Can Contextualizing User Embeddings Improve Sarcasm and Hate Speech Detection?

Kim Breitwieser
MaibornWolff GmbH
kim.breitwieser@maibornwolff.de

Abstract

While implicit embeddings so far have been mostly concerned with creating an overall representation of the user, we evaluate a different approach. By only considering content directed at a specific topic, we create sub-user embeddings, and measure their usefulness on the tasks of sarcasm and hate speech detection. In doing so, we show that task-related topics can have a noticeable effect on model performance, especially when dealing with intended expressions like sarcasm, but less so for hate speech, which is usually labelled as such on the receiving end.

1 Introduction

While using the syntax or semantics of sentences and words has been the backbone of Natural Language Processing (NLP) tasks for a long time, incorporating information about the authors themselves is a much more recent addition (Lucas et al., 2009; Zhang et al., 2018a; Li et al., 2017).

Existing approaches can be broadly grouped into explicit and implicit user modelling. Explicit representations include known information, such as the user's occupation or location (Mesgar et al., 2020), their personality or sociodemographic traits (Oraby et al., 2018), or even their emotional state (Rashkin et al., 2018). Implicit models, on the other hand, use highly dimensional vectors (embeddings) to capture abstract differences and similarities between users, without relying on concrete knowledge (Amir et al., 2017).

However, implicit approaches so far make use of an averaged user representation, for example by using a given user's post history regardless of content. Social psychology, however, has shown the impact a given social situation can have on observable behavior (Ross, 1977), a fact that could be possible to translate to social media posts as well. To this end, we define a *social situation* as a given topic, such as sports or politics, as well as the ensuing conversations about these topics. In doing so, we hope to improve the accuracy of sarcasm and hate speech detection, NLP tasks that will only increase in importance as the internet - and social media - continues to take up more of our time.

To this end, we use User2Vec, one of the earlier approaches to implicit user modelling, which has already been shown to increase performance on sarcasm classification (Amir et al., 2016).

1.1 Research Questions

- Can hate speech detection be improved by the usage of implicit user representation, in a similar way to sarcasm detection?
- Can these results be influenced by contextualizing the user on specific subsets of conversational data, implicitly modelling their behavior in different social situations?
- What are the implications behind the observed results, and how could they be made use of in future applications?

2 Related Work

2.1 User Embeddings in Social Media

Social media posts and other media can be used to infer a variety of user characteristics, such as demographics (Benton et al., 2016), mental health (Amir et al., 2017) or personality traits (Liu et al., 2016).

Purely text-based user embeddings are usually created using an unsupervised approach such as dimensionality reduction by Latent Dirichlet Allocation (LDA) (Schwartz et al., 2013; Song et al., 2015; Hu et al., 2017), Single Value Decomposition (SVD) (Kosinski et al., 2013; Gao et al., 2014) or by capturing contexts based on the Word2Vec family of word embeddings (Amir et al., 2016; Preoţiuc-Pietro et al., 2015). These approaches cluster the information contained in a given user's posts in order to discover patterns and similarities between

Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), pages 126 - 139

users. Aside from textual content, image content (Zhang et al., 2018b) and user relations, such as followers (Mishra et al., 2018), have been used in order to create user representations.

2.2 Sarcasm Detection

Sarcasm detection describes a classification problem, often binary in nature, though it can also be further differentiated in sarcasm as intended by the authors themselves, or perceived by external annotators (Shmueli et al., 2020a). Additionally, an alternative approach can be chosen in order to differentiate sarcasm from other expressions of humour or irony (Reyes et al., 2013).

Traditional approaches to detect sarcasm make use of explicit rules (Veale and Hao, 2010; Maynard and Greenwood, 2014; Riloff et al., 2013), as well as statistical measures (Hernández-Farías et al., 2015; Liu et al., 2014; Tsur et al., 2010), in order to differentiate sarcastic and non-sarcastic content. More recent, neural network-based approaches are able to implicitly construct complex, highly-dimensional feature representations from basic inputs, lessening the need for additional domain knowledge. These approaches often make use of RNN and CNN models (Ghosh and Veale, 2016), as well as Attention or Transformer architectures (Potamias et al., 2020). Some of them are also able to make use of auxiliary information, such as user embeddings (Amir et al., 2016; Hazarika et al., 2018).

2.3 Hate Speech Detection

Hate speech detection represents yet another classification problem, differentiating content expressing hate or encouraging violence, usually towards repressed minorities, from content without such tendencies. While hate speech can often be confused with the use of offensive phrases in everyday language, more recent approaches have made an effort to distinguish between these cases (Davidson et al., 2017; Warner and Hirschberg, 2012; Malmasi and Zampieri, 2017).

Similar to sarcasm detection, the classification of hateful content has also evolved from traditional methods, such as handcrafted rules (MacAvaney et al., 2019; Mondal et al., 2017), to employing neural network architectures such as RNNs and CNNs (Kovács et al., 2021). Especially in social media environments, emojis have recently been shown to provide a useful tool to resolve the ambiguity in words that can both be interpreted in a more neutral way or as part of hate speech (Wiegand and Ruppenhofer, 2021). Transformer models also have found their way into hate speech detection to great success, used either on their own or as part of more complex ensembles (Zampieri et al., 2019, 2020).

3 Datasets

3.1 Sarcasm

For the sarcasm classification task we use the Bamman dataset, based on the one used by Bamman and Smith (2015) and Amir et al. (2016). The dataset differentiates between sarcastic and non-sarcastic posts, using distant supervision based on the presence or absence of the hashtags *#sarcasm* or *#sarcastic*, which are removed prior to the actual task. The dataset contains a total of 8741 posts by 5797 users, divided into 4972 sarcastic (56.9%) and 3769 non-sarcastic (43.1%) posts.

