NLPOP: a Dataset for Popularity Prediction of Promoted NLP Research on Twitter

Leo Obadić, Martin Tutek, Jan Šnajder
Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
obadic.leo@gmail.com, {martin.tutek,jan.snajder}@fer.hr

Abstract

Twitter has slowly but surely established itself as a forum for disseminating, analysing and promoting NLP research. The trend of researchers promoting work not yet peer-reviewed (*preprints*) by posting concise summaries presented itself as an opportunity to collect and combine multiple modalities of data. In scope of this paper, we (1) construct a dataset of Twitter threads in which researchers promote NLP preprints and (2) evaluate whether it is possible to predict the popularity of a thread based on the content of the Twitter thread, paper content and user metadata. We experimentally show that it is possible to predict popularity of threads promoting research based on their content, and that predictive performance depends on modelling textual input, indicating that the dataset could present value for related areas of NLP research such as citation recommendation and abstractive summarization.

1 Introduction

The now not-so-recent neural revolution caused a widespread increase of interest in machine learning research. Through improvements obtained across the field by applying deep neural networks, every application of machine learning became open for researchers to publish work pushing pre-neural boundaries, whether that work applied a neural architecture to a problem (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014), unveiled the black box of deep neural networks (Simonyan et al., 2014; Li et al., 2016) or coming up with a new architecture altogether (He et al., 2016; Vaswani et al., 2017). The rapid progress paved way for more researchers to enter the field, which resulted in an ever increasing volume of research work published year by year.

The large volume of work meant that it is difficult for a single person to keep up to date with relevant research. Thus, a need emerged for a platform where work can be shared, filtered and discussed on a scale larger than research labs. Twitter, a microblogging social network emerged as the chosen forum. The otherwise prohibitive 280 character limit on each post ("tweet") can in this context be viewed as a feature – it promotes succinctness and discourages lengthy academic prose. A portion of researchers accepted that promoting your academic work on Twitter is something that you do – and if done well, it is believed that your research pedigree and citation count will increase. While this statement has not yet been put to test, the increase of posts promoting research work indicates that many believe it (Figure 1).

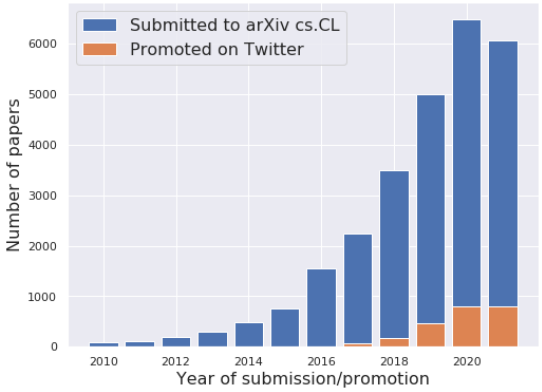

Figure 1: Distribution of the number of preprints published on arXiv under computational linguistics (cs.CL) and preprints promoted on Twitter as per the data in our dataset. Note that statistics for 2021 are incomplete.

Along with sharing a link to your paper, it is common to provide a concise summary outlining the main idea and contributions of your work in form of a post thread. In scope of this paper we aim to collect a dataset of Twitter threads promoting research work and evaluate whether the popularity of a post can be determined from the content of the thread, paper, or user information. We would like to emphasize that we do not believe that scientific work being popular implies that the work itself is good, but rather aim to analyse whether it is possi-

Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, pages 286 - 292

ble to determine factors which lead to higher visibility. Researchers could then use findings from such analysis to hopefully reach a broader audience.

2 Related Work

Predicting popularity of messages is a straightforward task from the perspective of machine learning and has been framed both as a regression (Lampos et al., 2014) and classification problem (Hong et al., 2011; Jenders et al., 2013; Subramanian et al., 2018; Fiok et al., 2020), while work on information cascades (Zhao et al., 2015; Li et al., 2017; Zhou et al., 2021) focuses on modeling the entire lifetime of a post as a point process.

