Understanding and Interpreting the Impact of User Context in Hate Speech Detection

Edoardo Mosca
TU Munich,
Department of Informatics,
Germany
edoardo.mosca@tum.de

Maximilian Wich
TU Munich,
Department of Informatics,
Germany
maximilian.wich@tum.de

Georg Groh
TU Munich,
Department of Informatics,
Germany
grohg@in.tum.de

Abstract

As hate speech spreads on social media and online communities, research continues to work on its automatic detection. Recently, recognition performance has been increasing thanks to advances in deep learning and the integration of user features. This work investigates the effects that such features can have on a detection model. Unlike previous research, we show that simple performance comparison does not expose the full impact of including contextual- and user information. By leveraging explainability techniques, we show (1) that user features play a role in the model's decision and (2) how they affect the feature space learned by the model. Besides revealing that—and also illustrating *why*—user features are the reason for performance gains, we show how such techniques can be combined to better understand the model and to detect unintended bias.

1 Introduction

Communication and information exchange between people is taking place on online platforms at a continuously increasing rate. While these means allow everyone to express themselves freely at any time, they are massively contributing to the spread of negative phenomenons such as online harassment and abusive behavior. Among those, which are all to discourage, online hate speech has attracted the attention of many researchers due to its deleterious effects (Munro, 2011; Williams et al., 2020; Duggan, 2017).

The extremely large volume of online content and the high speed at which new one is generated exclude immediately the chance of content moderation being done manually. This realization has naturally captured the attention of the *Machine Learning* (ML) field, seeking to craft automatic and scalable solutions (MacAvaney et al., 2019; Waseem et al., 2017; Davidson et al., 2017).

Methods for detecting hate speech and similar abusive behavior have been thus on the rise, consistently improving in terms of performance and generalization (Schmidt and Wiegand, 2017; Mishra et al., 2019b). However, even the current state of the art still faces limitations in accuracy and is yet not ready to be deployed in practice. Hate speech recognition remains an extremely difficult task (Waseem et al., 2017), in particular when the expression of hate is implicit and hidden behind figures of speech and sarcasm.

Alongside language features, recent works have considered utilizing user features as an additional source of knowledge to provide detection models with context information (Fehn Unsvåg and Gambäck, 2018; Ribeiro et al., 2018). As a general trend, models incorporating context exhibit improved performance compared to their pure text-based counterparts (Mishra et al., 2018, 2019a). Nevertheless, the effect, which these additional features have on the model, has not been interpreted or understood yet. So far, models have mostly been compared only in terms of performance metrics. The goal of this work is to shed light on the impact generated by including user features—or more in general context—into hate speech detection methods. Our methodology heavily relies on a combination of modern techniques coming from the field of *eXplainable Artificial Intelligence* (XAI).

We show that adding user and social context to models is the reason for performance gains. We also explore the model's learned features space to understand how such features are leveraged for detection. At the same time, we discover that models incorporating user features suffer less from bias in the text. Unfortunately, those same models contain a new type of bias that originates from adding user information.

Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media, pages 91–102
Online Workshop, June 10, 2021. ©2021 Association for Computational Linguistics

2 Related Work

2.1 Explainability for Recognition Models

A limited amount of research has focused on applying XAI techniques to the hate speech recognition case. For instance, Wang (2018) adapts a number of explainability techniques from the computer vision and applies them to a hate speech classifier trained on Davidson et al. (2017). Feature occlusion was used to highlight the most relevant words for the final classifier prediction and activation maximization selected the terms that the classifier captured and judged as relevant at a dataset-level. Vijayaraghavan et al. (2019) constructs an interpretable multi-modal detector that uses text alongside social and cultural context features. The authors leverage attention scores to quantify the relevance of different input features. Wich et al. (2020) applies post-hoc explainability on a custom dataset in German to expose and estimate the impact of political bias on hate speech classifiers. More in detail, left- and right-wing political bias within the training data is visualized via DeepSHAP-based explanations (Lundberg and Lee, 2017).

MacAvaney et al. (2019) combines together multiple simple classifiers to assemble a transparent model. Risch et al. (2020) reviews and compares several explainability techniques applied to hate speech classifiers. Their experimentation includes popular post-hoc approaches such as LIME (Ribeiro et al., 2016) and LRP (Bach et al., 2015) as well as self-explanatory detectors (Risch et al., 2020).

For our use case, we apply *post-hoc explainability* approaches (Lipton, 2018). We use external techniques to explain models that would otherwise be black-boxes (Arrieta et al., 2020). In contrast, *transparent models* are interpretable thanks to their intuitive and simple design.

2.2 Context Features for Hate Speech Detection

Models have been continuously improving since the first documented step towards automatic hate speech detection Spertus (1997). The evolution of recognition approaches has been favored by advances in *Natural Language Processing* (NLP) research (Mishra et al., 2019b). For instance, s.o.t.a detectors like Mozafari et al. (2020) exploit high-performing language models such as BERT (Devlin et al., 2019).

A different research branch took an alternative path and explored the inclusion of social context alongside text. These additional features are usually referred to with the terms *user features*, *context features*, or *social features*. Some tried incorporating the gender (Waseem, 2016) and the profile's geolocation and language (Galán-García et al., 2016). Others instead utilized the user's number of followers or friends (Fehn Unsvåg and Gambäck, 2018).

Modeling users' social and conversational interactions via their corresponding graph was also shown to be rewarding (Mishra et al., 2019b; Cecillon et al., 2019). Ribeiro et al. (2018) creates additional features by measuring properties like betweenness and eigenvector centrality. Mishra et al. (2018) and Mishra et al. (2019a) instead fed the graph directly to the model either embedded as matrix or via using graph convolutional neural network (Hamilton et al., 2017).