3.2 Hate Speech

For the hate speech classification task we use the Hatexplain dataset (Mathew et al., 2020), more specifically a subset collected from the social media platform Gab. This split is done due to differing post lengths between Twitter and Gab, with the latter containing a larger volume of hate speech content (Zannettou et al., 2018). The dataset differentiates between neutral, offensive and hate speech, with posts being labelled independently by 3 annotators using Amazon Mechanical Turk[1]. The dataset consists of 8365 posts by 1642 users, divided into 1588 neutral (19.0%), 2487 offensive (29.7%) and 4290 hate speech (51.3%) posts.

4 Experiments

4.1 Model Architecture

For our experiments, we use a CUE-CNN (Content and User Embedding Convolutional Neural Network) architecture, directly derived from the one used in Amir et al. (2016). A more in-depth description of the model itself, as well as the hyperparameters used during the experiments, can be found in the appendix. While the model itself does produce comparable results to state-of-the-art architecture, such as BERT (Devlin et al., 2018), they are still comparable with each other, given that hyperparameters don't change, as well with the previous experiments performed by Amir et al. (2016).

[1] https://www.mturk.com

4.2 Run Variations

In order to inspect the influence of different kinds of user embeddings, we train and evaluate the model using multiple configurations:

- **only word embeddings**, serving as a baseline without additional user embeddings. We use 400-dimensional Word2Vec embeddings to represent words.
- **only user embeddings**, using only 400-dimensional user embeddings created from the 500 most recent posts per user, excluding the post being evaluated.
- **word + user embeddings**, serving as a secondary baseline, combining the inputs of both previous runs.
- **topic-specific sub-user embeddings**, similar to the previous run, but replacing the regular user embeddings with embeddings created from posts that are most likely to belong to one of 10 topics defined for the dataset. This run is repeated for every topic.

4.3 Sub-User embeddings

In order to model individual topics present in the user's post history, we create sub-embeddings based on an LDA model (Blei et al., 2001). By selecting, for each user, 500 posts most likely to belong to a topic, we assume these embeddings to represent the user's behavior in a situation depicting them talking about the given topic, known to influence behavior in a notable way (Ross, 1977; Giles and Baker, 2008). These embeddings are trained just like the averaged user embeddings, but by sorting the user history based on the percentage of each post to belong to the given topic, instead of by date.

An overview over the topics defined for both datasets can be found in the appendix. Labels are based on the 30 most salient terms per topic, as provided by the LDA model.

5 Results

5.1 Randomized Embeddings

Before testing the embeddings themselves, we perform initial runs in order to compare embeddings created using User2Vec with randomly created ones or those obtained by randomly reassigning either the post histories or resulting user embeddings. These could still provide minor improvements, as some users authored more than a single post, potentially even present in both training and validation sets. As the results obtained can be assumed to be universally representative, we only performed these runs on the sarcasm dataset.

	accuracy
only word embeddings	73.87
random user embeddings	73.63
shuffled posts	73.79
shuffled user embeddings	**74.02**

Table 1: Results on the sarcasm dataset using varying degrees of randomly created user embeddings compared to not using any additional data at all. The best performing run is highlighted.

As can be seen in Table 1, using purely random 400-dimensional vectors provides no improvement over not using them, even resulting in slightly worse results due to noise caused by the random values. Using actual user posts, but assigning them to random users results in slightly better results compared to purely random embeddings, though still worse than not using any additional information at all. Training user embeddings properly, with all posts of a given user being used for the same embedding, and assigning these to random users finally results in slightly better values compared to the baseline, though all of the observed results could reasonably as well be attributed to variance.

We can therefore conclude that the sheer presence of additional information, presented in multiple levels of randomness, does not provide a noticeable improvement over solely relying on the textual contents alone.

5.2 Sarcasm

Experimental results on the sarcasm dataset can be found in Table 2.

Surprisingly, even only using the user embeddings without the actual post contents results in a performance increase. Since the dataset has been labelled using marker hashtags, and therefore represents the author's intention to write sarcastic content, we believe that the model is able to accurately represent intended expressions, as has previously also been shown by Amir et al. (2016).

As for the topic-specific sub-user embeddings, all of the topics provided by the LDA model were

	accuracy	sarcasm
only word embeddings	73.87	-
only user embeddings	76.83	-
word + user embeddings	80.89	-
topic 1 (politics)	**82.19**	39.16
topic 2 (everyday)	81.47	73.42
topic 3 (time)	81.59	59.76
topic 4 (sports)	80.89	74.98
topic 5 (media)	81.55	55.46
topic 6 (social media)	**81.97**	50.98
topic 7 (celebratory)	81.49	45.03
topic 8 (offensive)	81.49	53.32
topic 9 (school)	**82.31**	49.14
topic 10 (emojis)	81.87	54.76

Table 2: Results over multiple runs performed on the sarcasm dataset, averaged over 10-fold cross-validation. Highlighted cells mark the 3 highest scoring runs. The second column shows the percentage of topic-related posts labelled as sarcastic.

able to produce results that are significantly better than when using only the most recent 500 posts per user. While this could theoretically be attributed to LDA preferably selecting posts with a higher amount of tokens, this was not the case. The most likely conclusion in this case is therefore that the topic-specific embeddings implicitly filter out stopwords and other fillers, as well as putting emphasis on words that carry meaning in the topic context at hand.

