Improving Irony and Stereotype Spreaders Detection using Data Augmentation and Convolutional Neural Network

Notebook for PAN at CLEF 2022

Stefano Mangione[1], Marco Siino[1] and Giovanni Garbo[1]

[1]Università degli Studi di Palermo, Dipartimento di Ingegneria, Palermo, 90128, Italy

Abstract

In this paper we describe a deep learning model based on a Data Augmentation (DA) layer followed by a Convolutional Neural Network (CNN). The proposed model was developed by our team for the Profiling Irony and Stereotype Spreaders (ISSs) task proposed by the PAN 2022 organizers. As a first step, to classify an author as ISS or not (nISS), we developed a DA layer that expands each sample in the dataset provided. Using this augmented dataset we trained the CNN. Then, to submit our predictions, we apply our DA layer on the samples within the unlabeled test set too. Finally we fed our trained CNN with the augmented test set to generate our final predictions. To develop and test our model we used a 5-fold cross validation on the labelled training set. The proposed model reaches a maximum accuracy of 0.92 and an average accuracy of 0.89 over the five folds. Meanwhile, on the provided test set the proposed model reaches an accuracy of 0.9278.

Keywords

irony, stereotypes, author profiling, text classification, Twitter, data augmentation, convolutional neural network

1. Introduction

The author profiling task proposed at PAN@CLEF2022 [1] was about Profiling Irony and Stereotype Spreaders (ISSs) on Twitter [2]. The task was to investigate whether or not an author of a given Twitter thread is likely to spread tweets containing irony and stereotypes. The organizers provided a labeled English dataset, consisting of 420 samples. Each sample represents a single author. For each author a set of 200 tweets is provided. The unlabeled test set provided consists of 180 samples. The model we used to compete for the task consists of a Data Augmentation (DA) layer followed by a shallow Convolutional Neural Network (CNN). Broadly speaking, our model preprocess each sample in the dataset to expand it. The DA layer is based on a back-translation mechanism discussed in this work. This DA layer is applied both to the labeled training set and to the unlabeled test set. Then, both for the training and the

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

stefano.mangione.tlc@unipa.it (S. Mangione); marco.siino@unipa.it (M. Siino); giovanni.garbo@unipa.it (G. Garbo)
https://github.com/marco-siino (M. Siino)
0000-0003-0926-4902 (S. Mangione); 0000-0002-4453-5352 (M. Siino); 0000-0001-8837-0214 (G. Garbo)

CEUR Workshop Proceedings (CEUR-WS.org)

prediction phases, we use the augmented dataset. The model used is a shallow CNN already used in similar text classification tasks. This model is able to reach state-of-the-art results on several binary text classification tasks.

Our paper is organized as follows. In Section 2 related works about deep learning methods for text classification are presented. In Section 3 we describe the methodology to augment the dataset, training the model and make predictions. In Section 4 we report the results of our tests over a 5-fold cross validation and on the test set. In Section 5 we propose future works and conclude the paper.

2. Related work

Recent approaches about the detection of stereotypes are proposed in [3, 4] while some interesting methods and discussions about irony detection are proposed in [5, 6].

In this work, to address the problem of detecting ISSs on Twitter, we started from the analysis and study of state-of-the-art techniques for text classification [7, 8, 9]. Then we look at the results of the last year author profiling task hosted at PAN@CLEF 2021. The best performing model consisted of a CNN fully described in [10]. Furthermore, given the performance reached in a similar text classification task [11] and, as discussed in [12, 13], the fact that deep AI models are finally able to outperform classic techniques used in natural language processing, we decided to use a deep learning-based approach based on a CNN for our final submission.

The first attempt to implement CNN for text classifications purposes was conducted in [14] in which, for the first time, a CNN was used to address a text classification task. The CNN obtained promising performances compared to state-of-the-art models. Within a CNN, a common representation of words is based on word embeddings [15, 16].

