallow machine learning methods are the most employed ones, and among these, the most used are SVM, RF, LR and distance-based approaches. Generally, since PAN 2017, deep-learning techniques were introduced. Particularlly, the methods have focused on the use of Convolutional Neural Nets (CNN) and Recurrent Neural Nets (RNN) [25][5].

The goal of our approach is to fuse general stylistic features with deep learning-based representations. The stylistic feature vector will be used as an input representation to be part of the learning process in our deep-learning architecture.

3 Our Proposal

The motivation of our approach is twofold: firstly the ability of deep-learning methods to learn feature representations that are omitted in hand-craft features engine, also the dexterity of these methods for modeling abstraction levels beyond of human bounds. Secondly, author profiling task based on stylistic and linguistic features combined with shallow supervised learning methods have been well studied in previous research works. These features have proved to be adequate descriptors to determine some author characteristics such as: age, sexual gender, language variety, personality, etc. Keeping this in mind, our proposal learn two representations for each tweet in an account profile, which are combined with a whole account's profile representation based on stylistic features as can be shown in Figure. 1.

The first representation is based on Convolutional and Long Short Term Memory networks analyzing the tweets at word level *(encoder word-level, R1)*, the second representation is learned by using the same architecture without sharing the weights of the network, but at this time the tweets are analyzed at character level *(encoder character-level, R2)*. Finally, the last representation is based on stylistic features *(R3)*, which contain relationships between the use of grammatical structures like nouns, adjectives, lengths of words, function words, etc.

Our overall classification model is based on LSTM neural nets with attention mechanism (LSTM-Att). It receives as input the sequence of tweets in a Twitter's feed. Firstly, each tweet in the sequence is encoded by a dense vector which is composed by the representations obtained by the encoder at word-level and character-level respec-

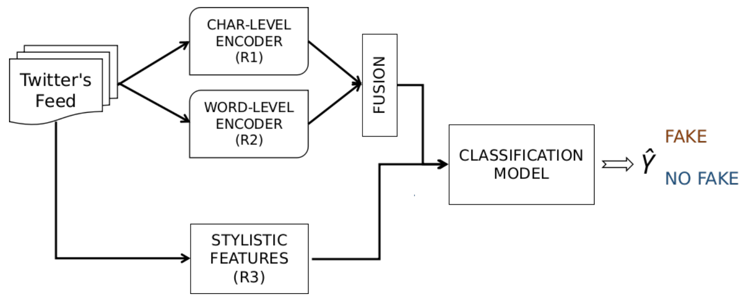


Figure 1: Overall Fake News Spreaders classification model.

tively. Later, the output vector learned by our LSTM-Att is combined with the stylistic features and fed into a dense layer to classify the account as fake spreader or not.

Details about our proposal are presented in the next subsections. Particularly, Section 3.1 introduces the preprocessing phase carried out on the tweets. Following, in Section 3.2 the stylistic features used in this work for modeling the Twitter's feeds are described. Later, Section 3.3 presents the architecture of the encoder models at word and character levels for obtaining the deep learning-based representation of the messages. Finally, Section 3.4 describes our overall classification model (LSTM-Att) and explains how the representations are fused to classify the Twitter's feed as fake spreader or not.

3.1 Preprocessing

In the preprocessing step, the tweets are cleaned. Firstly, the numbers and dates are recognized and replaced by a corresponding wildcard which encodes the meaning of these special tokens. Afterwards, tweets are tokenised and morphologically analyzed by means of FreeLing[3][17]. Also, we deal with the problem of character flooding. To this end, all repetitions of three or more contiguous characters are normalized to only two characters. Notice that, this normalization is applied when the messages are processing at word level, in case of character level we do not apply the normalization step.

3.2 Stylistic Feature

A key aspect of our proposal is the input representation based on statistical style features that capture information from distinct lexical and syntactical linguistic layers. For the representation of a text, it was defined 177 features for capturing relevant characteristics of the writing style of the author. Features were structured in six subsets considering different textual layers. These layers are boolean, character, sentence, paragraph, syntactic and the document.
Examples for each of the layers subset of features:

1. Boolean layer: Uses the same word to finish a sentence and to begin the next sentence.

2. Character layer: Average length of words.
3. Sentence layer: Average number of words. Average number of distinct prepositions.
4. Paragraph layer: Average number of sentence. Average number of words.
5. Syntactic layer: Proportion of nouns over adjective.
6. Text layer: Average length of sentence.