Among these topics, *politics*, *social media*, and *school* produce slightly better results, which can be proven to be statistically significant using the Wilcoxon-Test (Wilcoxon, 1945). Sarcasm is especially prevalent in online conversations (Hancock, 2004), and has been shown to positively correlate with social media reactions such as likes and retweets (Peng et al., 2019). Using Pearson's *r*, we can obtain a correlation of -0.7305 (p = 0.0165) between the obtained accuracy scores and the percentage of topic-related posts labelled as sarcastic. We can therefore conclude that certain topics are indeed more or less likely to harbour sarcastic remarks, with those containing a lesser degree of sarcasm being moderately more useful in detecting outliers.

5.3 Hate Speech

Experimental results on the hate speech dataset can be found in Table 3.

	accuracy	hate sp.
only word embeddings	62.28	-
only user embeddings	54.49	-
word + user embeddings	**63.47**	-
topic 1 (everyday)	62.80	49.10
topic 2 (jews)	62.62	37.29
topic 3 (gun control)	62.49	39.92
topic 4 (social media)	62.82	46.06
topic 5 (election)	62.26	43.52
topic 6 (religion)	62.69	43.85
topic 7 (terrorism)	62.59	56.08
topic 8 (racism/sexism)	**63.12**	50.94
topic 9 (australia)	62.79	52.12
topic 10 (foreign politics)	**62.88**	57.17

Table 3: Results over multiple runs performed on the hate speech dataset, averaged over 10-fold cross-validation. Highlighted cells mark the 3 highest scoring runs. The second column shows the percentage of topic-related posts labelled as hate speech.

Here, we can only observe minor absolute when using user embeddings in addition to the posts themselves, though they are still high enough to be deemed statistically significant. For this dataset, the user embeddings alone also perform worse on their own, which can be attributed to hate speech generally being considered to be a perceived phenomenon, especially since the dataset in question has been labelled by external annotators, and not by the authors themselves. Perceived phenomena like this have been shown to be impacted by user embeddings to a lesser extent, due to a potential discrepancy between assigned labels and the author's original intentions (Roussos and Dovidio, 2018; Oprea and Magdy, 2019).

Using any form of topic-based sub-user embeddings turned out to slightly lower absolute results, though some of them are still seen as minor improvements when evaluating individual posts. Given the general nature of the Gab social platform and its tendency to mainly harbour less moderated conversations regarding political topics (Zannettou et al., 2018), it can be assumed that all of these topics are subject to hate speech in some form. In order to prove this, we can again use Pearson's *r*, arriving at a correlation of 0.4873 (p = 0.1531). Based

on this, we conclude that the individual topics are not expressive enough, which causes averaged embeddings to be able to better capture indicators pointing towards hateful content. In comparison, posts belonging to a single topic are more focused towards it, which represents noise in the scope of our classification task.

Though the results obtained using topic-based sub-user embeddings are generally close to each other and the averaged baseline, the topics *racism/sexism* and *politics/foreign* seem to be slightly more useful in detecting hate speech than other topics. Since the dataset was created using gender- and race-related hate speech targets, this seems intuitive (Mathew et al., 2020), in addition to foreign ethnicities generally being a regular target of hate speech (Silva et al., 2016).

6 Conclusion and Future Work

We showed that user embeddings created using User2Vec (Amir et al., 2017) provide helpful information, capable of aiding the classification of intended expressions that can be observed as a general tendency for a given user, such as sarcasm. On the other hand, their usefulness is generally lower when used for the classification of perceived expressions, such as hate speech. Additionally, we were able to create specialized sub-user embeddings, capturing information about the user when exposed to a specific situation, such as when talking about school-related topics. Depending on how these topics relate to the task, we were able to increase the performance as opposed to using generalized embeddings. This seems to be especially true for binary classification tasks, which can be noticeably impacted by selecting a topic known to lean heavily towards one of the labels, which we have shown to be true for politics-related content being particularly low on sarcasm. We used a relatively simple LDA model to categorize posts by topic, but more sophisticated approaches should be able to even more properly select relevant data points.

Aside from the individual user, social connections can play a big role, potentially elevating the usefulness of user embeddings beyond the detection of characteristical, intended behavior. This information can be used in order to model user reactions, therefore providing insight about how a given user is perceived by others. Doing so could overcome one of the limitation of user embeddings, making it possible to more accurately detect perceived behavior such as hate speech (Roussos and Dovidio, 2018; Oprea and Magdy, 2019). And even the word embeddings, which we chose to leave fixed for all experiments, can be contextualized, as the information contained in a word can differ depending on the user who authored it (Welch et al., 2020).

Lastly, by assigning topic probabilities to individual posts, these could be used for the filtering of social media streams based on personal interest. Extending this approach to the author themselves, and by assuming that proficiency in a given topic can be approximated by having authored a high number of related posts (Ericsson et al., 1993; Ericsson, 2002), it could be possible to filter users based on the topics they are knowledgeable about.

7 Ethical and Privacy Considerations

Given that both the tasks of sarcasm and hate speech classification, as well as the models proposed in order to tackle it, aim at the labelling of users and their authored content, as well as a possible future application extending to a form of user rating or filtering based on their assumed proficiency, there are certain ethical implications. These do not only exist for correctly made statements, but also for potential misclassifications (Rudman and Glick, 2012). We therefore strongly advise to not use the proposed models as the sole basis for decisions made concerning the fate of humans, such as to which candidates to pick given a certain position, if an assumed level of proficiency would ever be used to make such a decision.