While previous work explored the usage of a wide range of context features (Fehn Unsvåg and Gambäck, 2018), detection models have only been compared in terms of performance metrics. Besides accuracy, researchers have not focused on other changes that such features could have on the model. Our work shows that indeed this addition entails a large impact on the recognition algorithm's behavior and substantially changes its characteristics.

3 Experimental Setup

In this section, we describe in detail the different datasets and detection models that we include in our interpretability-driven analysis.

3.1 Data and Preprocessing

Previous research has produced several datasets to support further developments in the hate speech detection area (Founta et al., 2018; Warner and Hirschberg, 2012). Some became relatively popular to benchmark and test new ideas and improvements in recognition techniques. For our experimentation, we pick the DAVIDSON (Davidson et al., 2017) and the WASEEM (Waseem and Hovy, 2016) datasets. The choice was motivated by their variety of speech classes and popularity as detection benchmarks.

Both benchmarks consist of a collection of tweets coupled with classification tasks with three possible classes. DAVIDSON contains $\sim 25,000$ tweets of which $1,430$ are labeled as *hate*, $19,190$ as *offensive*, and $4,163$ as *neither* (Davidson et al., 2017). As classification outcomes in WASEEM instead, we have *racism*, *sexism*, and *neither*. The three classes contain $3,378$, $1,970$, and $11,501$ tweets respectively (Waseem and Hovy, 2016). We were not able to retrieve the remaining 65 of the original $16,914$ samples.

We follow the same preprocessing steps for both datasets. First, terms belonging to categories like *url, email, percent, number, user,* and *time* are annotated via a category token. For instance, "*341*" is replaced by "*<number>*". After that, we apply word segmentation and spell correction based on Twitter word statistics. Both methods and statistics were provided by the *ekphrasis* [1] text preprocessing tool (Baziotis et al., 2017).

In addition to the tweets that represent the text (or content) component of our input features, we also retrieve information about the tweet's authors and their relationships. In a similar fashion as done in Mishra et al. (2018), we construct a *community graph* $G = (V, E)$ where each node represents a user and two nodes are connected if at least one of the two users follows the other one. We were able to retrieve $|V| = 6,725$ users and $|E| = 19,597$ relationships for DAVIDSON, while for WASEEM we have $|V| = 2,024$ and $|E| = 9,955$.

The respective average node degrees are $2,914$ and $4,918$ and the overall graphs' densities:

$$D = \frac{2 \cdot |E|}{|V|(|V| - 1)}$$

are 0.00087 and 0.00486 respectively.

We immediately notice that both graphs are very sparse. In particular, we have $3,393$ users not connected to anyone in DAVIDSON and 927 in WASEEM. For reference, Mishra et al. (2018) achieves a graph density of 0.0075 on WASEEM, with only ~ 400 authors being solitary, i.e. with no connections. We assume the difference is reasonable as data availability considerably decreases over time.

3.2 Detection Models

Our experimentation and findings are based on the comparison of two detection models, one that solely relies on text features and one that instead incorporates context features. To better capture their behavioral differences, we build them to be relatively simple and also to not differ in the text-processing part.

The first model, shown in figure 1, computes the three classification probabilities only based on the tweets' content. The input text is fed to the model as *Bag of Words* (BoW), which is then processed by two fully connected layers. We refer to this model as *text model*.

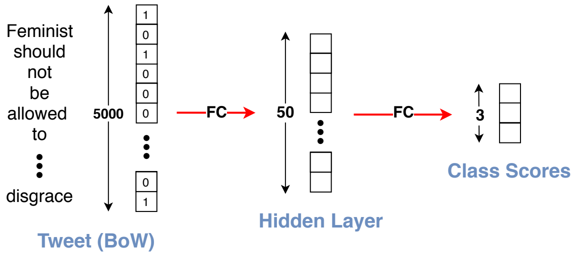


Figure 1: Architecture of the text model.

The second model instead leverages the information coming from three input sources: the tweet's text, the user's vocabulary, and the follower network. The first input is identical to what is fed to the text model. The second is constructed from all the tweets of the author in the dataset and aims to model their overall writing style. Concretely, we merge the tweets' BoW representations, i.e. we apply a logical-OR to their corresponding vectors. The third is the author's follower network and describes their online surrounding community. On a more technical note, this can be extracted as a row from the adjacency matrix of our community graph described in section 3.1. Note that s.o.t.a hate speech detector used similar context features (Mishra et al., 2018, 2019a). We refer to this model as *social model*.

As sketched in figure 2, the different input sources are initially processed separately in the model's architecture. After the first layer, the intermediate representations from the different branches are concatenated together and fed to two more layers to compute the final output. Note that the text- and social models have the same dimensions for their final hidden layer and can be seen as equivalent networks working on different inputs.

4 Proposed Analysis

We now describe our methodology in detail. Recall that our models differ precisely on the usage of user features. As we will see shortly, their comparison beyond accuracy measurements sheds light on the different model properties and hence on the potential impact of incorporating context features.

[1] https://github.com/cbaziotis/ekphrasis

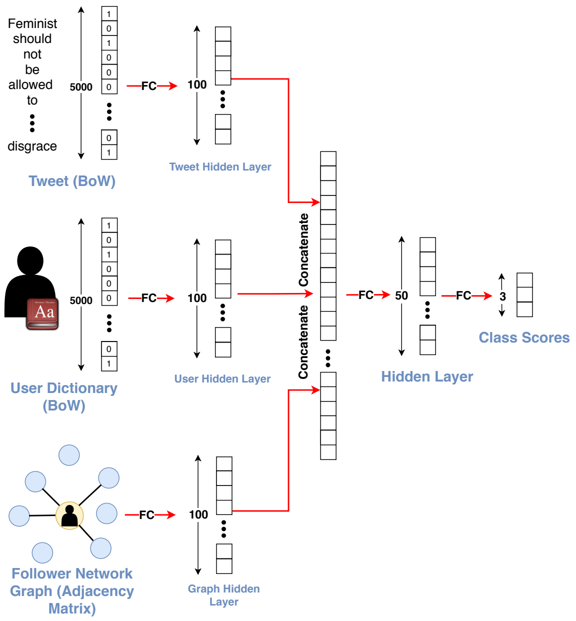


Figure 2: Architecture of the social model.