Our stylistic feature-based representation are completely independent of the textual genre, and in our proposal, the representation was build for each profile, see Figure. 2., for that reason several features can appear as duplicated value because it is considered that the tweet is a document which contains only one paragraph, and this paragraph is formed by one sentence.

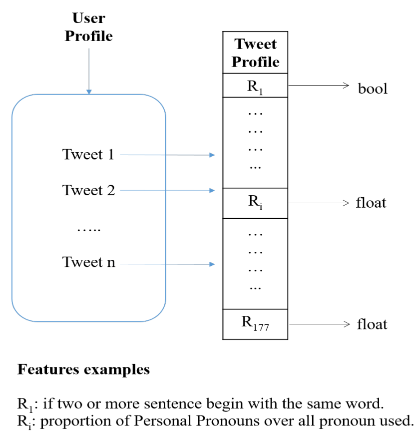


Figure 2: Tweets feed representation.

Notice that in the representation were involved different data structure values, since are computed boolean stylistic features as float data features.

3.3 CNN+LSTM Encoder Architecture

This section introduces the architecture of our CNN+LSTM encoder model shown on Figure. 3.which can be divided in two parallel sub-architectures which do not share weights but have the same structure. One of the encoder's sub-architecture processes the tweets at word-level whereas the another at character-level. This model aims at encoding syntactic and semantic properties of the tweets which could be correlated with the usage of fake and unverified content in the messages.

CNN Layers Our encoder model receives as input a tweet as a zero padded sequence, in order of having the same length l_s. This sequence is passed into an embedding layer. Notice that, at word-level this layer is set up with fixed weights from Google's *Word2Vec*

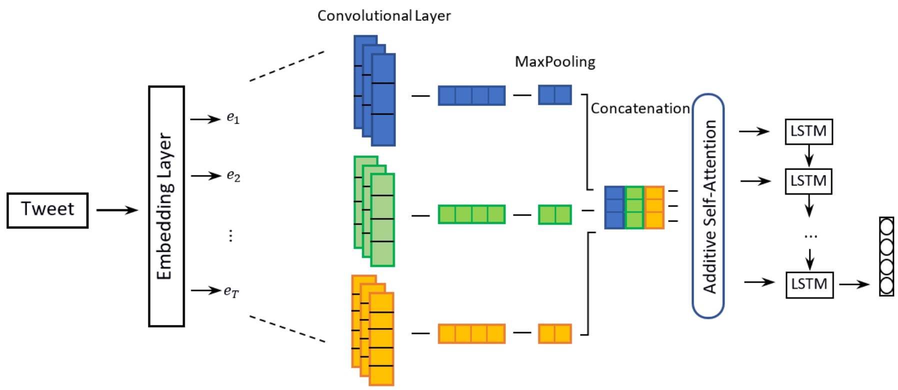


Figure 3: The architecture of the CNN+LSTM encoder

pre-trained embedding [16], whereas at character-level its weights are initialized randomly and set up as trainable.

As output of the embedding layer a matrix of real values $\hat{E}_{l_s \times d}$ is obtained, where each row is a dense vector that represents an element in the tweet sequence. Later, over that matrix $\hat{E}_{l_s \times d}$ is applied convolutional operations (*conv-op*) through a CNN layer composed by filters $F_k \in \Re^{h_k \times l_s}$ with different window sizes h_k to learn and capture short-term spacial relationships between the elements of the sequence. The *conv-op* for each set of filters with the same windows size, transforms its input into a matrix $C \in \Re^{l_s \times n_f}$, with n_f the number of filters used for current window size, on which is applied the non-linearity function ReLU.

For every filter k at position p of the sequence matrix the convolution operation is defined as:

$$C_{p,k} = \sum_{i=0}^{h_k} \sum_{j=0}^{ls} \hat{E}_{p+i,j} F_{k_{i,j}} \tag{1}$$

In order to preserve the output dimensions, the convolutional layer uses same padding before the *conv-op*. For every set of filters also was applied dropout [30] to reduce overfitting. Later, the outputs of each set of filters is passed through a maxpooling layer to keep relevant information, the windows size for maxpooling is the same as its corresponding filter. Therefore, these representations are again zero padded to be concatenated.