As for the topic of privacy, all data used by us in the creation of our models, as well as subsequent evaluations, are publicly available on the Twitter and Gab social media platforms. It should be noted that the Developer Agreements of these platforms forbid the usage of their data for the purpose of surveillance or in order to perform discriminatory actions, as exemplary outlined in Pardo et al. (2013). We therefore explicitly state that the scope of this work is strictly limited to the evaluation of models based on publicly available data in order to approach the problem of topic-based sub-user embeddings and their influence on sarcasm and hate speech classification, and not used to discriminate or surveil individual users based on the information obtained.

8 Acknowledgments

The experiments performed as part of this paper were conducted on infrastructure and under research mentoring provided by the Conversational AI and Social Analytics (CAISA) Lab[2], without which it might not have been possible to finish this work. Additional thanks go to Lucie Flek and Martin Potthast for their constructive feedback.

References

Silvio Amir, Glen Coppersmith, Paula Carvalho, Mário J. Silva, and Byron C. Wallace. 2017. Quantifying mental health from social media with neural user embeddings. *CoRR*, abs/1705.00335.

Silvio Amir, Byron C. Wallace, Hao Lyu, and Paula Carvalho Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media.

David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. In *ICWSM*.

Adrian Benton, Raman Arora, and Mark Dredze. 2016. Learning multiview embeddings of Twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Berlin, Germany. Association for Computational Linguistics.

David Blei, Andrew Ng, and Michael Jordan. 2001. Latent dirichlet allocation. volume 3, pages 601–608.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Karl Ericsson. 2002. Attaining excellence through deliberate practice: Insights from the study of expert performance. *The Pursuit of Excellence in Education*, pages 21–55.

Karl Ericsson, Ralf Krampe, and Clemens Tesch-Roemer. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100:363–406.

Huiji Gao, J. Mahmud, J. Chen, Jeffrey Nichols, and M. Zhou. 2014. Modeling user attitude toward controversial topics in online social media. In *ICWSM*.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Howard Giles and Susan C Baker. 2008. Communication accommodation theory. *The international encyclopedia of communication*.

Fréderic Godin. 2019. *Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing*. Ph.D. thesis, Ghent University, Belgium.

Jeffrey T. Hancock. 2004. Verbal irony use in face-to-face and computer-mediated conversations. *Journal of Language and Social Psychology*, 23(4):447–463.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. Cascade: Contextual sarcasm detection in online discussion forums.

Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. 2015. Applying basic features from sentiment analysis for automatic irony detection. In *Pattern Recognition and Image Analysis*, pages 337–344, Cham. Springer International Publishing.

Tianran Hu, Haoyuan Xiao, Thuy vy Thi Nguyen, and Jiebo Luo. 2017. What the language you tweet says about your occupation.

Michal Kosinski, David Stillwell, and Thore Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the national academy of sciences*, 110(15):5802–5805.

György Kovács, Pedro Alonso, and Rajkumar Saini. 2021. Challenges of hate speech detection in social media. *SN Computer Science*, 2(2):1–15.

Justin Kruger, Nicholas Epley, Jason Parker, and Zhi-Wen Ng. 2006. Egocentrism over e-mail: Can we communicate as well as we think? *Journal of personality and social psychology*, 89:925–36.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Data distillation for controlling specificity in dialogue generation. *CoRR*, abs/1702.06703.

Leqi Liu, Daniel Preotiuc-Pietro, Zahra Riahi Samani, Mohsen E Moghaddam, and Lyle Ungar. 2016. Analyzing personality through social media profile picture choice. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10.

[2] https://caisa-lab.github.io

Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*, pages 459–471. Springer.

J.M. Lucas, Fernando Fernández-Martínez, J. Salazar, Javier Ferreiros, and Rubén San-Segundo. 2009. Managing speaker identity and user profiles in a spoken dialogue system. *Procesamiento del lenguaje natural, ISSN 1135-5948, Nº. 43, 2009, pags. 77-84*, 43.

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):e0221152.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 467–472, Varna, Bulgaria. INCOMA Ltd.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4238–4243, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mohsen Mesgar, Edwin Simpson, Yue Wang, and Iryna Gurevych. 2020. Generating persona-consistent dialogue responses using deep reinforcement learning.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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.

Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th acm conference on hypertext and social media*, pages 85–94.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2859, Florence, Italy. Association for Computational Linguistics.

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T. S., Stephanie M. Lukin, and Marilyn A. Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. *CoRR*, abs/1805.08352.

F. M. R. Pardo, P. Rosso, Irina Chugur, Martin Potthast, Martin Trenkmann, Benno Stein, B. Verhoeven, and W. Daelemans. 2013. Overview of the author profiling task at pan 2013. In *CLEF*.

Wei Peng, Achini Adikari, Damminda Alahakoon, and John Gero. 2019. Discovering the influence of sarcasm in social media responses. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9.

Rolandos Potamias, Georgios Siolas, and Andreas Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32.

Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015. An analysis of the user occupational class through Twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764, Beijing, China. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. I know the feeling: Learning to converse with empathy. *CoRR*, abs/1811.00207.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

Ellen Riloff, A. Qadir, Pallavi Surve, L. Silva, N. Gilbert, and R. Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. *Proceedings of EMNLP*, pages 704–714.

Lee Ross. 1977. The intuitive psychologist and his shortcomings: Distortions in the attribution process. volume 10 of *Advances in Experimental Social Psychology*, pages 173–220. Academic Press.

Gina Roussos and John F. Dovidio. 2018. Hate speech is in the eye of the beholder: The influence of racial attitudes and freedom of speech beliefs on perceptions of racially motivated threats of violence. *Social Psychological and Personality Science*, 9(2):176–185.

