An SVM Ensamble Approach to Detect Irony and Stereotype Spreaders on Twitter

Notebook for PAN at CLEF 2022

Daniele Croce[1,2], Domenico Garlisi[1,2] and Marco Siino[1]

[1]Università degli Studi di Palermo, Dipartimento di Ingegneria, Palermo, Italy
[2]CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Parma, Italy

Abstract
The problem we address in this work is classifying whether a Twitter user has spread Irony and Stereotype or not. We used a text vectorization layer to generate Bag-Of-Words sequences. Then such sequences are passed to three different text classifiers (Decision Tree, Convolutional Neural Network, Naive Bayes). Our final classifier is an SVM. To test and validate our approach we used the dataset provided for the author profiling task organized by PAN@CLEF 2022. Our team (*missino*) submitted the predictions on the provided test set to participate at the shared task. Over several cross fold validation our approach was able to reach a maximum binary accuracy on the best validation split equal to 0.9474. On the test set provided for the shared task our model is able to reach an accuracy of 0.9389.

Keywords
PAN2022, author profiling, SVM, ensamble, text classification, irony, stereotype

1. Introduction

The organizers of "Profiling spreaders of irony and stereotype on Twitter" task [1] at PAN 2022 [2] provided 200 tweets per 420 users, where half of the users are confirmed to have spread Irony and Stereotype (IS) on Twitter and the other half have not. Task participants are required to develop techniques to separate the IS spreaders from the non IS (nIS) spreaders. Differently from previous years, the organizers provided an English dataset only. Indeed, in the previous edition of the author profiling task, a Spanish dataset, for multilingual approaches, was also provided.

To address the task, we propose an SVM as a last stage classifier. In the first stage a text vectorization layer is used to generate Bag-Of-Words sequences. Then such sequences are passed to three different text classifiers: Decision Tree (DT), Convolutional Neural Network (CNN) and a Naive Bayes-based model (NB). Predictions made by these three classifiers are provided to the final classifier (i.e. SVM) which provides the final prediction. Our final predictions on the unlabeled samples on the provided dataset were submitted on TIRA platform [3]. The remaining of this work is organized as follows. In Section 2 we present some related works on similar text

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
marco.siino@unipa.it (M. Siino)
0000-0001-7663-4702 (D. Croce); 0000-0001-6256-2752 (D. Garlisi); 0000-0002-4453-5352 (M. Siino)

classification tasks. In Section 3 we describe our approach in detail. In Section 4 we present our results. In Section 5 we conclude our paper and in Section 6 we discuss some future work.

2. Related work

Relevant approaches about the detection of stereotypes are proposed in [4, 5] while some methods and discussions about irony detection are proposed in [6, 7].

Regarding text classification approaches, most works investigated traditional approaches such as Support Vector Machines (SVM)[8]. For example, in [9] author proposes an SVM classifier with character and word n-gram features to determine whether the author of a Twitter feed is keen to be a spreader of fake news. In [10] authors developed systems that use character n-grams as features in combination with a linear SVM and Logistic Regression (LR) depending on the language (e.g., English or Spanish). Using SVM and LR, authors in [11], explored how powerful and scalable matrix factorization-based classification can be in a multilingual setting, where the learner is presented with the data from multiple languages simultaneously. Other SVM-based approaches for shared tasks hosted at PAN are discussed in [12, 13, 14, 15, 16].

Decision tree is one of the most common machine learning approach for text classification; some relevant application and works are discussed in [17, 18, 19].

Convolutional Neural Networks (CNNs) have also been proved to be effective on several text classification tasks. In the 2021 edition of the author profiling task organized by PAN [20], the winning team [21] used a shallow CNN to detect hate speech spreaders on Twitter. In a similar task authors used a Multi-Channel CNN to detect patronizing and condescending language [22].

Finally, ensembles of classifiers have been used by various authors in literature, such as SVM, Random Forest and Naive Bayes with XGBoost [23]; Decision Tree, Random Forest and XGB [24]; SVM, Logistic Regression, Random Forest and Extra Tree [25, 26]. However, depending on the specific classification task, performances of each available architecture can differ considerably.

It is worth noting a relevant increase in the use of Explainable Artificial Intelligence methods in place of the black box-based approaches. A few of these methods are based on graph and used in real-world applications such as text classification [27], traffic prediction [28], computer vision [29] and social networking [30].

3. Our approach

3.1. The dataset

A shown in Table 1, the PAN 2022 Profiling Irony and Stereotype Spreaders on Twitter task consists of an English corpus containing 600 XML files. Each of these files contains 200 tweets from a Twitter user. Because of the size of the corpus, we avoided splitting the corpus into a training and a development set. Instead, we used cross-validation techniques to prevent overfitting. The dataset provided all URLs, hashtags and user mentions which were changed to standardized tokens. However we performed an additional preprocessing step on the provided dataset to remove the tag *documents*, *CDATA* and *author* from each sample in the dataset. Finally we lowercased all characters in the dataset.