Self-Attention Layer Each element in the new sequence obtained from the CNN, as features map, provides relevant or not information about the message. In order to highlight the most important elements for encoding the message instead making the network pays attention to all elements alike, they are scored by its relative importance over the other elements on its context, with a *self-attention* layer [32].

Let x_t a dense vector that represents the t^{th} element in the sequence, self-attention layer learns how important it is and provides as output a new dense vector $\hat{x}_t$, that capture the relations of x_t with the other elements $x_{t'}$ in the sequence as $g_{t,t'}$:

$$g_{t,t'} = \tanh(W_x x_t + W_{xt'} x_{t'} + b_g) \tag{2}$$

Where $W_x, W_{xt'} \in \Re^{n_u \times d}$ are weight matrices which encode the representation of x_t and $x_{t'}$, to compute their compatibility as $a_{t,t'}$:

$$a_{t,t'} = \sigma(W_a g_{t,t'} + b_a) \tag{3}$$

With W_a a weight matrix to give them a non-linearity combination, b_a their respective bias term, and σ the sigmoid function. Then the final representation of x_t is the weighted sum of all elements in sequence (4).

$$\hat{x}_t = \sum_{i=0} a_{t,t'} x_{t'} \tag{4}$$

LSTM Layer The output of the attention layer is fed into a (LSTM) layer [10]. LSTM networks are a special kind of RNNs, which are specialized on analyzing sequential data. RNNs have a main cell unit (the recurrent unit) which explores the data sequence one element in each time step (left to right order). This network shares the information captured in the previous step, for computing the new hidden state at the current time step.

Let h_{t-1} the last computed hidden state, $x_t \in \Re^{1 \times d}$, the t^{th} element in the input sequence and f a non-linearity function. The current hidden state h_t is defined as:

$$h_t = f(W_x x_t + W_h h_{t-1} + b_h) \tag{5}$$

Where $W_x \in \Re^{n_u \times d}$ and $W_h \in \Re^{n_u \times n_u}$ are the weights matrices and $b_h \in \Re^{1 \times n_u}$ the bias term.

RNNs suffer from the problems of vanishing and exploding gradients [9], which hamper learning of long term dependencies among the elements in the input sequence. This limitation is what the more complex structure of LTSM tries to solve, with gates which learn to decide what information to preserve or forget from the previous time step.

Let $W_f, W_i, W_o \in \Re^{n_u \times d}$ and $U_f, U_i, U_o \in \Re^{n_u \times n_u}$ the weights matrices of forget, input and output gates respectively and $b_f, b_i, b_o \in \Re^{1 \times n_u}$ their bias terms respectively.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{8}$$

A potential output $\hat{c}_t$ is computed considering the previous hidden state h_{t-1} and the current element of the sequence x_t.

$$\hat{c}_t = \sigma(W_c x_t + U_c h_{t-1} + b_c) \tag{9}$$

Where $U_c \in \Re^{n_u \times d}$ and $U_c \in \Re^{n_u \times n_u}$ are weights matrices and $b_c \in \Re^{1 \times n_u}$ is a bias term. This potential output is combined with the vector computed by the input gate and added up to the output preserved from the previous time step as:

$$c_t = f_t c_{t-1} + i_t \tanh(\hat{c}_t) \tag{10}$$

Finally, the hidden state h_t at current position t is defined as:

$$h_t = o_t \tanh(c_t) \tag{11}$$

The LSTM layer used by our encoder was set up with 64 units, using dropout to prevent over-fitting.

From the set of hidden states of the LSTM layer's outputs is taken the last one, which has encoded all information of relationships among the elements in the input sequence and is fed into a fully connected layer with 32 units. Then the encoded representation obtained by both layers, at word and character levels, is concatenated and fed into another dense layer with 32 units.

As output of this encoder we get a fused feature representation learned from word-level and character-level over the message. Our encoder model was trained with a supervised learning task above the provided data for "*Profiling Fake News Spreaders on Twitter*" task[18], where we used as target for each tweet, whether it belongs to a fake new spreader account or not, which implied adding an extra fully connected layer with one neuron for making the prediction.

