Improving LSTMs' under-performance in Authorship Attribution for short texts

Christian Oliva
Luis F. Lago-Fernández
christian.oliva,luis.lago@uam.es
Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain

Santiago Palmero Muñoz
David Arroyo
santiago.palmero,david.arroyo@csic.es
Institute for Physical and Information Technologies, Spanish National Research Council, 28006 Madrid, Spain

ABSTRACT

We present a novel approach for conducting authorship attribution over tweets using Long-Short Term Memory networks (LSTMs). Vanilla LSTMs use the last hidden state for prediction. Our strategy introduces a mechanism based on Max Pooling to process all the hidden states simultaneously, which helps the model to better detect authors' stylometry. We obtain a 4% accuracy improvement with respect to vanilla LSTMs.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Security and privacy** → *Social network security and privacy*.

KEYWORDS

Authorship Attribution, LSTM, Stylometry

ACM Reference Format:

Christian Oliva, Luis F. Lago-Fernández, Santiago Palmero Muñoz, and David Arroyo. 2018. Improving LSTMs' under-performance in Authorship Attribution for short texts. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Natural Language Processing (NLP) is still a challenging problem in many research areas, such as Authorship Attribution (AA). AA is a multi-class classification problem whose goal is the identification of the author of a given text, given a set of potential authors. AA is a very useful tool to assess information credibility in social media, which eventually could help in tasks as bot, spam or fake news detection [5]. Nowadays, Twitter is one of the most relevant social media. AA in Twitter is difficult to be carried out due to the short length of the tweets' text. Traditional Machine Learning approaches, and also Deep Learning with Convolutional Neural Networks, have proven to be effective for this problem. However, Long-Short Term Memory networks (LSTMs) are not as good as expected [7]. In this work, we propose a novel approach for AA using LSTMs. We obtain a 4% accuracy improvement with respect to the state of the art.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 MODEL DESIGN

Recurrent neural architectures are profusely used in the characterization and classification of natural language. Among the different architectures, LSTMs is one the most popular [1]. Considering stylometry, LSTMs are able to characterize temporal information in terms of their internal timeline. Nevertheless, the inner characteristics of the model are adequate to characterize global temporal dependencies but not the temporal nuances associated to relevant stylometric features [4, 6, 8]. In order to overcome this limitation, we have to take into account how text style is modeled by means of the prediction given by the LSTM output.

In our approach, prediction is carried out considering all the hidden states of an LSTM at every step. Instead of processing the final output with just the last hidden state $h_{t=T}$, we recollect every state from $h_{t=0}$ to $h_{t=T}$ ($h_{t[0:T]}$). Then, we add a Dense layer with $ReLu$ activation function to generate new attribute vectors ($d_{t[0:T]}$) breaking the limits of $tanh$, ranged on [-1, +1]. Note that all $h_{t[0:T]}$ share the same Dense layer D. Following D, we introduce a Max Pooling (MP) operation for 1−dimensional temporal data to downsample the time dimension by taking the maximum value over the complete temporal window of size T (d_{mp}). Finally, we apply the $softmax$ function to obtain the final prediction. We show an explanatory diagram in figure 1 with an example with $T = 5$.

3 EXPERIMENTS

In this work, AA is performed using a very well known dataset for AA in Twitter [5]. The dataset has a total of 6212 Twitter users (i.e., potential text authors) and 6.2×10^6 tweets. From this pool of users, 10 subsamples of 50 users, with 1000 tweets each, have been randomly created. We compare the performance of the vanilla LSTM architecture (Embedding-Recurrent-Softmax) with our proposed approach (*LSTM+D+MP*). In order to dissociate the effects of the Dense and the Max Pooling layers, we also consider an LSTM model with a Dense layer to compare them in the same scenario (*LSTM+D*). The LSTM models have been implemented using Keras[1].

We look for the best hyperparameters in the first subsample of the 10 available subsamples. Then, we get an average test accuracy over the remaining ones. This experiment is repeated for a different number of users in the range $[5, 50]$. All the networks are trained to minimize the cross-entropy loss using the Adam optimizer [2]. They have 200 units in the embedding layer, 400 units in the LSTM layer, and 1000 units in D[2]. The rest of the hyperparameters (see table 1) are searched using the Hyperband algorithm [3].

[1] https://keras.io/layers/recurrent/#lstm [visited on 11 March 2022]
[2] This configuration of D applies to *LSTM+D* and *LSTM+D+MP* models.