4.1 Training and Performance

We apply the same training and testing procedure to all models and datasets. We keep the 60% of the data for training while splitting the remaining equally between validation and test set, i.e. 20% each.

Tables 1 and 2 report our results in terms of F1 scores for WASEEM (Waseem and Hovy, 2016) and DAVIDSON (Davidson et al., 2017) respectively. To increase our confidence in their validity, we average the performance over five runs with randomly picked train/validation/test sets. We observe different trends for the two datasets.

Speech Class	Text Model	Social Model
Racism	0.711	0.735
Sexism	0.703	0.832
Neither	0.881	0.907
Overall	0.829	0.872

Table 1: F1 Scores on Waseem and Hovy (2016).

On WASEEM, the social model considerably outperforms (by 4.3%) our text model. The performance gain is general and not restricted to any single class. Quite surprisingly, our text model performs better on racist tweets than sexist ones, although the sexism class is almost twice as big. This suggests that sexism is, at least in this case, somewhat harder to detect by just looking at the tweet content. On the contrary, our social model shows an impressive improvement in the sexism class (almost 13%), suggesting the presence of detectable patterns in sexist users and their social interactions.

Speech Class	Text Model	Social Model
Hate	0.154	0.347
Offensive	0.939	0.939
Neither	0.809	0.815
Overall	0.876	0.886

Table 2: F1 Scores on Davidson et al. (2017).

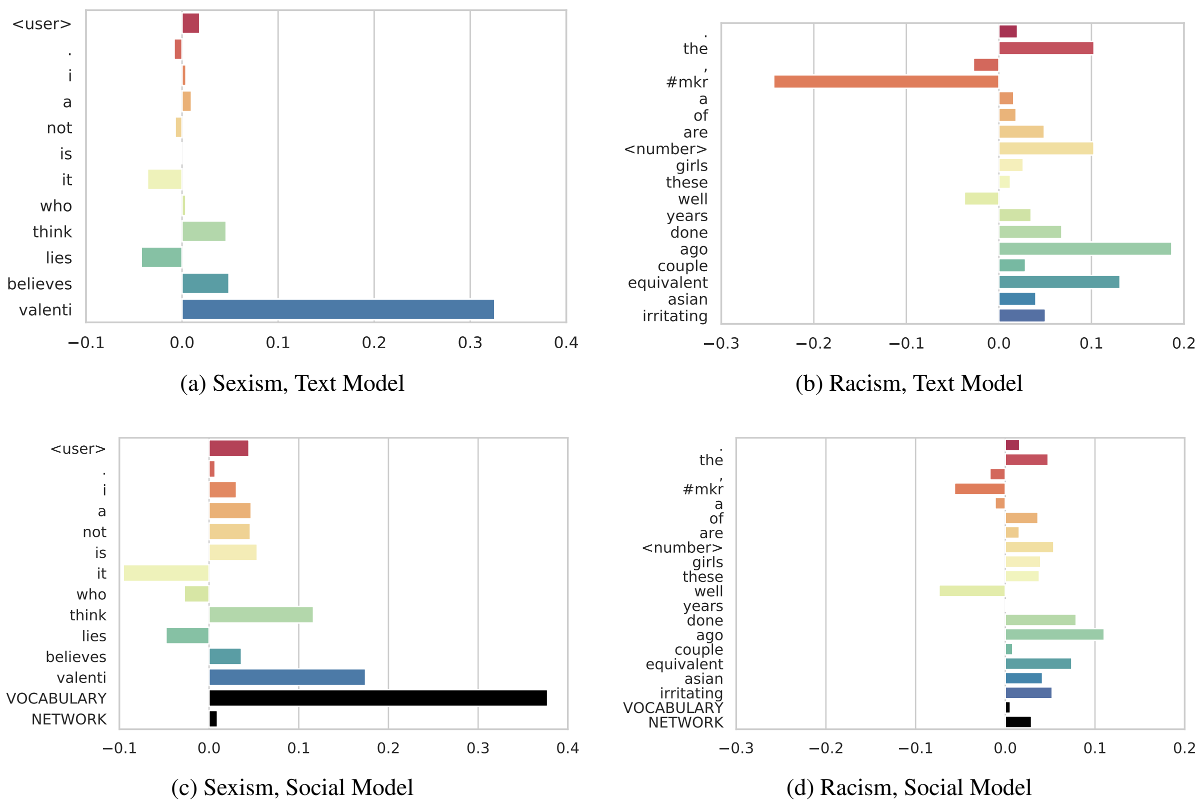
On DAVIDSON, we only observe a contained improvement (1%). Moreover, the jump in performance is restricted to the hate class, containing a tiny amount of samples. We believe the difference between the two datasets should be expected due to the lower amount of user data available for DAVIDSON. Considering these results, we focus on applying our technique on the WASEEM dataset in the remainder of this paper. Nevertheless, the respective results on DAVIDSON can be found in the appendix A. While on both datasets we do not outperform the current s.o.t.a—Mishra et al. (2019a) on WASEEM and Mozafari et al. (2020) on DAVIDSON—our results are comparable and thus satisfactory for our purposes.