3.4 LSTM-Att Classifier model

Our final classifier model has two inputs for an account profile. Firstly, it receives the sequence of tweets, previously encoded by our CNN+LSTM encoder. Notice that, for each tweet in the account profile we obtain a dense vector that encodes its content. The set of vectors corresponding to each account's tweet are interpreted as sequential data, even when do not necessarily exist temporal relationships among them.

The second input is a fixed length vector of stylistic features extracted from the account profile. The architecture of our classification model (LSTM-Att) can be observed in Figure. 4.

The input sequence of vectors is fed into a Bidirectional-LSTM (BiLSTM) [29] layer, which consists in a 64 units LSTM cell, that returns two concatenated hidden states per each element in the sequence, one corresponding to a forward pass trough the sequence and another to the backward pass.

BiLSTM layer detects not just relations of an element with the previous ones, but also with the elements that appear after it. The hidden state in the time step t is $h_t = \overrightarrow{h}_t \overleftarrow{h}_t$. Also, on this layer we applied dropout to prevent over-fitting. After that, the BiLSTM output is passed into a self-attention layer, where they are weighted by its

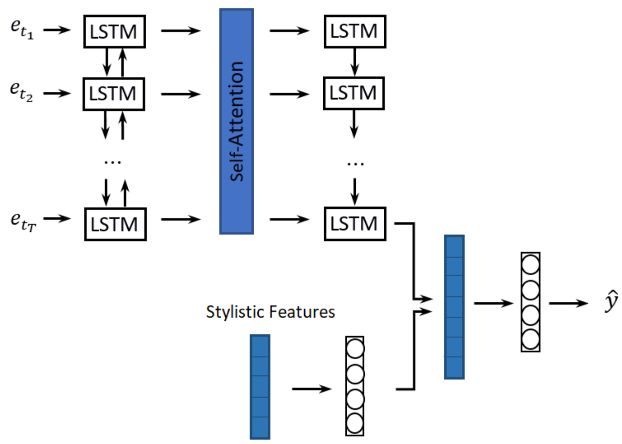


Figure 4: The architecture of the LSTM-Att classification model

relevance. Then, the output of the attention layer is passed into a simple LSTM layer, where we only keep the last hidden state as the profile's deep representation.

The profile's style-based representation is fed into a fully connected layer with 32 unit, to synthesize its information, using the sigmoid as activation function. Then, both representation are concatenated and fed into a new fully connected layer whit 32 units, with ReLU as the activation function. The output of the previous layer is passed into the last dense layer, which give us a real number between 0 and 1, representing the probability that the profile be a fake news spreader.

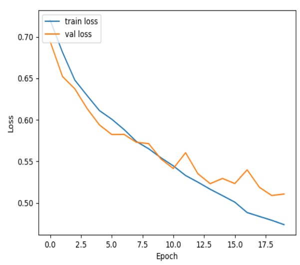
The final decision of our model is binary, if the output of our model is equal or greater than 0.5, the account profile is considered as a fake news spreader, in other case, it is considered as no fake news spreader.

4 Experiments and Results

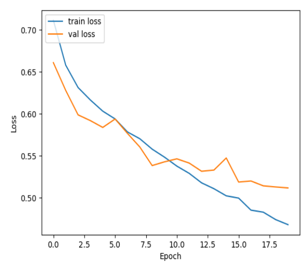
For training and developing our model we used Keras [4] framework with Tensorflow[1] backend on a GTX1050 with 4GB. The datasets used in this work are balanced. Nevertheless the data provided for the task "*Profiling Fake News Spreaders on Twitter*" is relatively small, since we have just 300 examples between fake news spreader and not fake news spreader profiles for English and Spanish, which may difficult the deep-learning models' training phase. The CNN+LSTM encoders and the LSTM-Att classifier models were trained using the *Adam* Optimizer [13], and the loss function was binary cross-entropy.

During the training phase carried out on the CNN+LSTM encoder the 90% of the dataset provided by the organizers was used for training whereas the 10% remaining was used as validation dataset. Firstly, the number of filters for each window size and the learning rate (lr) used by the Adam optimizer method were tuned. In the Figure. 5 we shown the hyper-parameter settings which produce better reductions of the loss curve. Particularly, fixing the learning rate $lr = 0.003$ and using 32 filters per windows size (3, 4, 5) achieved the best performance and produced the best trade-off respect to validation-training loss. We hypothesize that this setting has better performance w.r.t ones with more filters as result of the number of parameters which need to be learned by more complex model from a relative small dataset. It is worth to remark that we applied an early stopping strategy to prevent the over-fitting, we consider a patience value equal to 10 epochs.

