Profiling irony and stereotype spreaders on Twitter based on term frequency in tweets

Dhaval Taunk[1], Sagar Joshi[1] and Vasudeva Varma[1]

[1]*International Institute of Information Technology, Hyderabad*

Abstract

The use of stereotypes, irony, mocking and scornful language is prevalent on social media platforms such as Twitter. Identification or profiling of users who are involved in the spread of such content is beneficial for monitoring its spread. In our work, we study the problem of profiling irony and stereotype spreaders on Twitter as a part of the PAN shared task in CLEF 2022. We experiment with machine learning models applied on a TF-IDF representation of user tweets, and find Random Forest to be the best working one.

Keywords

Irony, Stereotype, Random Forest, TF-IDF, Twitter

1. Introduction

Metaphorical and figurative style of writing presents a subtle way of communicating across a message on social media. The nature of the message being conveyed can be distinguished based on the use of such linguistic nuances in a message being propagated. Their usage in directed ways can make the message to be either generally harmless, potentially hurtful, or even inherently toxic in nature. The identification of such content is beneficial not only for shielding often targeted demographic groups, but also for reasons such as a better understanding of the textual content on social media. As pointed out in [1], a better understanding of sarcasm and irony in text can help improve sentiment analysis because of the difficulty in semantic understanding of text introduced by sarcasm. Apart from the understanding of sarcastic content, a profiling of users who tend to propagate such content can benefit to understand differences in the patterns of sarcasm originating from different sources. It can also ease tracking users indulging in the spread of toxic content through subtle means.

In our work, we focus on the task of profiling spreaders of ironical and stereotypical content on Twitter, as a part of the PAN [2] shared task [3] in CLEF 2022. In this task, we work on a Twitter feed of a set of users in English containing user-level annotations to indicate if the user is a spreader of ironical and stereotypical content. In our implementation of the solution, we treat all the user tweets as a single input and experiment with basic text preprocessing followed by a simple TF-IDF representation. This essentially models the task with simple term-frequency based information from past tweets. Inspite of the simplicity of modeling, however, experiments

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

dhaval.taunk@research.iiit.ac.in (D. Taunk); sagar.joshi@research.iiit.ac.in (S. Joshi); vv.iiit.ac.in (V. Varma)
https://dhavaltaunk08.github.io/ (D. Taunk); https://www.iiit.ac.in/~vv (V. Varma)

CEUR Workshop Proceedings (CEUR-WS.org)

with simple, lightweight machine learning models gave encouraging results on this task. In the subsequent sections, we detail our methodology, experiments and results.

2. Related Work

Author profiling is a well-studied problem for the identification of social media users indulging in the spread of fake news, hate speech, gender bias, and other related aspects. Diego et. al. [4] attempted to solve the problem of detecting hate speech against immigrants and women on Twitter in a multilingual setting. To determine if a tweet is hateful or not, they applied SVM on top of TF-IDF representations. Rangel et. al. [5] experimented with several algorithms on labeled Twitter data to detect fake news. Random label generation, LSTM, Neural Network with word n-gram features, SVM with character level n-gram features, and other models were tested by the authors.

Carracedo et. al. [6] solve the hate speech detection problem by using a Support Vector Machine (SVM) classifier over n-gram features. For detecting irony, Reyes et. al. [7] constructed a three-layer model. They create signatures using three linguistic characteristics: pointedness, counterfactuality, and temporal compression. In emotional circumstances, three language properties are activation, imagery, and pleasantness. Unpredictability based on two literary characteristics: temporal and contextual imbalance. They use the above-mentioned layers to turn text into feature vectors and estimate whether or not an input text contains irony.

3. Methodology

The aim of this task is to determine whether or not a given Twitter user is using irony/stereotype in their tweets. This can help in easing the identification of sources creating or propagating such content. In the following sections, we describe our approach on modeling this problem with our experiment based on term-frequency based text preprocessing in conjunction with classical machine learning algorithms.

3.1. Data Description

The dataset provided by the task organizers had 420 training instances, each having 200 past tweets for a Twitter user - thus corresponding to 420 user profiles to be labeled. 210 of the user profiles belonged to the class that spreads irony, whereas the remaining 210 did not, hence the dataset to be worked upon was balanced. The test set contained 180 user profiles also bearing 200 tweets per user (but with labels not included as a part of the dataset). The tweets were all monolingual in English. Since there was no explicit dataset provided for validation, we split the given training dataset into train and dev split with a split ratio of 80:20.

3.2. Text Pre-processing

In the data provided, user ids, urls and hashtags were already replaced by #USER#, #URL# and #HASHTAG# placeholders respectively. As a part of our preprocessing, we tried out completely removing the placeholders versus keeping them as a part of the pipeline to see the impact of these features have in determining a piece of content as ironic or stereotypical.

4. Experiments

This section explains the approach we followed for the task. Our experimentation pipeline consisting of pre-processing followed by featured extraction and ML-based modeling can be described as below.