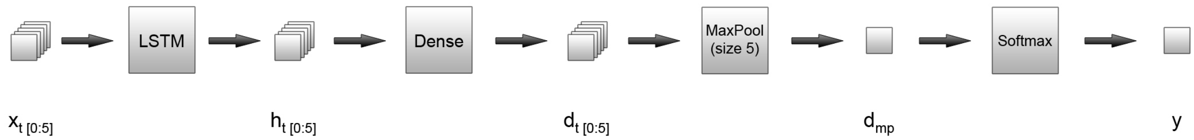

Figure 1: LSTM with MP diagram. Example of an input with 5 time steps. As a consequence, the MP must be of size 5.

Table 1: Table of hyperparameters search. *Dense Dropout hyperparameter is used in *LSTM+D* and *LSTM+D+MP*.**

Hyperparameter	Range of values	Step
Learning rate	$[5{\cdot}10^{-4}, 5{\cdot}10^{-2}]$	log sampling
Embedding output Dropout	[0.0, 0.8]	0.01
LSTM Dropout	[0.0, 0.8]	0.01
LSTM output Dropout	[0.0, 0.8]	0.01
*Dense Dropout**	[0.0, 0.8]	0.01

4 ANALYSIS AND RESULTS

We compare the performance of the three recurrent models following the methodology described in the previous section (sec. 3). Figure 2 shows the test accuracy versus the number of Twitter users. Note that the more users, the more complexity.

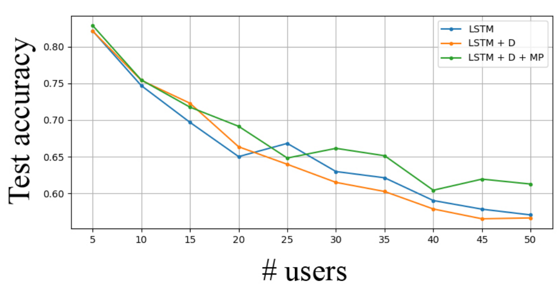

Figure 2: Test accuracy versus the number of users in AA

In the figure, we observe two different behaviors. First, when the number of users is low, the three models perform similarly, with an accuracy close to 82% On the other hand, when complexity increases as a result of increasing the number of Twitter users, the curve for the *LSTM+D+MP* model separates from the others, achieving more than 60% accuracy. It is worth noting that the improvement is not due to the extra parameters of the Dense layer, since the inclusion of this layer (*LSTM+D* model) is not enough to beat the vanilla LSTM. The model needs the MP operation to process the complete set of states and not just the last one. We show in table 2 the test accuracy of the three presented models in the case of 50 users. In the table, we can see that the MP strategy improves the accuracy in more than 4%.

Table 2: Test accuracy in AA with 50 users

Model	Text accuracy
LSTM	0.5706±0.0343
LSTM+D	0.5664±0.0377
LSTM+D+MP	**0.6127±0.0353**

5 CONCLUSION

In this work, we have presented a novel strategy to address the AA task in Twitter with recurrent neural networks. We have shown that the model needs to process all its internal states together to keep the relevant information through the timeline when the problem is complex enough. The proposed model introduces a Max Pooling layer after a Dense layer for this task. We have shown that our model increases the performance over the vanilla LSTM with more than 4% accuracy.

This work needs further research, the model must be tested in other problems, such as text classification or sentiment analysis. Also, it could be interesting to analyze how the relevance of some stylometric features gets dissipated through the timeline.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 872855 (TRESCA project), as well as from Comunidad de Madrid (Spain) under the project CYNAMON (no. P2018/TCS-4566), cofunded with FSE and FEDER EU funds, Spanish Government under project MINECO/FEDER PID2020-114867RB-I0, and Grant PLEC2021-007681 (project XAI-DisInfodemics) funded by MCIN/AEI/ 10.13039/501100011033 and by European Union NextGeneration EU/PRTR.

REFERENCES

[1] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9 (12 1997), 1735–80.
[2] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).
[3] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research* 18, 185 (2018), 1–52.
[4] Piotr Mirowski and Andreas Vlachos. 2015. Dependency recurrent neural language models for sentence completion. *arXiv preprint arXiv:1507.01193* (2015).
[5] Anderson Rocha, Walter J. Scheirer, Christopher W. Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne R. B. Carvalho, and Efstathios Stamatatos. 2017. Authorship Attribution for Social Media Forensics. *IEEE Transactions on Information Forensics and Security* 12, 1 (2017), 5–33.
[6] Yunita Sari. 2018. *Neural and Non-neural Approaches to Authorship Attribution.* Ph. D. Dissertation. University of Sheffield.
[7] Prasha Shrestha, Sebastian Sierra, Fabio A González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers.* 669–674.
[8] Geoffrey Zweig, John C Platt, Christopher Meek, Christopher JC Burges, Ainur Yessenalina, and Qiang Liu. 2012. Computational approaches to sentence completion. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 601–610.