4.2 Shapley Values Estimation

We now apply a first post-hoc explainability method. For each feature we calculate its corresponding *Shapley value* (Shapley, 1953; Lundberg and Lee, 2017). That is, we quantify the relevance that each feature has for the prediction of a specific output. Shapley values have been shown—both theoretically and empirically—to be an ideal estimator for feature relevance (Lundberg and Lee, 2017).

As exact Shapley values are exponentially complex to determine, we use accurate approximation methods as done in (Lundberg and Lee, 2017; Štrumbelj and Kononenko, 2014). Figure 3 shows concrete examples in which Shapley values are calculated for both models on two test tweets from WASEEM.

For our social model, we consider the user vocabulary and the follower network as single features for simplicity. Notably, the context is used by the social model and can play a significant role in its prediction. Hence, we can confirm the context features to be the reason for the performance gains. We can empirically exclude that the differences between the text- and the social model architectures justify the jump in performance.



(a) Sexism, Text Model

(b) Racism, Text Model

(c) Sexism, Social Model

(d) Racism, Social Model

Figure 3: Example of features contribution, computed via Shapley value approximation, for our text and social models. In (a) and (c) we use as input the tweet "*<user> I think Arquette is a dummy who believes it. Not a Valenti who knowingly lies.*". The sexist tweet refers to the actress Patricia Arquette, who spoke in favour of gender equality, and the feminist writer Jessica Valenti. Some words are missing in the plot as our BoW dimension is limited during preprocessing. In (b) and (d), we use the racist tweet "*These girls are the equivalent of the irritating Asian girls a couple of years ago. Well done, 7. #MKR*". The hashtag refers to the Australian cooking show "*My Kitchen Rules*".

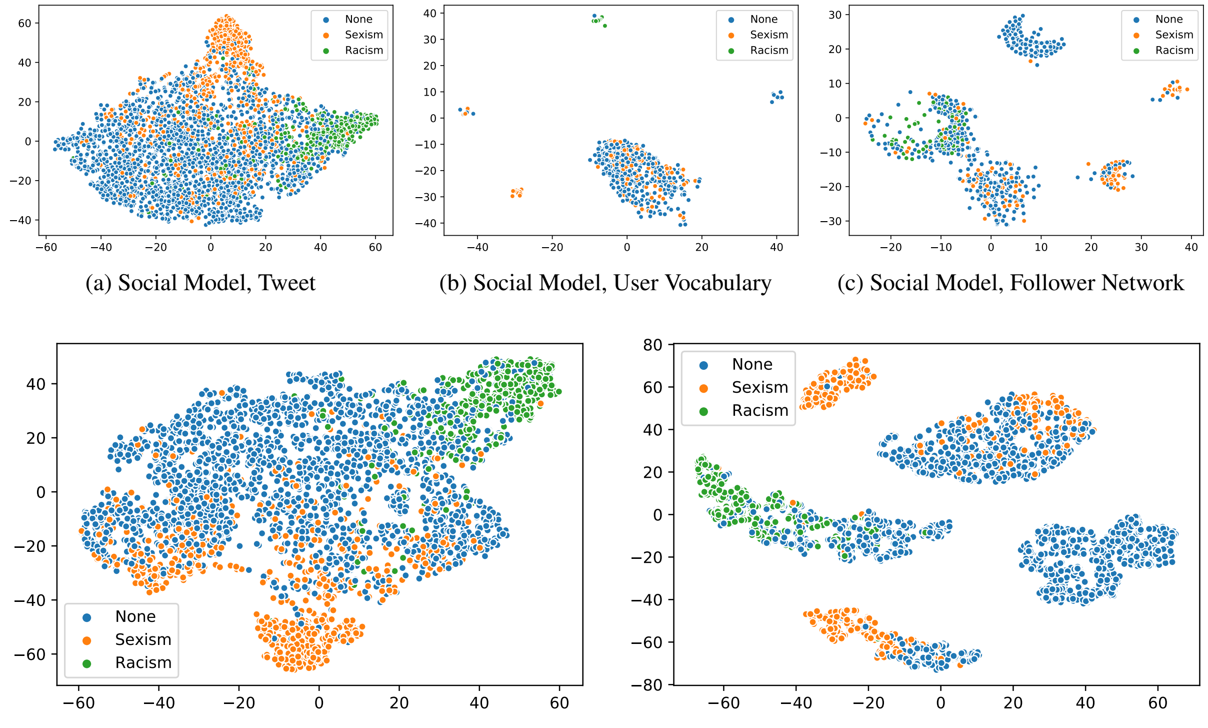
4.3 Feature Space Exploration

We have seen that detection models can benefit from the inclusion of context features. We now focus on understanding *why* this is the case. Shapley values and more in general feature attribution methods can quantify *how much* single features contribute to the prediction. Yet, alone, they do not give us any intuition to answer our why-question.

We look at the feature space learned by our models, which can be considered a global explainability technique. For our text model, we remove the last layer and feed the tweets to the remaining architecture. The output is a 50-dimensional embedding for each tweet. We employ the *t-Distributed Stochastic Neighbor Embedding* (t-SNE) (Van der Maaten and Hinton, 2008) to reduce the embeddings to two dimensions for visualization purposes.

The resulting plot, in figure 4d, shows all the tweets in a single cluster. Racist tweets look more concentrated in one area than sexist ones, suggesting that sexism is somewhat harder to detect for the model. This result is coherent with our per-class performance scores.

We apply the same procedure to the social model. In this case, we visualize the hidden layer of each separate branch as well as the final hidden layer analogous to the text model. Not surprisingly, the tweet branch (figure 4a) looks very similar to the feature space learned by our text model. The user's vocabulary branch (figure 4b) instead shows the samples distributed in well-separated clusters. Notably, racist tweets have been restricted to one cluster and we can also observe pure-sexist and pure-neither clusters. The follower network branch (figure 4c) looks similar though cluster separation is not as strong. Once more, we notice racism more concentrated than sexism, which is considerably more mixed with regular tweets. To some extent, this result is in line with the notion of *homophily* among racist users (Mathew et al., 2019).