Furthermore, we wanted to investigate the impact of DA to improve the performance of a deep learning-based model for a text classification task. An attempt to perform DA is presented in [17] where dataset samples are augmented using four basic operation: synonym replacement, random insertion, random swap, and random deletion. Over five classification tasks, authors prove that performance are increased using DA.

However, our team developed a fully automated DA tool which could employ other types of operations over the dataset provided. Such an approach could be the one proposed in [18] based on back-translation.

Back-translation has been proven effective by extensive experiments conducted in [19]. A back-translation technique was proved effective on question-and-answer tasks, as discussed in [20]. In this work, the author improved performance by generating new data through reverse translation that translates English to French and back to English.

Authors in [21] propose a deep learning-based method that fuses a back-translation method, and a paraphrasing technique for data augmentation. A final stage using deep classifiers (Long short-term memory network and CNN) is evaluated to seek enhanced classification results. The evaluation is made over five publicly available datasets. Compared with state-of-the-art results, performance of the proposed method demonstrate the effectiveness and soundness of it.

Finally it is worth reporting a relevant increase in the use of Explainable AI (XAI) methods in place of black box-based approaches. A few of these methods are based on graphs and used in real-world applications such as text classification [22], traffic prediction [23], computer vision [24] and social networking [25].

3. Methodology

3.1. The dataset

The dataset consists of a set of 600 Twitter authors. For each author a set of 200 tweets is provided. The labeled training set provided by the organizers contains 420 authors. The test set is composed of the 180 remaining ones. Authors in the training set are labeled as "I" (ISS) or "NI" (nISS). Our final submission consists of a zip file containing predictions for each non-labeled author within the test set. A single XML file corresponds to a single author and contains 200 tweets from the author.

3.2. Dataset preprocessing

Before our DA layer we preprocess our dataset to remove useless information. More specifically, we remove the opening tag *CDATA* from every XML file. Then we removed the starting tag *<documents>* opening each sample. Finally we removed the opening and closing tag *<author lang="en">* and we lowercased all the text.

3.3. Dataset augmentation

In Figure 1 is shown the overview of our proposed framework. The very first stage takes a sample from the dataset. The sample is preprocessed as described in the previous section. Such a preprocessed sample is then augmented. To perform the augmentation our framework back-translates each sample. We implement this operation, within our framework, performing online request to the *GoogleTranslator* API from the *deep_translator* library. Full documentation of this module is available online[1]. Thanks to this module we translate each sample in dataset from English to Italian. Then we translate back from Italian to English and finally, we merge the original sample with the back-translated one.

The rationale behind such an augmentation strategy is easily explainable with the following running example. In the example, making use of the Google Translate Tool online, a tweet contained in one of the samples from the provided dataset is translated and then back-translated.

- ORIGINAL (ENG) -> *"Yeah; on paper, kinda shitty business model expecting banks et al to buy something they have never needed, and never will need. But we know the real business model is extracting fiat cash from morons; it seems to be playing out swimingly."*
- TRANSLATED (ENG to IT) -> *"Sì; sulla carta, un modello di business di merda che prevede che le banche e altri acquistino qualcosa di cui non hanno mai avuto bisogno e di cui non avranno mai bisogno. Ma sappiamo che il vero modello di business è estrarre denaro fiat dagli idioti; sembra che stia nuotando."*