L.A. Rudman and P. Glick. 2012. *The Social Psychology of Gender: How Power and Intimacy Shape Gender Relations*. Texts in Social Psychology. Guilford Publications.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social

media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Boaz Shmueli, Lun-Wei Ku, and Soumya Ray. 2020a. Reactive supervision: A new method for collecting sarcasm data.

Boaz Shmueli, Lun-Wei Ku, and Soumya Ray. 2020b. Reactive Supervision: A New Method for Collecting Sarcasm Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2553–2559, Online. Association for Computational Linguistics.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10.

Xuemeng Song, Liqiang Nie, Luming Zhang, Maofu Liu, and Tat-Seng Chua. 2015. Interest inference via structure-constrained multi-source multi-task learning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. pages 765–770.

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, Montréal, Canada. Association for Computational Linguistics.

Charlie Welch, Jonathan Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Exploring the value of personalized word embeddings. pages 6856–6862.

Michael Wiegand and Josef Ruppenhofer. 2021. Exploiting emojis for abusive language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 369–380, Online. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Savvas Zannettou, Barry Bradlyn, Emiliano De Cristofaro, Haewoon Kwak, Michael Sirivianos, Gianluca Stringini, and Jeremy Blackburn. 2018. What is gab: A bastion of free speech or an alt-right echo chamber. pages 1007–1014.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Wei Zhang, Wen Wang, Jun Wang, and Hongyuan Zha. 2018b. User-guided hierarchical attention network for multi-modal social image popularity prediction. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1277–1286, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

A Word embeddings

For our experiments we made use of 400-dimensional Word2Vec embeddings created from a Twitter-based corpus (Godin, 2019), both to create the user and sub-user embeddings, as well as creating the inputs to the model itself. The dataset consists of 3039345 tokens with an OOV (out-of-vocabulary) rate of 9.0% on the sarcasm dataset, as well as 13.5% on the hate speech dataset, both significantly lower than other, widely used word embeddings. This is because the vocabulary contains several emoji as well as other vocabulary specifically suited to use on social media data.

Table 4 shows a comparison between the datasets we previously selected as candidates for use.

B User2Vec

User2Vec aims to create implicit user representations based on the author's posting history, maximizing the probability of a given sentence, defined by each individual word in that sentence, to belong to that user:

$$P(S|user_j) = \sum_{w_i \in S} \log P(w_i|\mathbf{u_j}) + \sum_{w_i \in S} \sum_{w_k \in C(w_i)} \log P(w_i|\mathbf{e_k}) \quad (1)$$

	vocabulary	dimensions		OOV (%)
Twitter GloVe	1193514	200	sarcasm	16.2
			hate speech	20.2
GoogleNews Word2Vec	3000000	300	sarcasm	23.4
			hate speech	25.3
Twitter Word2Vec	3039345	400	sarcasm	**9.0**
			hate speech	**13.5**

Table 4: OOV words for a set of pre-selected candidate word embeddings, for both of the datasets used during the experiments.

Here, $S = \{w_0, w_1, ..., w_N\}$ represents a sentence authored by $user_j$. The probability itself can be decomposed into 2 formulas, the first one being conditional on the user representation $\mathbf{u_j}$, the second one being conditional on a window C of predefined size around the embedding $\mathbf{e_k}$ of any given word in the sentence. Since the latter probability is independent from the user itself, it represents a static value over all users and does not need to be considered in the model:

$$P(S|user_j) \propto \sum_{w_i \in S} \log P(w_i|\mathbf{u_j}) \quad (2)$$

The resulting approximation is very similar to Paragraph Vectors, a variation on Word2Vec (Mikolov et al., 2013), creating embeddings for paragraphs and documents instead of individual words, when considering each user as its own document consisting of the content authored by that user (Le and Mikolov, 2014). Using a log-linear model, the probability for each word $P(w_i|\mathbf{u_j})$ can be estimated using Softmax as follows:

$$P(w_i|x) = \frac{\exp(\mathbf{W_i} \cdot x + b_i)}{\sum_k \exp(\mathbf{W_k} \cdot x + b_k)} \quad (3)$$

Where $\mathbf{W}$ and b represent the weights and biases of said model and x is the feature vector being optimized in order to represent the user. The downside of this approach is the necessity to iterate over each word in the vocabulary, which is potentially very expensive. In order to reduce the cost of this operation, negative sampling is utilized in order to minimize the following Hinge-Loss:

$$\mathcal{L}(w_i, user_j) = \sum_{w_l \in V} \max(0, 1 - \mathbf{e_i} \cdot \mathbf{u_j} + \mathbf{e_l} \cdot \mathbf{u_j}) \quad (4)$$

We chose The following hyperparameters, adapted from the values published as part of the User2Vec model:

- 15 negative samples per word.
- Maximum vocabulary size of 50000, though this limitation was never reached in practice
- Only consider words with a minimum frequency of 5 across the input corpus.
- Initial learn rate of 5e-5, decaying over time as learn progress slows down.
- 25 maximum epochs, aborting the training process after not observing progress after 5 epochs (patience).

C Model Architecture (CUE-CNN)

Similar to how images are represented as a 2-dimensional arrangement of pixels, sentences can be seen as a list of words, each of which by itself is represented by a highly-dimensional vector, also resulting in a 2-dimensional matrix of scalar values. CNNs can make use of this structure in order to incorporate local spatial information and not only process individual words, but also their relation to neighbouring words. This allows them to interpret the overall sentence structure, as well as the relation between individual dimensions of the word embeddings.