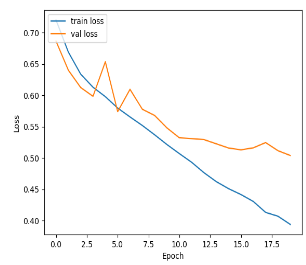
Spanish language



(a) lr: 2e-3, conv. filters per window size: 32

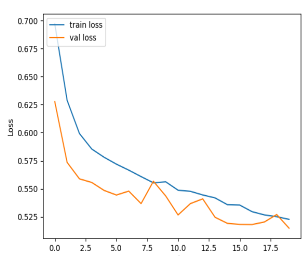


(b) lr: 3e-3, conv. filters per window size: 32

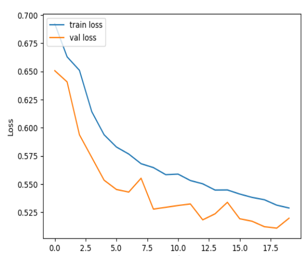


(c) lr: 3e-3, conv. filters per window size: 100

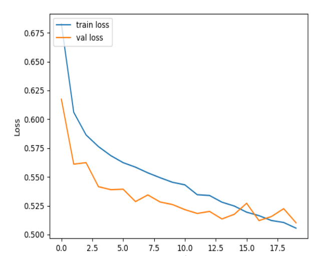
English language



(d) lr: 2e-3, conv. filters per window size: 32



(e) lr: 3e-3 conv. filters per window size: 32



(f) lr: 3e-3, conv. filters per window size: 100

Figure 5: Loss curves of the training configurations for the encoder model.

Considering that, the profiling task proposed by PAN' 2019: *Bots and Gender Profiling 2019* [19] is closed-related with the current profiling task, we hypothesize that, performing transfer-learning from the weights learned on the dataset and the task introduced in 2019, would improve the effectiveness of our model. For that, we evaluated two distinct strategies. In first one *(Fine-tuning)*, we train our model LSTM-Att on the PAN' 2019 dataset and later retrain the model on the dataset provided for the fake news spreader detection task. The second one *(Fixed-weights)*, is based on the idea of training the model on the PAN' 2019 dataset and fixing all weights except those from the last two dense layers. Later, is retrained using the dataset provided for the fake news spreader detection task to learn the new weights of these two layers. As can be observed in Table. 1, applying transfer-learning improved the performance of our LSTM-Att model, particularly for the English language. Regarding the both strategies, we can observe that the *Fine-tuning* outperform the *Fixed-weights* in both languages.

Table 1: Impact of performing transfer learning on our LSTM-Att model.

Transf strategy	**EN**	**ES**
No transf	0.720	0.736
Fine tuning	**0.791**	**0.761**
Fixed weights	0.787	0.728

During training phase carried out on the LSTM-Att model was used a cross-validation method to obtain a more realistic an unbiased performance evaluation. On each cross-validation step, the dataset was split in 10 % for validation and 90% for training. Also as we mentioned above, the tweets which compose the accounts' sequence have no necessary temporal relationships among them. Keeping this in mind, on every epoch the order of the elements in the sequence was random shuffled to prevent learning some kind of wrong patterns.
For training our model we tuned the learning parameter used by the Adam method, choosing the one with higher accuracy average among the validation subsets used in the cross-validation method. As we showed in Table. 2 the setting which obtained the better performance was with $lr = 0.001$.

Table 2: LSTM-Att classification model with stylistic features.

Learning Rate	**English**		**Spanish**	
	val	train	val	train
3e-3	0.956	0.962	0.890	0.960
2e-3	0.946	0.972	0.893	0.963
1e-3	**0.960**	0.982	**0.880**	0.950

Also we tried to exclude the profile's style-based features vector in order to avoid introducing hand-crafted features, but as we can see in Table. 3 this representation helps our model to improve its accuracy.