Table 1
Number of sample/authors in the dataset provided.

Language	Labelled (Training set)	Unlabelled (Test set)	Total
English	420	180	600

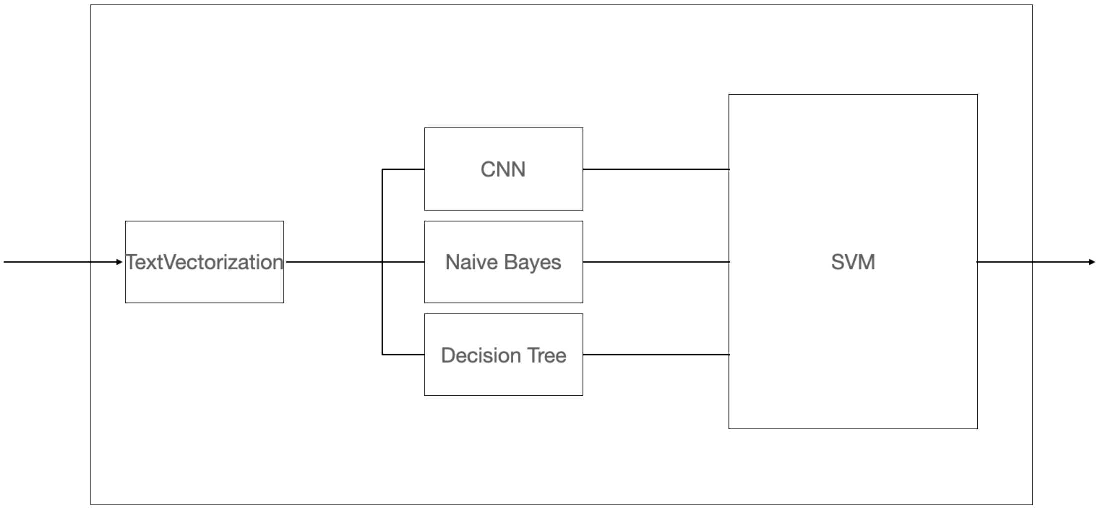

Figure 1: Overview of the proposed ensamble model.

3.2. The proposed model

Our proposed model is shown in Figure 1. After a *TextVectorization*[1] layer we provided the tokenized text to the CNN, Naive Bayes and Decison Tree classifiers.

The Naive Bayes and Decision Tree are implemented using the *scikit-learn* package while for the CNN we implemented the shallow network discussed in [21].

After we collected the prediction from CNN, Naive Bayes and the Decision Tree on each sample of the dataset, we provided these predictions as input to an SVM. Same pipeline is implemented both for training and testing phase of our model. During the training phase we provide predictions related labels to the SVM. For the test phase we provided the unlabelled sample from the test set to the CNN, Naive Bayes and the Decision Tree. Providing the output predictions from these classifiers as input to the SVM, we collected the final prediction to be submitted on TIRA.

[1]https://www.tensorflow.org/api_docs/python/tf/keras/layers/TextVectorization

Table 2
Results of single classifiers over 5-fold. The best results over 5-fold are expressed in terms of binary accuracy. The standard deviation over the 5-fold is shown in the latest column.

Model	Accuracy	σ
CNN	0.9079	**0.0158**
Naive Bayes	0.8947	0.0268
Decision Tree	0.8816	0.0579
SVM (ensamble)	**0.9474**	0.0377

3.3. Experimental setup

We developed our software using the Python language (version 3.7) on Google Colab[2]. To build our models we mainly used the scikit-learn[3] package, numpy[4] and TensorFlow[5] . Our code is available Google Drive as a Jupyter Notebook[6].

4. Results

In Table 2 are shown the results obtained by the single classifiers used and by the SVM ensamble on the best running fold over a 5-fold cross validation. Results of the SVM are obtained using as samples the predictions of the first layer classifier over the five folds.

As can be noted the performance of the SVM ensamble significantly outperforms single classifiers within our proposed framework. However the standard deviation over the five folds is not smaller with regards to the CNN and Naive Bayes. As communicated by the organizers, on the test set provided our model is able to reach an accuracy of 0.9389.

5. Conclusion

In this notebook, we summarized our work process of preparing a software for the PAN 2022 Profiling Irony and Stereotype Spreaders on Twitter task. To find the best performing models we performed a 5-fold cross validation over the labelled samples in the dataset. After finding the models achieving the best accuracies during the cross-validation, we fitted these on the best fold training set. Then we trained an SVM on the predictions of the three chosen classifiers. So for our final software, we decided to create a model which was a classifiers taking as input the predictions of three parallel classifiers (CNN, Naive Bayes, Decision Tree). For each sub-model, we used grid search and cross-validation to find the best performing parameters and fitted the models on the best training data with these parameters. To get a final prediction for each user, we trained an SVM that used the predictions of the sub-models as features. Using the ensemble

[2]https://colab.research.google.com/
[3]https://scikit-learn.org/
[4]https://numpy.org/
[5]https://www.tensorflow.org
[6]https://colab.research.google.com/drive/1EWCxAHxWWAkFg-Y8dveXuxrh82hyOh96?usp=sharing

model, we were able to achieve improved performances over all tests. Overall, our final model was able to identify IS spreaders with a maximum binary accuracy of 0.9474 on a single fold.