The CUE-CNN (Content and User Embedding Convolutional Neural Network) model, as shown on figure 1, combines these embedded sentences $\mathbf{S}$ with pretrained user embeddings $\mathbf{U}$ to incorporate both the text contents themselves, as well as information about their authors. By doing so, it takes into account the user's post history and usage of words in relation to other users in the same vector space. As the user embeddings have previously

been created based on the same word embeddings that are also used in order to encode the posts themselves, this represents a connection between the sentence currently being classified, as well as other sentences authored by the same user in the past (Amir et al., 2016).

The embedded sentences are first fed to a convolutional layer consisting of 3 filters of different sizes in order to capture spatial relations in different granularities. Each filter $\mathbf{F}$ gets combined with sub-matrices of a sentence using a sliding window approach, with the results being subjected to a non-linear ReLU activation function α in order to create feature maps $\mathbf{m_i}$ of the same size as the filter:

$$\mathbf{m_i} = \alpha\left(\mathbf{F} \cdot \mathbf{S}_{[i:i-h+1]} + b\right) \tag{5}$$

These filter maps are fed to a max pooling layer being applied to the maximum length of sentences present in the dataset, in order to transform them to scalar values:

$$\mathbf{f_k} = [\max(\mathbf{m_1}) \oplus ... \oplus \max(\mathbf{m_M})] \tag{6}$$

These values are then concatenated over all 3 filters as well as the pretrained embedding of the sentence's author $\mathbf{U_u}$, obtaining the representation of the full model input $\mathbf{c}$. Alternatively, the user embeddings can be left empty, measuring the model's base performance when only processing the text contents.

$$\mathbf{c} = [\mathbf{f_1} \oplus \mathbf{f_2} \oplus \mathbf{f_3} \oplus \mathbf{U_u}] \tag{7}$$

The combined values are then passed to another ReLU activation function α, as well as being subjected to dropout, randomly setting a certain fraction of input nodes to zero in order to prevent the model from overfitting on the training data. A final dense layer then reduces the vector to the previously defined number of possible output classes. The entire model can therefore be formulated as:

$$P(y = k|s, u; \theta) \propto \mathbf{Y_k} \cdot \alpha(\mathbf{H} \cdot \mathbf{c} + \mathbf{h}) + \mathbf{b_k} \tag{8}$$

with $\theta = \{\mathbf{Y}, \mathbf{b}, \mathbf{H}, \mathbf{h}, \mathbf{F_1}, \mathbf{F_2}, \mathbf{F_3}, \mathbf{E}, \mathbf{U}\}$ consisting of - in order - the weights and biases of the output and hidden layers, the convolutional filters being applied to the input sentences, the pretrained word embeddings used in order to represent these sentences in matrix form, and pretrained user embeddings based on the same word embeddings (Amir et al., 2016).

Both word and user embeddings are frozen and not updated during training, and the same hyperparameters are used for all classification tasks, being as follows:

- 80/10/10 training/validation/test split, created from 10 identically sized folds and evaluated using cross-validation. For the final scores, fold results are summed and averaged.
- 50 epochs using a batch size of 32, without early stopping or checkpointing.
- Categorical Cross-Entropy Loss independent of the number of classes, so the model generalizes beyond binary classification without the need for change.
- Adadelta optimization, using a learn rate of 0.005, 0.95 momentum and weight decay of 0.001.
- 3 CNN filters, sized at 4, 6 and 8, respectively.
- 200 filters maps as CNN layer output.
- Hidden layer size of 100.
- Dropout probability of 0.15 between the hidden and output layer.

Training, validation and test sets are created in a stratified fashion based on their ground truth labels, making sure the label distribution in all folds is representative for the whole dataset.

D Preprocessing

Prior to the experiments, we filtered the datasets to only include users with at least 1000 authored historical posts, to ensure there is enough data to properly evaluate different conditions on each user in the dataset. Since the datasets each provide user ids for the Twitter and Gab platform, respectively, we used these to obtain the post history for each user. For Twitter, this has been done using the API, which is limited to the most recent 3000 posts per user[3]. For Gab, we used a publicly available dataset[4]. We further preprocess each example using the following pipeline:

[3] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/timelines/api-reference/get-statuses-user_timeline