[1] https://deep-translator.readthedocs.io/en/latest/

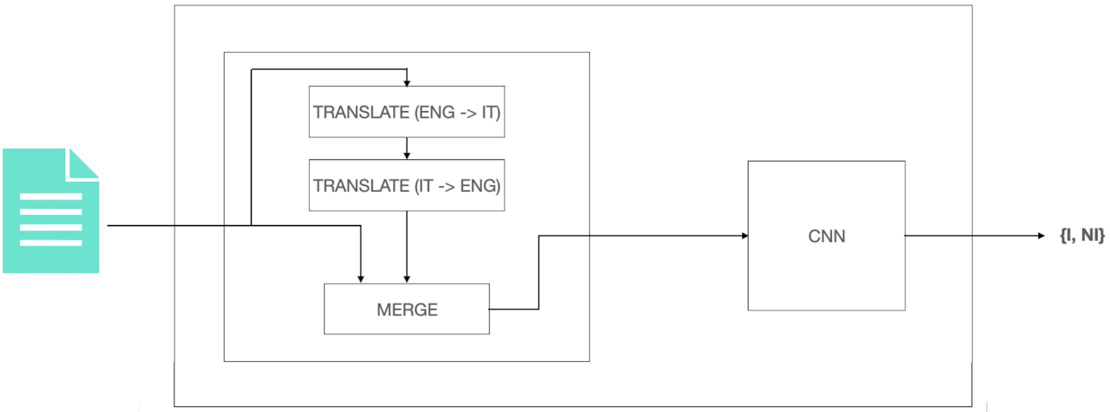

Figure 1: Overall architecture of the proposed framework. Both for training and testing phase the CNN proposed in [10] operates on an augmented dataset built as shown in Figure. Each sample is augmented back-translating the contained text and merging it with the original source.

- TRANSLATED (IT to ENG) -> *"Yup; on paper, a shitty business model that requires banks and others to buy something they never needed and never will. But we know that the real business model is to extract fiat money from idiots; it looks like it is swimming."*

As can be seen from our running example from the dataset provided, the process of back-translating a sample replace, delete and insert words, in fact augmenting the information available for the subsequent CNN-based model. It is worth reporting that while maintaining the characteristics of ironic text (in the first part of an augmented sample), an increased number of words and sentences could provide more information available for the training of the CNN.

In our experiments we also used back-translation to generate a separate sample for each original sample in the training set. However there was no improvement in term of accuracy over the five fold validation.

3.4. Model training

The architecture of the CNN-based model used in our work is very similar to the one discussed in [10]. It is a shallow CNN as depicted in Figure 2.

The model is compiled with a binary cross entropy loss function; this function calculates loss with respect to two classes (i.e., 0 and 1) as defined in (1). These classes are obtained after thresholding the output of the last single dense unit of the CNN. Positive values as output are considered as ISS (i.e., 1) and negative ones as nISS (i.e., 0).

$$\text{Loss}_{BCE} = -\frac{1}{N}\sum_{n=1}^{N}[y_n \times \log\left(h_\theta(x_n)\right) + (1-y_n) \times \log\left(1-h_\theta(x_n)\right)] \tag{1}$$

Optimization is performed with an Adamic optimizer [26] after giving each batch of data as input. We performed a binary search for finding the optimal batch size. The model achieved the best overall accuracy with a batch size of 1. Our model, developed in TensorFlow, is publicly available as a Jupyter Notebook on GitHub.

	Fold Nr.						
	1	2	3	4	5	AVG	σ
Accuracy	0.8947	0.8684	0.9211	0.8684	0.9079	0.8921	0.0210
Loss	0.3322	0.2710	0.3024	0.4102	0.2129	0.3057	0.0655

Table 1
Results achieved by the model on a 5-fold cross validation on the training set provided. In this case the DA layer is used before using each augmented sample as input for the CNN.

4. Results

In Table 1 are reported the results obtained adopting a 5-fold cross validation. The table reports accuracy and loss values achieved on the validation set used at each fold, together with the arithmetic mean and standard deviation. For each fold we trained the model for 5 epochs. That is motivated by the start of overfitting by the sixth epoch. In the same table, the higher accuracies and the related losses are shown over the training epochs, with respect to the validation set used at the fold indicated. As can be noted, some splits achieved a better performance and this could be due to a higher level of similarity between the considered train and validation sets. The model trained on the best fold was used to make predictions on the test set provided for the competition. Predictions were uploaded on TIRA [27]. As reported in the final ranking[2], the proposed model (namely, *stm*) reaches an accuracy of 0.9278.