(a) Social Model, Tweet

(b) Social Model, User Vocabulary

(c) Social Model, Follower Network

(d) Text Model

(e) Social Model

Figure 4: WASEEM tweets, colored by label, in the features space learned by our text model (d) and social model (a,b,c for the independent branches, e combined).

Intuitively, being able to divide users into different clusters based on their behavior should be helpful for classification at later layers. This is confirmed by the combined feature space plot (figure 4e). Indeed, tweets are now structured in multiple clusters instead of a single one as for our text model. Also in this case, we observe several pure or almost-pure groups.

The corresponding visualizations and results for DAVIDSON can be found in appendix A.

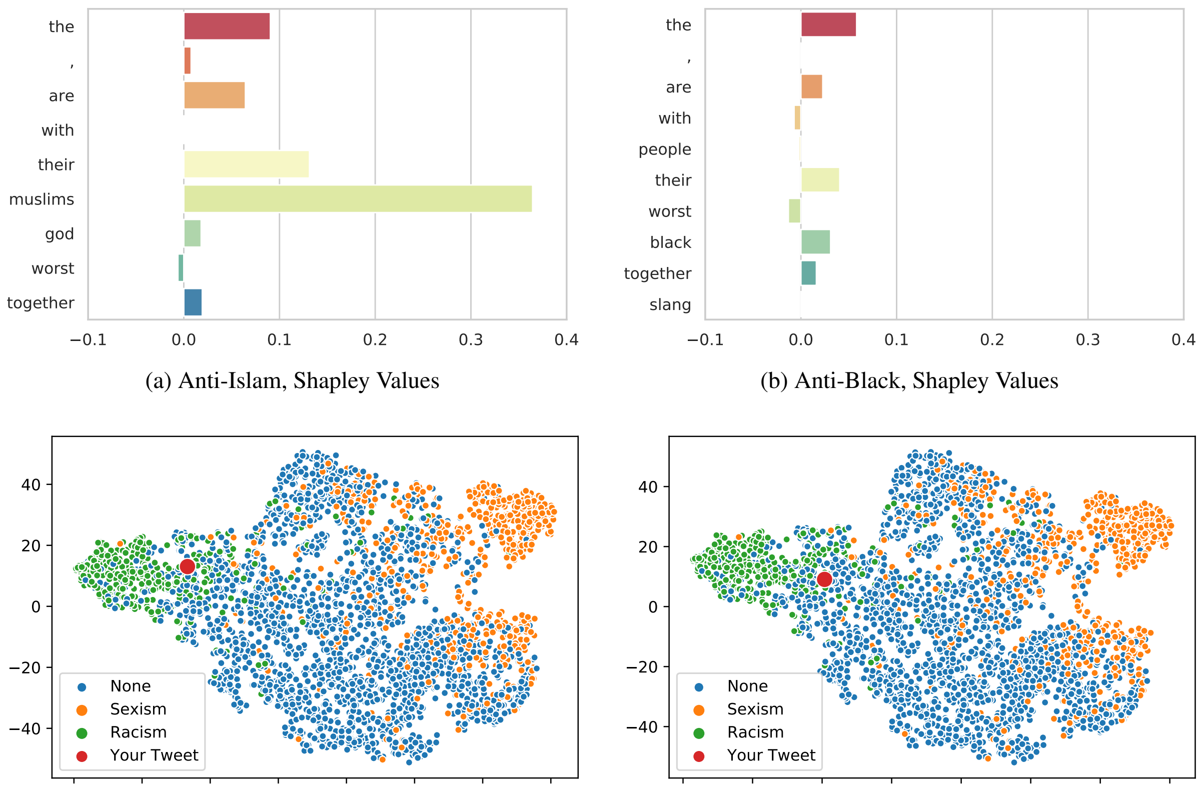
4.4 Targeted Behavioral Analysis: Explaining a Novel Tweet

We have seen how different explainability techniques convey different types of information on the examined model. Computing Shapley values and visualizing the learned feature space can also be used in combination as they complement each other. If used together, they can both quantify the relevance of each feature as well as show how certain types of features are leveraged by the model to better distinguish between classes.

So far, our explanations are relative to the datasets used for model training and testing. However, to better understand a classifier it should also be tested beyond its test set. This can be simply done by feeding the model with a novel tweet. Via artificially crafting tweets, we can check the model's behavior in specific cases. For instance, we can inspect how it reacts to specific sub-types of hate.

Let us consider the anti-Islamic tweet "*muslims are the worst, together with their god*". If fed to our model, it is classified as racist with a 75% confidence following our expectations. Figures 5a and 5c show explanations for the tweet. We can see that the word "*muslim*" plays a big role by looking at its corresponding Shapley value. At the same time, the projection of the novel tweet onto the feature space shows how the sample is collocated together with the other racist tweets by the text model.

If we now change our hypothetical tweet to be anti-black—"*black people are the worst, together with their slang*"—we observe a different model behavior (figures 5b and 5d). In fact, now the tweet is not classified as racist. No word has a substantial impact on the prediction. We can also notice a slight shift of the sample in the features space, away from the racism cluster. If changing the target of the hate changes the prediction, then the model/dataset probably contains bias against that target. Model interpretability further reveals how



(a) Anti-Islam, Shapley Values

(b) Anti-Black, Shapley Values

(c) Anti-Islam, Embedding in Latent Space

(d) Anti-Black, Embedding in Latent Space

Figure 5: Features contribution (Shapley values w.r.t. the racism class) and embedding in the text model's latent space of an islamophobic and a anti-black racist tweets. The two sentences had, according to our text model, the 75% and 24% probability of being racist respectively.