[4] https://files.pushshift.io/gab/

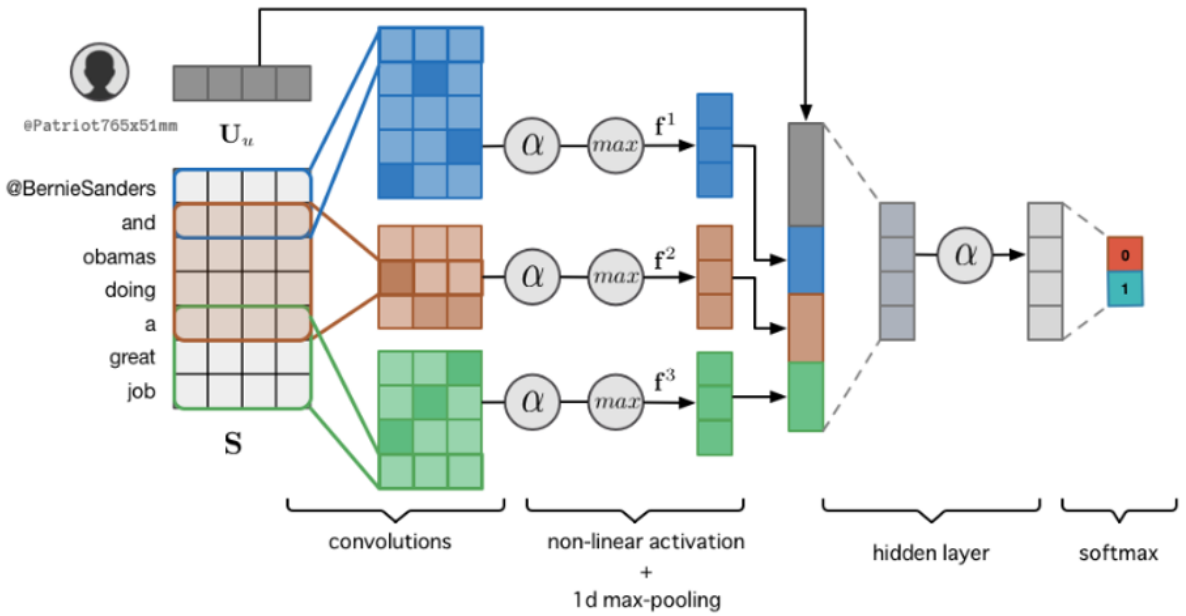

Figure 1: CUE-CNN (Content and User Embedding Convolutional Neural Network) model used for the classification tasks.

- Squashing all whitespace characters to single spaces. Social media content in particular can often contain repeated spaces or newlines, as well as possible non-standard whitespace characters not part of the pretrained embeddings and therefore removed during tokenization.
- Converting all text to lowercase, reducing the amount of OOV (out-of-vocabulary) tokens and increasing the information that can be obtained from each datapoint.
- Reducing repeated characters to a maximum of 3. Social media content is prone to expressions like *"wowwwwww!"* or *"riiiight"*. While still unlikely in most cases, this increases the chances to find an embedding for tokens like these.
- Replacing all user mentions with *@user* and hyperlinks with *url*. Individual user mentions and especially web URLs are unlikely to be present in the embeddings, but their positioning and frequency in the text can still be useful for the task.
- Special tokenization for smileys, which usually consist of mostly punctuation characters. They are an important tool to convey emotions in social media environments (Kruger et al., 2006), so special care is taken in order to make sure they are left intact.

E Additional experiments

In addition to the datasets described in the main paper, we also ran the same experiments on the SPIRS dataset, another sarcasm dataset, which additionally contains labels for intended and perceived sarcasm (Shmueli et al., 2020b). This allows us to more accurately describe the impact of our method on these criteria, while leaving the general task the same.

It should be noted that, while the dataset also provides additional information in the form of cue, oblivious, and eliciting tweets, these were not used for our experiments.

Experimental results on this dataset can be found in Table 5.

As with the Bamman dataset, we can see that user embeddings noticeably improve performance on the dataset. More importantly, though, Table 6 shows the change of misclassification rate for the intended and perceived parts of the dataset, when using user embeddings in addition to the word embedding baseline. While we can observe an improvement in both cases, it's more noticeable on intended sarcasm, whose misclassification rate reduced by 54.24%, while the error on the perceived part only reduced by 44.48%. This observation seems to prove our assumption that user embeddings, at least those solely created from the user's post history, are more helpful for the classification

	accuracy	sarcasm
only word embeddings	67.44	-
only user embeddings	**87.03**	-
word + user embeddings	83.73	-
topic 1 (covid-19)	83.88	49.89
topic 2 (sports)	81.36	43.44
topic 3 (politics)	83.59	48.92
topic 4 (race & gender)	**84.39**	49.87
topic 5 (offensive)	83.18	37.99
topic 6 (media)	82.71	46.19
topic 7 (numbers)	82.68	59.99
topic 8 (love)	81.32	48.05
topic 9 (slang)	**84.31**	53.72
topic 10 (happiness)	81.57	47.06

Table 5: Results over multiple runs performed on the SPIRS dataset, averaged over 10-fold cross-validation. Highlighted cells mark the 3 highest scoring runs. The second column shows the percentage of topic-related posts labelled as sarcastic.

of intended expressions.

	run	error (%)
intended	word embeddings	27.58
	word + user emb.	12.62
perceived	word embeddings	39.93
	word + user emb.	22.17

Table 6: Distribution of misclassfications on the SPIRS dataset between intended and perceived sarcasm. When adding user embeddings, the overall error decreases, while the relative error on examples labeled as perceived sarcasm increases.

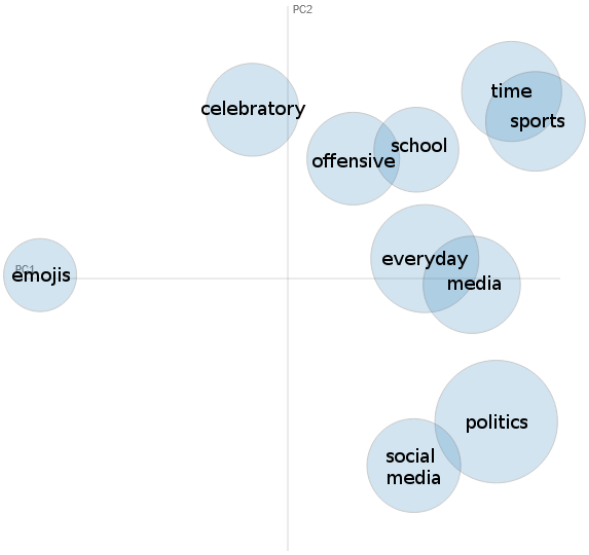

Figure 2: Distribution and partial overlap of LDA topics created from the sarcasm dataset's post history.