5. Conclusion and future works

In this paper we described our submitted framework for the participation of our team at the Profiling ISSs on Twitter task at PAN 2022. It consists of a DA layer followed by a CNN based on a single convolutional layer. To get a more accurate evaluation of the model performance, we run several 5-fold cross validation for each different hyperparameter configuration. After finding the model achieving the highest accuracy during our cross validation tests, we trained such a model on the best train fold to submit our predictions on the unlabeled test set.

In future works, we expect to evaluate performances on several languages and other back-translation strategies. Even conducting an error analysis on misclassified authors could maybe lead to improved performances on the classification task proposed. Another direction to improve accuracy in profiling ISSs could be to add more complexity to the model, maybe using some additional layers. Given the dimension of the dataset provided some other techniques of data augmentation could be also applied. Finally, some investigation on the content of each tweet could guide us in applying some techniques to remove noise (i.e., not relevant features) from the input samples before the training and testing phases of our model.

Acknowledgments

We would like to thank the anonymous reviewers for their comments and suggestions that have

[2] https://pan.webis.de/clef22/pan22-web/author-profiling.html

helped to improve the presentation of the paper.

CRediT Authorship Contribution Statement

Stefano Mangione: Writing - review & editing, Methodology. **Marco Siino:** Conceptualization, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - Original draft, Writing - review & editing. **Giovanni Garbo:** Writing - review & editing, Methodology.

References

[1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection, in: A. Barron-Cedeno, G. D. S. Martino, M. D. Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), volume 13390 of *Lecture Notes in Computer Science*, Springer, 2022, p. 1.

[2] O.-B. Reynier, C. Berta, R. Francisco, R. Paolo, F. Elisabetta, Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022, p. 1.

[3] J. Sánchez-Junquera, B. Chulvi, P. Rosso, S. P. Ponzetto, How do you speak about immigrants? taxonomy and stereoimmigrants dataset for identifying stereotypes about immigrants, Applied Sciences 11 (2021) 3610.

[4] J. Sánchez-Junquera, P. Rosso, M. Montes, B. Chulvi, et al., Masking and bert-based models for stereotype identication, Procesamiento del Lenguaje Natural 67 (2021) 83–94.

[5] S. Zhang, X. Zhang, J. Chan, P. Rosso, Irony detection via sentiment-based transfer learning, Information Processing & Management 56 (2019) 1633–1644.

[6] E. Sulis, D. I. H. Farías, P. Rosso, V. Patti, G. Ruffo, Figurative messages and affect in twitter: Differences between# irony,# sarcasm and# not, Knowledge-Based Systems 108 (2016) 132–143.

[7] M. Thangaraj, M. Sivakami, Text classification techniques: A literature review., Interdisciplinary Journal of Information, Knowledge & Management 13 (2018).

[8] B. Altınel, M. C. Ganiz, Semantic text classification: A survey of past and recent advances, Information Processing & Management 54 (2018) 1129–1153.

[9] R. Oshikawa, J. Qian, W. Y. Wang, A survey on natural language processing for fake news detection, arXiv preprint arXiv:1811.00770 (2018).

[10] M. Siino, E. Di Nuovo, I. Tinnirello, M. La Cascia, Detection of hate speech spreaders using convolutional neural networks, in: PAN 2021 Profiling Hate Speech Spreaders on Twitter@ CLEF, volume 2936, CEUR, 2021, pp. 2126–2136.

[11] M. Siino, M. La Cascia, I. Tinnirello, McRock at SemEval-2022 Task 4: Patronizing and Condescending Language Detection using Multi-Channel CNN and DistilBERT, in: Pro-

ceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Association for Computational Linguistics, 2022, p. 1.

[12] H. Wu, Y. Liu, J. Wang, Review of text classification methods on deep learning, CMC-Computers, Materials & Continua 63 (2020) 1309–1321.

[13] S. Hashida, K. Tamura, T. Sakai, Classifying tweets using convolutional neural networks with multi-channel distributed representation, IAENG International Journal of Computer Science 46 (2019) 68–75.