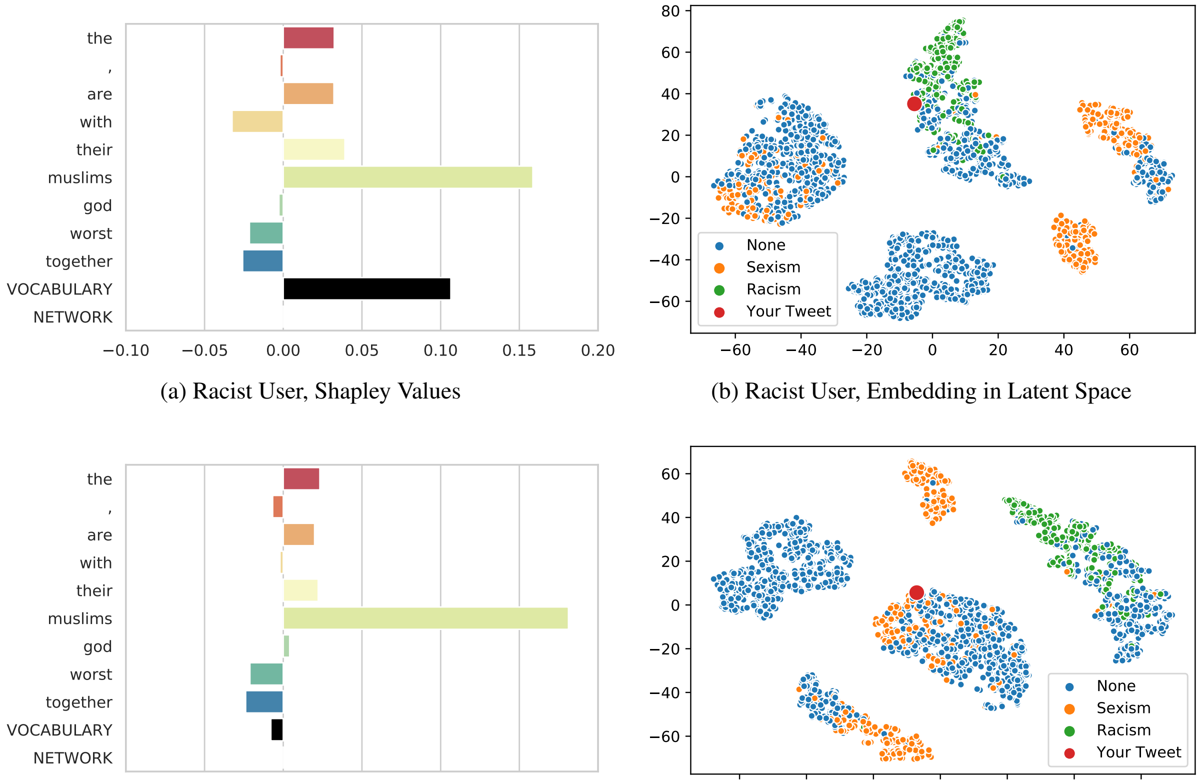
its behavior reacts to different targets.

We run the same experiment with our social model. This time, it correctly classifies the anti-black tweet as racist (55% confidence). This suggests that text bias could be mitigated by using models that do not only rely on the text input. However, the social model is much more sensitive to changes in the user-derived features. To test this, we feed the model the same tweet and only change the author that generated it. For a fair comparison, we pick one random user with other racist tweets, one random user with other sexist tweets, and one random user with no hateful tweets in the dataset. We refer to these users as racist, sexist, and regular users respectively.

Our crafted tweet is classified as racist when coming from a racist user (64%). However, it is instead judged non-hateful in both the other cases (12% and 19% for a sexist and user with no hate background respectively). Evidently, racist tweets also need some contribution from the social features to be judged as racist.

A very informative explanation comes again from both the Shapley values and the feature space exploration (figure 6). On the left side, we can see the Shapley value for the racist and regular users. Results relative to the sexist user are analogous to the regular user and reported in the supplementary material (A.3). All the words have a similar contribution to the racism class in all cases. However, the difference in the authors plays a substantial role in the decision. Only the racist user positively contributes to the racism class. On the right side of 6, we can see the embedding in the latent space for each case. Different input authors cause the tweet to be embedded in different clusters. Only in the first one the model actually considers the possibility of the tweet being racist.

Hence, while adding user-derived features might mitigate the effects of bias in the text, it generates a new form of bias that could discriminate users based on their previous behavior and hinder the model from classifying correctly hateful content.



(a) Racist User, Shapley Values

(b) Racist User, Embedding in Latent Space

(c) Regular User, Shapley Values

(d) Regular User, Embedding in Latent Space

Figure 6: Features contribution (w.r.t. racism class) and embeddings of the islamophobic tweet in the social model's latent space. The two pairs of plots are w.r.t. two predictions done with different users as input: a racist one (a,b, 64%), and a regular one (c,d, 19%).

5 Conclusion and Future Work

In our work, we investigated the effects of user features in hate speech detection. In previous studies, this was done by comparing models based on performance metric. We have shown that post-hoc explainability techniques provide a much deeper understanding of the models' behavior. In our case, when applied to two models that differ specifically on the usage of context features, the in-depth comparison reveals the impact that such additional features can have.

The two utilized techniques—*Shapley values estimation* and *learned feature space exploration*—convey different kinds of information. The first one quantifies how each feature plays a role but does not tell us what is happening in the background. The second one illustrates the model's perception of the tweets but does not provide any quantitative information for the prediction. Furthermore, we have seen that artificially crafting and modifying a tweet can be useful to examine the models' behavior in particular scenarios. In concrete examples, the two approaches worked as bias detectors present in the text as well as in the user features.