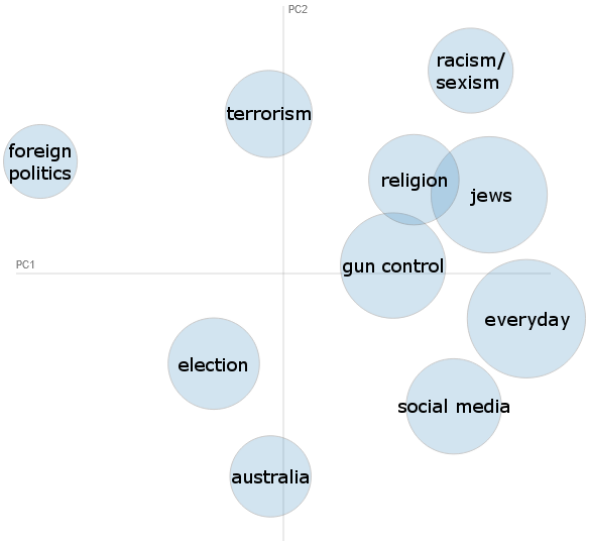

Figure 3: Distribution and partial overlap of LDA topics created from the hate speech dataset's post history.

F Comparing topic-specific sub-user embeddings

In order to extract the topics used as a basis for our sub-user embeddings, we create an LDA model from each dataset's entire post history. The model is created using a single pass over the data, ignoring all tokens appearing either only once or in more than 99% of posts.

Figure 2 and 3 visualize 10 topics created from the sarcasm and hate speech dataset's respective post history in 2D space, across all users. Each of these topics represents a subspace of the overall text corpus, sometimes with partial overlap indicating a regular overlap between contents. Though, as this visualization represents a major dimensionality reduction from the original 400-dimensional vector space, not all relations between topics can be observed this way. We inferred the topic labels by taking into account the 30 most salient terms per topic, as presented by the underlying LDA model.

In order to compare the performance between these topics, we used the Wilcoxon signed-rank test (Wilcoxon, 1945), after using - for each author - the 500 posts with the highest probability of belonging to a given topic as input into our model. We performed tests on all possible topic pairs, with the final results being listed in the tables below, as well as being referenced in the main paper.

	politics	everyday	time	sports	media	social m.	celebratory	offensive	school	emojis
baseline	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
politics	-	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.9484
everyday	0.0000	-	0.0746	1.0000	0.9698	1.0000	0.9922	0.1200	0.0000	0.0000
time	0.0000	0.9254	-	1.0000	0.9977	1.0000	1.0000	0.8633	0.0000	0.0027
sports	0.0000	0.0000	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
media	0.0000	0.0302	0.0023	1.0000	-	0.9673	0.7412	0.0012	0.0000	0.0000
social m.	0.0000	0.0000	0.0000	1.0000	0.0327	-	0.4334	0.0001	0.0000	0.0000
celebratory	0.0000	0.0078	0.0000	1.0000	0.2588	0.5666	-	0.0001	0.0000	0.0000
offensive	0.0000	0.8800	0.1367	1.0000	0.9988	0.9999	0.9999	-	0.0000	0.0022
school	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	-	1.0000
emojis	0.0516	1.0000	0.9973	1.0000	1.0000	1.0000	1.0000	0.9978	0.0000	-

Table 7: Comparison between different topic-specific sub-user embeddings on the sarcasm dataset using the Wilcoxon signed-rank test. Highlighted table cells signify that the *column* run is achieving a significantly lower loss than the *row* run, over all test examples ($\alpha = 0.05$). The baseline consists of averaged embeddings created from the most recent 500 posts/user, unrelated to any specific topic.

	everyday	jews	gun control	social m.	election	religion	terrorism	racism/ sexism	australia	foreign p.
baseline	0.0419	0.3017	0.9896	0.7707	0.9881	0.0030	0.0063	0.0007	0.0004	0.0000
everyday	-	0.9682	0.9997	0.9852	1.0000	0.1843	0.3166	0.1323	0.0175	0.0470
jews	0.0318	-	0.9896	0.8436	0.9997	0.0011	0.0717	0.0003	0.0000	0.0006
gun control	0.0003	0.0104	-	0.2422	0.5258	0.0000	0.0000	0.0000	0.0000	0.0000
social m.	0.0148	0.1564	0.7578	-	0.6677	0.0000	0.0005	0.0001	0.0000	0.0000
election	0.0000	0.0003	0.4742	0.3323	-	0.0000	0.0000	0.0000	0.0000	0.0000
religion	0.8157	0.9989	1.0000	1.0000	1.0000	-	0.8107	0.4108	0.1589	0.2117
terrorism	0.6834	0.9283	1.0000	0.9995	1.0000	0.1893	-	0.0472	0.0101	0.0278
racism/ sexism	0.8677	0.9997	1.0000	0.9999	1.0000	0.5892	0.9528	-	0.7103	0.1534
australia	0.9825	1.0000	1.0000	1.0000	1.0000	0.8411	0.9899	0.2897	-	0.2128
foreign p.	0.9530	0.9994	1.0000	1.0000	1.0000	0.7883	0.9722	0.8466	0.7872	-

Table 8: Comparison between different topic-specific sub-user embeddings on the hate speech dataset using the Wilcoxon signed-rank test. Highlighted table cells signify that the *column* run is achieving a significantly lower loss than the *row* run, over all test examples ($\alpha = 0.05$). The baseline consists of averaged embeddings created from the most recent 500 posts/user, unrelated to any specific topic.