1. **Pre-processing**: Each of the tweets in a user profile of 200 tweets is preprocessed to either remove or keep the placeholders as described in Section 3.2. We found keeping the placeholders to perform better, hence we adopted this pre-processing in calculating the features.
2. **Feature Extraction**: A pre-processed tweet text is represented using a TF-IDF vector as a simple term-frequency based tweet representation and the representation of a user is calculated as a summation of all the tweet vectors. The TF-IDF vector values are determined based on the frequency information of words in the train split for all the words in the train split vocabulary. The terms not present in the train split vocabulary are ignored when computing the feature vectors for dev and test splits.
3. **Modeling**: This step involved trying out different machine learning techniques on the TF-IDF features extracted from the data. We experimented with classifiers based on Logistic Regression, K Nearest Neighbors, Support Vector Machines, Random Forest and XGBoost.

For training and making predictions on the test set, the Scikit-learn [8] package was utilised. Different hyperparameters settings were experimented in training the model. To begin, all of the models were trained using the Scikit-learn package's default parameters.

The best hyperparameter settings were determined for each of the models used using Random Search CV on an 80/20 train/dev split. The selected hyperparameters for finetuning and the range of the values tried out are summarized in Table 1.

4.1. Results

The results on the dev set using the best model configurations obtained after hyperparameter tuning are shown in Table 2.

Accuracy was the metric chosen by the task organizers for final evaluation. From the results, it can be seen that Random Forest Classifier gave the best performance on the metric as well as in the precision score for the positive class. F1 score had a tie-up between Random Forest Classifier and XGBoost Classifier, the latter of which turned out to be pretty competitive in the overall performance, while also having the highest recall. K Neighbors Classifier gave the worst performance across all the metrics. The best performing Random Forest Classifier was used to take predictions on the test set and was submitted for evaluation on the TIRA [9] platform, which gave an accuracy of 95%.

Table 1
Hyperparameter variables and ranges of values for which hyperparameter finetuning was performed across models

Model Name	Hyperparameter	Values
Logistic Regression	C	1e-03, 1e-02, 1e-01, 1e+00, 1e+01, 1e+02, 1e+03
	penalty	l1, l2
K Neighbors Classifier	n_neighbors	3,4,5,6,7
	weights	uniform, distance
	algorithm	auto, ball_tree, kd_tree, brute
SVM	C	0.01,0.1,1,10,100
	kernal	linear, poly, rbf
	gamma	auto, scale
Random Forest	n_estimators	200, 211, 222, 233, 244, 255, 266, 277, 288, 300
	max_features	auto, sqrt
	max_depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110
	min_samples_split	2, 5, 10
	min_samples_leaf	1, 2, 4
	bootstrap	True, False
XGBoost	n_estimators	200, 211, 222, 233, 244, 255, 266, 277, 288, 300
	max_features	auto, sqrt
	max_depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110
	min_samples_split	2, 5, 10
	min_samples_leaf	1, 2, 4
	bootstrap	True, False

Table 2
Results obtained on dev set after training using the best model configurations determined by hyperparameter tuning

Model	Accuracy	F1 Score	Precision	Recall
Logistic Regression	87	85	89	82
K Neighbors Classifier	77	75	78	72
SVM	88	87	89	85
Random Forest Classifier	**90**	**89**	**91**	87
XGBoost Classifier	89	**89**	86	**92**

5. Conclusion

Based on our ML-based experiments on term frequency representation of user tweets, we were able to achieve a respectable performance which was consistent across datasets used for validation and testing. Hence, if the dataset distribution used for the task matches with the data encountered in actual, in-the-wild tweets, a user-profiling with system good performance can be achieved with minimalistic lightweight techniques.

References

[1] M. Sykora, S. Elayan, T. W. Jackson, A qualitative analysis of sarcasm, irony and related #hashtags on twitter, Big Data & Society 7 (2020) 2053951720972735. URL: https://doi.org/10.1177/2053951720972735. doi:10.1177/2053951720972735. arXiv:https://doi.org/10.1177/2053951720972735.

[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: 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] 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.

[4] B. Diego, A. Oscar, I. C. A., GSI-UPM at SemEval-2019 task 5: Semantic similarity and word embeddings for multilingual detection of hate speech against immigrants and women on Twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 396–403. URL: https://aclanthology.org/S19-2070. doi:10.18653/v1/S19-2070.

[5] F. M. R. Pardo, 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. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_267.pdf.

[6] À. A. Carracedo, R. J. Mondéjar, Profiling hate speech spreaders on twitter, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 1801–1807. URL: http://ceur-ws.org/Vol-2936/paper-152.pdf.

[7] A. Reyes, P. Rosso, On the difficulty of automatically detecting irony: beyond a simple case of negation, Knowl. Inf. Syst. 40 (2014) 595–614. URL: https://doi.org/10.1007/s10115-013-0652-8. doi:10.1007/s10115-013-0652-8.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[9] 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.