We believe that analyzing detection models is vital for understanding how certain features shape the way data is processed. Accuracy alone is by no means a sufficient metric to decide which model to prefer. Our work shows that even models that perform significantly better can potentially lead to new types of bias. We urge researchers in the field to compare recognition approaches beyond accuracy to avoid potential harm to affected users.

Data scarcity is still a main issue faced by current researchers, especially when it comes to context features. We believe that larger and more complete datasets will improve our understanding of how certain features interact and will help future research in advancing both in accuracy and bias mitigation.

Acknowledgments

This paper is based on a joined work in the context of Edoardo Mosca's master's thesis (Mosca, 2020).

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7).

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

Noé Cecillon, Vincent Labatut, Richard Dufour, and Georges Linarès. 2019. Abusive language detection in online conversations by combining content- and graph-based features. In *ICWSM International Workshop on Modeling and Mining Social-Media-Driven Complex Networks*, volume 2, page 8. Frontiers.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maeve Duggan. 2017. *Online harassment 2017*. Pew Research Center.

Elise Fehn Unsvåg and Björn Gambäck. 2018. The Effects of User Features on Twitter Hate Speech Detection. In *Proc. 2nd Workshop on Abusive Language Online*, pages 75–85.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proc. 11th ICWSM*, pages 491–500.

Patxi Galán-García, José Gaviria de la Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo García Bringas. 2016. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. *Logic Journal of the IGPL*, 24(1):42–53.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zachary C Lipton. 2018. The mythos of model interpretability. *Queue*, 16(3):31–57.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8).

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM conference on web science*, pages 173–182.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019a. Abusive language detection with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 2145–2150.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019b. Tackling online abuse: A survey of automated abuse detection methods. *arXiv preprint arXiv:1908.06024*.

Edoardo Mosca. 2020. Explainability of hate speech detection models. Master's thesis, Technical University of Munich. Advised and supervised by Maximilian Wich and Georg Groh.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. *Studies in Computational Intelligence*, 881 SCI:928–940.

Emily R Munro. 2011. The protection of children online: a brief scoping review to identify vulnerable groups. *Childhood Wellbeing Research Centre*.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 1135–1144.

Julian Risch, Robin Ruff, and Ralf Krestel. 2020. Offensive language detection explained. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 137–143.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proc. 5th Intl. Workshop on Natural Language Processing for Social Media*, pages 1–10.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of Innovative Applications of Artificial Intelligence (IAAI)*, pages 1058–1065.

Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Prashanth Vijayaraghavan, Hugo Larochelle, and Deb Roy. 2019. Interpretable Multi-Modal Hate Speech Detection. In *Intl. Conf. Machine Learning AI for Social Good Workshop*.

Cindy Wang. 2018. Interpreting neural network hate speech classifiers. In *Proc. 2nd Workshop on Abusive Language Online*, pages 86–92.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proc. First Workshop on NLP and Computational Social Science*, pages 138–142.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Maximilian Wich, Jan Bauer, and Georg Groh. 2020. Impact of politically biased data on hate speech classification. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 54–64.

Matthew L Williams, Pete Burnap, Amir Javed, Han Liu, and Sefa Ozalp. 2020. Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. *The British Journal of Criminology*, 60(1):93–117.

A Results on the Davidson Dataset

A.1 Feature Space learned by the Text Model

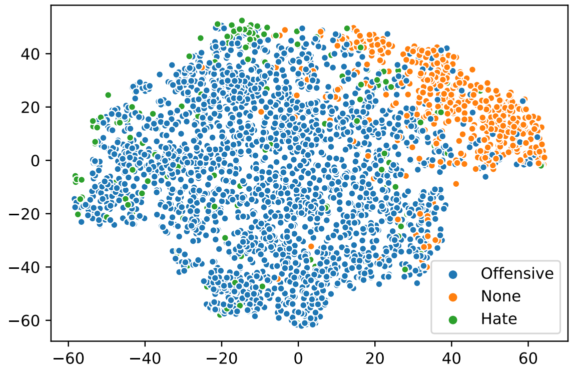


Figure 7: DAVIDSON tweets, colored by label, in the feature space learned by the text model.