6. Future work

We assume that it would be beneficial to conduct some qualitative research about the tweets in the dataset to better understand the vocabulary used by IS and nIS spreaders. Another promising direction for achieving higher accuracy in profiling IS spreaders is to test several other first-stage classifiers. Perhaps implementing some transformer-based model [31]. Such models could be employed both as a first stage classifier or as the final ensemble predictor. Another way could be using some pre-trained embedding from common transformer as ELECTRA[32] or RoBERTa[33] instead of a simple *Text Vectorization* layer.

Another interesting aspect to further investigate is about the number of relevant tweets containing irony and stereotype in the feed of authors labelled as IS spreaders. Finally, some additional form of noise removal from the actual dataset could be performed to improve the overall performances of the proposed ensamble.

CRediT Authorship Contribution Statement

Daniele Croce: Writing - review & editing. **Domenico Garlisi:** Writing - review & editing. **Marco Siino:** Conceptualization, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - Original draft, Writing - review & editing.

References

[1] 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.

[2] 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.

[3] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, Tira integrated research architecture, in: Information Retrieval Evaluation in a Changing World, Springer, 2019, pp. 123–160.

[4] 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.

[5] 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.

[6] S. Zhang, X. Zhang, J. Chan, P. Rosso, Irony detection via sentiment-based transfer learning, Information Processing & Management 56 (2019) 1633–1644.
[7] 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.
[8] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM transactions on intelligent systems and technology (TIST) 2 (2011) 1–27.
[9] J. Pizarro, Using n-grams to detect fake news spreaders on twitter., in: CLEF (Working Notes), 2020.
[10] I. Vogel, M. Meghana, Fake news spreader detection on twitter using character n-grams. notebook for pan at clef 2020, arXiv preprint arXiv:2009.13859 (2020).
[11] B. Koloski, S. Pollak, B. Skrlj, Multilingual detection of fake news spreaders via sparse matrix factorization., in: CLEF (Working Notes), 2020.
[12] D. Espinosa, H. Gómez-Adorno, G. Sidorov, Profiling fake news spreaders using characters and words n-grams notebook for pan at clef 2020, volume 2696, 2020.
[13] J. L. Fernández, J. A. L. Ramírez, Approaches to the profiling fake news spreaders on twitter task in english and spanish., in: CLEF (Working Notes), 2020.
[14] A. Hashemi, M. R. Zarei, M. R. Moosavi, M. Taheri, Fake news spreader identification in twitter using ensemble modeling. notebook for pan at clef 2020., in: CLEF (Working Notes), 2020.
[15] M. Lichouri, M. Abbas, B. Benaziz, Profiling fake news spreaders on twitter based on tfidf features and morphological process, in: CEUR Workshop Proceedings, volume 2696, 2020.
[16] E. Fersini, J. Armanini, M. D'Intorni, Profiling fake news spreaders: Stylometry, personality, emotions and embeddings., in: CLEF (Working Notes), 2020.
[17] J. Su, H. Zhang, A fast decision tree learning algorithm, in: Aaai, volume 6, 2006, pp. 500–505.
[18] S. Bahassine, A. Madani, M. Kissi, An improved chi-sqaure feature selection for arabic text classification using decision tree, in: 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA), IEEE, 2016, pp. 1–5.
[19] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, Journal of Applied Science and Technology Trends 2 (2021) 20–28.
[20] F. Rangel, G. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling hate speech spreaders on twitter task at pan 2021, in: CLEF, 2021.
[21] 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.
[22] 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: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Association for Computational Linguistics, 2022.
[23] T. Niven, H.-Y. Kao, H.-Y. Wang, Profiling spreaders of disinformation on twitter: Ikmlab and softbank submission., in: CLEF (Working Notes), 2020.
[24] C. Ikae, J. Savoy, Unine at pan-clef 2020: Profiling fake news spreaders on twitter., in: CLEF (Working Notes), 2020.

[25] A. Shrestha, F. Spezzano, A. Joy, Detecting fake news spreaders in social networks via linguistic and personality features, in: Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum, 2020.
[26] M. Siino, I. Tinnirello, M. La Cascia, T100: A modern classic ensemble to profile irony and stereotype spreaders, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.
[27] 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.
[28] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926 (2017).
[29] 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.
[30] 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.
[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[32] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).
[33] 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, arXiv preprint arXiv:1907.11692 (2019).