[14] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).

[15] G. E. Hinton, et al., Learning distributed representations of concepts, in: Proceedings of the eighth annual conference of the cognitive science society, volume 1, Amherst, MA, 1986, p. 12.

[16] S. Wang, W. Zhou, C. Jiang, A survey of word embeddings based on deep learning, Computing 102 (2020) 717–740.

[17] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6382–6388.

[18] J. Ma, L. Li, Data augmentation for chinese text classification using back-translation, in: Journal of Physics: Conference Series, volume 1651, IOP Publishing, 2020, p. 012039.

[19] S. Edunov, M. Ott, M. Auli, D. Grangier, Understanding back-translation at scale, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 489–500.

[20] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, Q. V. Le, Qanet: Combining local convolution with global self-attention for reading comprehension, arXiv preprint arXiv:1804.09541 (2018).

[21] D. R. Beddiar, M. S. Jahan, M. Oussalah, Data expansion using back translation and paraphrasing for hate speech detection, Online Social Networks and Media 24 (2021) 100153.

[22] F. Lomonaco, G. Donabauer, M. Siino, Courage at checkthat! 2022: Harmful tweet detection using graph neural networks and electra, in: Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022, p. 1.

[23] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926 (2017).

[24] P. Pradhyumna, G. Shreya, et al., Graph neural network (gnn) in image and video understanding using deep learning for computer vision applications, in: 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2021, pp. 1183–1189.

[25] M. Siino, M. La Cascia, I. Tinnirello, Whosnext: Recommending twitter users to follow using a spreading activation network based approach, in: 2020 International Conference on Data Mining Workshops (ICDMW), IEEE, 2020, pp. 62–70.

[26] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.

[27] 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, p. 1. doi:10.1007/978-3-030-22948-1_5.

A. Online Resources

The source code of our model is available via

- GitHub

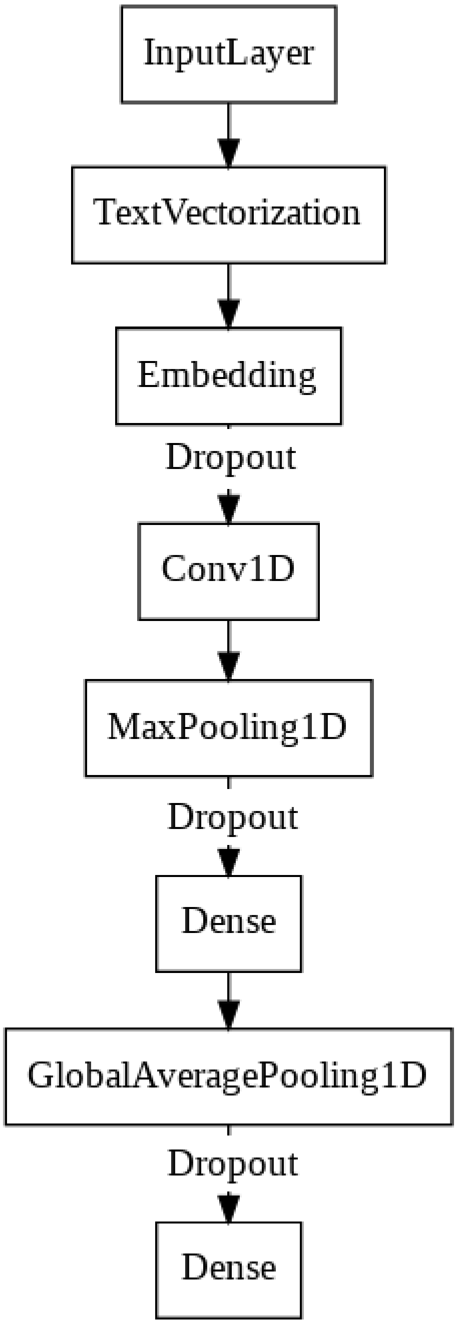

Figure 2: The shallow CNN used in this work and discussed in [10].