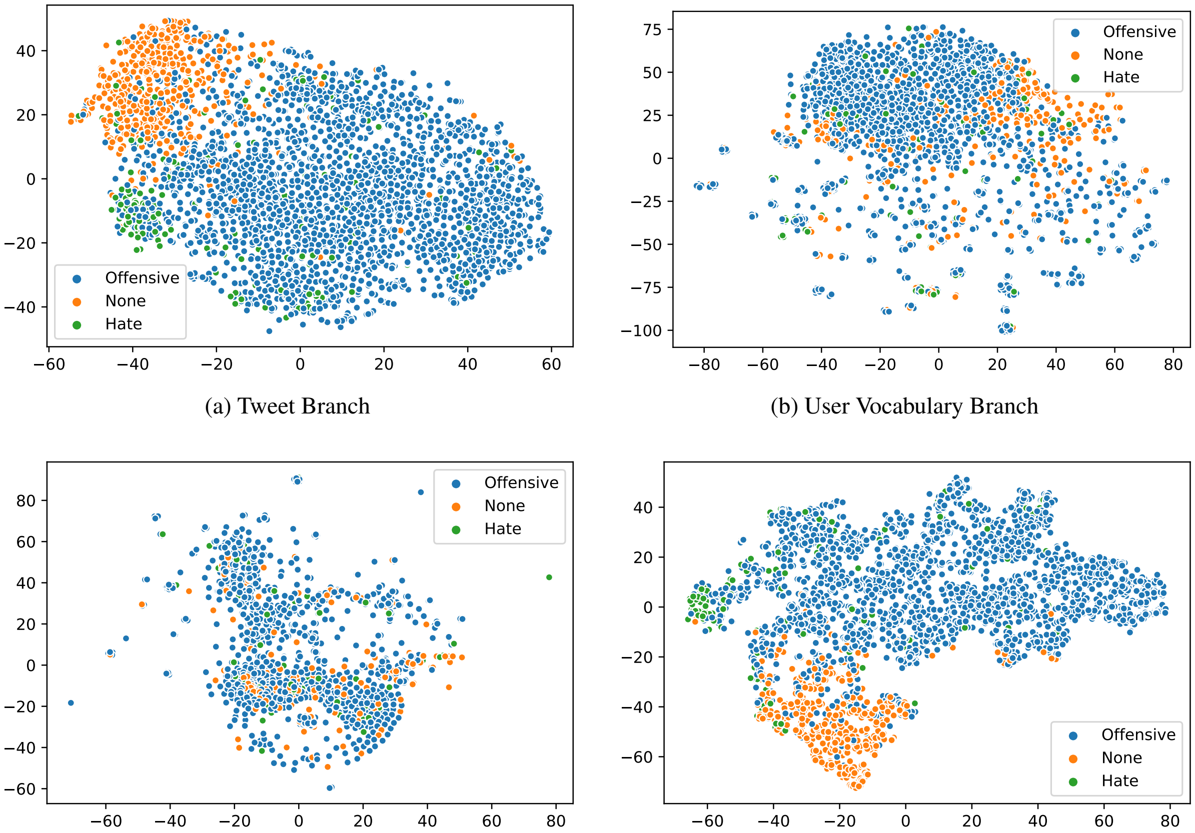
Figure 7 shows the feature space learned by our text model on DAVIDSON. Overall, the distribution looks similar as the one of WASEEM visualized in figure 4d. We can notice that hate tweets are extremely sparse and mixed with the offensive ones. This is reflected by the poor model performance on the hate class, possibly caused by the conceptual overlap that these two classes have. On the other hand, non-harmful tweets are mostly concentrated in one area of the plot, confirming the satisfactory F1 scored achieved.

A.2 Feature Space learned by the Social Model

Figure 8 shows the feature space learned by our social model on DAVIDSON. As done for WASEEM, we report the plots both for the single branches as well as for their combination. The tweet branch (figure 8a) has a similar structure to figure 7. However, hateful tweets are also concentrated in a small portion of the space. This reflects the improved performance that the social model had on the hate class. This suggests that the information coming from the other input sources reinforces the signal backpropagated to the tweet branch, resulting in a less chaotic mixture of hateful and offensive tweets. The user vocabulary (figure 8b) and the follower network branch (figure 8c) do not present the same characteristics as seen on WASEEM. In this case, we do not have the data points separated into multiple clusters. The same goes for the overall learned feature space (figure 8d), where all the tweets are contained in one single cloud. This is consistent with what we observed in terms of F1 Scores. In contrast to what occurred on WASEEM, user features did not cause a substantial impact on the feature space on DAVIDSON and thus did not produce a large leap in performance.

A.3 Complement to Figure 6

Figure 6 compares the model's behavior on the same tweet but with different authors, one racist and one regular. For completeness, figure 9 shows the corresponding plots—Shapley values and embedding onto the features space—for the same tweet when generated by a sexist user. The result is analogous to the one obtained with the regular user. Also in this case the tweet is not classified as racist (12% confidence). The estimated Shapley values show a substantial impact of the user vocabulary against the racism class. The embedding onto the latent space shows once more that changing the author caused the tweet to embed in a different cluster, hence excluding the possibility of the content being classified correctly.



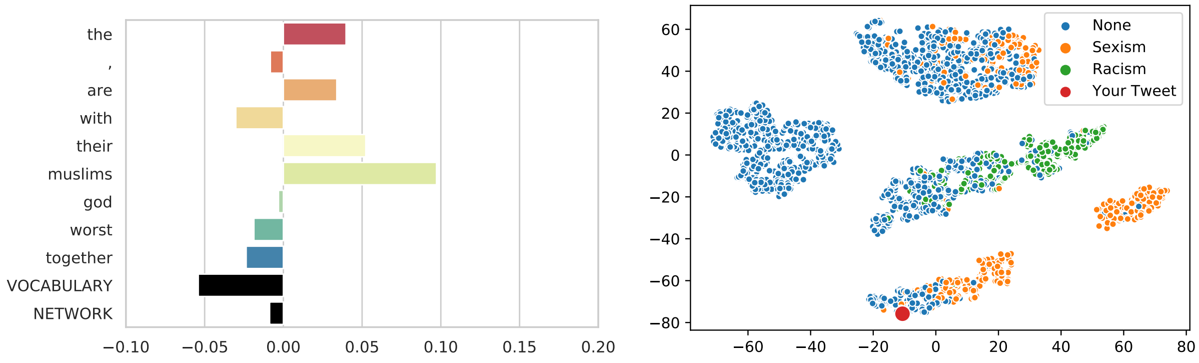
(a) Tweet Branch

(b) User Vocabulary Branch

(c) Follower Network Branch

(d) Last Hidden Layer

Figure 8: Latent space visualization of our social model on DAVIDSON, colored by label. The features are extracted from the single branches before the concatenation: tweet (a), user's vocabulary (b), follower network (c). The last plot (d) shows instead the final learned features space, after all branches are combined and processed together.



(a) Sexist User, Shapley Values

(b) Sexist User, Embedding in Latent Space

Figure 9: Features contribution (w.r.t. racism class) and embeddings of the islamophobic tweet in the social model's latent space. The pair of plots are w.r.t. the prediction done with sexist author.