Regarding official evaluation and results, our system's submission for facing the task "*Profiling Fake News Spreaders on Twitter*" used the same architecture for the CNN+LSTM encoders and the LSTM-Att classifier. Also we considered the same values for the hyper-parameters for both languages, English and Spanish, on the training phase. Moreover, taking into account the positive impact of the style-based features in

Table 3: Impact of the stylistic-based representation in our LSTM-Att model.

LSTM-Attl	**English**		**Spanish**	
	val	train	val	train
Style-based feat.	**0.960**	**0.982**	0.880	**0.950**
No style-based feat.	0.950	0.9633	**0.886**	0.934

our model we introduced this representation in the final submission. After the evaluation on the test dataset proposed by the organizers by means of the accuracy measure, our model obtained an acc=0.705 and acc=0.720 for English and Spanish languages respectively. Regarding the official ranking, a first glance at Table. 4 allows to observe that our submission was ranked as 26th from a total of 66 teams and several baselines. Notice that, our system outperforms the LSTM baseline which is based on deep-leaning method.

Table 4: Official results for the task: "*Profiling Fake News Spreaders on Twitter*" on the test dataset for both languages.

POS	**Team**	**EN**	**ES**	**AVG**
1	bolonyai20	0.7500	0.8050	0.7775
1	pizarro20	0.7350	0.8200	0.7775
-	SYMANTO (LDSE) [27]	0.7450	0.7900	0.7675
3	koloski20	0.7150	0.7950	0.7550
	⋮			
-	SVM + c nGrams	0.6800	0.7900	0.7350
	⋮			
26	**LSTM-Att**	**0.7050**	**0.7200**	**0.7125**
	⋮			
-	NN + w nGrams	0.6900	0.7000	0.6950
-	EIN [7]	0.6400	0.6400	0.6400
-	LSTM	0.5600	0.6000	0.5800
-	RANDOM	0.5100	0.5000	0.5050
	⋮			
65	margoes20	0.570	-	
66	wu20	0.560	-	

5 Conclusions and Future Work

In this paper we described our system for participating in the PAN 2020 Author Profiling task: "*Profiling Fake News Spreaders on Twitter*". Our proposal is based on LSTM neural nets with attention mechanism (LSTM-Att). It receives as input the sequence of tweets in a Twitter's feed. Firstly, each tweet in the sequence is encoded by a dense vector which is composed by the fusion of the representations obtained by the encoder at word-level and character-level respectively. Later, the output vector learned by our LSTM-Att is combined with the stylistic features and fed into dense layers to classify the account as fake spreader or not. The results shown that considering both the stylistic representation and the deep representations learned at word-level and character-level by our encoder CNN+LSTM obtains the best effectiveness based on the accuracy measure. Due to encouraging results of our approach, we think that including other linguistic features, mainly those related with affective information and personality traits could be a way to increase the effectiveness. Also, we plan to consider other deep representation methods based on unsupervised deep language modeling. We would like to explore these ideas in the future work.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `http://tensorflow.org/`, software available from tensorflow.org
2. Aragon, M., Carmona, M.A., Montes, M., Escalante, H.J., Villaseñor Pineda, L., Moctezuma, D.: Overview of mex-a3t at iberlef 2019: Authorship and aggressiveness analysis in mexican spanish tweets (08 2019)
3. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: Freeling 1.3: Syntactic and semantic services in an open-source NLP library. In: Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., Tapias, D. (eds.) Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006. pp. 2281–2286. European Language Resources Association (ELRA) (2006), `http://www.lrec-conf.org/proceedings/lrec2006/summaries/198.html`
4. Chollet, F., et al.: Keras. `https://keras.io` (2015)
5. Dias, R.F.S., Paraboni, I.: Combined CNN+RNN bot and gender profiling. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2380/paper_61.pdf`
6. Fontcuberta, J.R.P., la Peña Sarracén, G.L.D.: Bots and gender profiling using a deep learning approach. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2380/paper_221.pdf`
7. Ghanem, B., Rosso, P., Rangel, F.: An Emotional Analysis of False Information in Social Media and News Articles. ACM Transactions on Internet Technology (TOIT) 20(2), 1–18 (2020)
8. Giachanou, A., Ghanem, B.: Bot and gender detection using textual and stylistic information. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of

CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2380/paper_169.pdf
9. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997), https://doi.org/10.1162/neco.1997.9.8.1735
11. Johansson, F.: Supervised classification of twitter accounts based on textual content of tweets. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2380/paper_154.pdf
12. Joo, Y., Hwang, I.: Author profiling on social media: An ensemble learning approach using various features. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2380/paper_174.pdf
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
14. Litvinova, T., Pardo, F.M.R., Rosso, P., Seredin, P., Litvinova, O.: Overview of the rusprofiling PAN at FIRE track on cross-genre gender identification in russian. In: Majumder, P., Mitra, M., Mehta, P., Sankhavara, J. (eds.) Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017. CEUR Workshop Proceedings, vol. 2036, pp. 1–7. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2036/T1-1.pdf
15. López-Santillán, J., González-Gurrola, L.C., Montes-y-Gómez, M., Alonso, G.R., Prieto-Ordaz, O.: An evolutionary approach to build user representations for profiling of bots and humans in twitter. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2380/paper_176.pdf
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 3111–3119 (2013), http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
17. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012. pp. 2473–2479. European Language Resources Association (ELRA) (2012), http://www.lrec-conf.org/proceedings/lrec2012/summaries/430.html
18. Pardo, F.M.R., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings (Sep 2020), CEUR-WS.org

19. Pardo, F.M.R., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: Proceedings of the CEUR Workshop, Lugano, Switzerland. pp. 1–36 (2019)
20. Pardo, F.M.R., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., Sánchez-Junquera, J.: On the author profiling and deception detection in arabic shared task at FIRE. In: Majumder, P., Mitra, M., Gangopadhyay, S., Mehta, P. (eds.) FIRE '19: Forum for Information Retrieval Evaluation, Kolkata, India, December, 2019. pp. 7–9. ACM (2019), `https://doi.org/10.1145/3368567.3368586`
21. Pardo, F.M.R., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., Sánchez-Junquera, J.: Overview of the track on author profiling and deception detection in arabic. In: Mehta, P., Rosso, P., Majumder, P., Mitra, M. (eds.) Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019. CEUR Workshop Proceedings, vol. 2517, pp. 70–83. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2517/T2-1.pdf`
22. Pardo, F.M.R., Rosso, P., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th author profiling task at PAN 2018: Multimodal gender identification in twitter. In: Cappellato, L., Ferro, N., Nie, J., Soulier, L. (eds.) Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings, vol. 2125. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2125/invited_paper_15.pdf`
23. Pardo, F.M.R., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at PAN 2017: Gender and language variety identification in twitter. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. CEUR Workshop Proceedings, vol. 1866. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-1866/invited_paper_11.pdf`
24. Pardo, F.M.R., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016. CEUR Workshop Proceedings, vol. 1609, pp. 750–784. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1609/16090750.pdf`
25. Petrík, J., Chudá, D.: Bots and gender profiling with convolutional hierarchical recurrent neural network. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2380/paper_214.pdf`
26. Pizarro, J.: Using n-grams to detect bots on twitter. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2380/paper_183.pdf`
27. Rangel, F., Franco-Salvador, M., Rosso, P.: A Low Dimensionality Representation for Language Variety Identification. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 156–169. Springer (2016)
28. Rosso, P., Pardo, F.M.R., Hernandez-Farias, I., Cagnina, L.C., Zaghouani, W., Charfi, A.: A survey on author profiling, deception, and irony detection for the arabic language. Lang. Linguistics Compass 12(4) (2018), `https://doi.org/10.1111/lnc3.12275`
29. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45(11), 2673–2681 (1997), `https://doi.org/10.1109/78.650093`

30. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958 (2014), http://dl.acm.org/citation.cfm?id=2670313
31. Valencia-Valencia, A.I., Gómez-Adorno, H., Rhodes, C.S., Pineda, G.F.: Bots and gender identification based on stylometry of tweet minimal structure and n-grams model. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2380/paper_216.pdf
32. Zheng, G., Mukherjee, S., Dong, X.L., Li, F.: Opentag: Open attribute value extraction from product profiles. In: Guo, Y., Farooq, F. (eds.) Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018. pp. 1049–1058. ACM (2018), https://doi.org/10.1145/3219819.3219839