Work in the area of computational linguistics mainly focuses on analysing the underlying causes of popularity: Tan et al. (2014) evaluate whether wording affects popularity of posts and find a number of patterns which popular posts adhere to, Jaech et al. (2015) analyse how use of language gets people involved in online discussions, while approaches such as Karimi et al. (2016) and Zarezade et al. (2017) aim to help users reach a larger audience. Only recently has the effect of social media on collaboration between researchers been analysed (Gorska et al., 2020) although there have been indications that younger generations of scholars prefer using social media to foster collaboration (Murthy and Lewis, 2015) – which also might relate to the newly discovered phenomenon of preference for citing recent work (Bollmann and Elliott, 2020).

3 Dataset

When constructing our dataset, we limit ourselves to posts promoting academic research in the field of NLP on Twitter in English language. This choice was motivated by two reasons: (1) we believe that for a popularity prediction model to be successful, the domain should be narrow and (2) as all authors of this paper are involved in NLP research, we have a deep personal interest in whether it is possible to determine what constitutes a "good" post which promotes academic reseach on social media.

We first selected a set of NLP researcher Twitter users, which we then manually validated. We then fetched all posts of these users that contained a link which resolved on arXiv[1], and then selected a subset of these posts which formed threads containing comments from the same root user. The latter step was done to avoid bot accounts which automatically share all preprints and as an attempt to ensure that threads contain a summary of the paper referred to. Nevertheless, these simple rules are by no means exhaustive. It is likely that the dataset contains threads from users which do not summarize the paper, while it definitely contains summaries written by users that are not authors of the paper. While we considered manually validating each thread, we chose not to as doing so would make scaling the dataset in the future infeasible. For the sake of space, we omit the detailed description of dataset construction to Appendix A.

3.1 Data Feature Groups

Once finalized, our NLP preprint popularity dataset[2] (henceforth NLPOP) consists of four distinct input feature groups: (1) the preprint title and abstract text, encoded separately (PAPER), (2) the Twitter thread text (THREAD), (3) Twitter user biographical data (BIO) and (4) numeric metadata features (NUM) of the user profile and the Twitter thread. It also contains two target variables: (1) the number of likes and (2) the number of retweets.

The first three feature groups consist of textual data, but differ in style and content. The preprint title and abstract contain the academic style writeup of the research work, the thread text consists of a brief summary which elaborates the key points of the paper in a more informal manner, while the biographical data is a personal description of the researcher. The numeric features consist of various metadata which might be useful for the prediction of the model pertaining to either the user: (1) account creation timestamp, (2,3) number of followers and followings, (4) number of tweets for that user, (5) number of favourites and (6) the number of lists the user is in; or pertaining to the tweet: (7) tweet creation timestamp, and (8) the hour of day (in UTC) the tweet was posted at.

We summarize the statistics of the dataset in Table 1. We do not propose a single pre-made dataset split as multiple ways the dataset could be split exist, which we comment on in Appendix B.

4 Methodology

We will first define the notion of popularity. While some other works (Tan et al., 2014; Zhao et al., 2015) have considered only the final number of

[1] https://arxiv.org/

[2] The dataset is available at https://github.com/lobadic/nlpop

	Dataset size	2292
	Distinct users	858
Feature	Avg.	Std.
Likes	65.6	124.2
Retweets	15.3	36.6
BIO*	5.7	5.7
PAPER*	218.8	195.6
THREAD*	149.0	157.9

Table 1: Dataset statistics. For textual features (annotated with *) the average and standard deviation pertain to length in words. Statistics for the number of likes and retweets are computed on raw scores.

reshares (*retweets*) as the popularity criterion, we also consider predicting the number of *likes* a post receives (Jenders et al., 2013) as another task.

4.1 Task Formulation

As both of our target variables are numeric, a natural course of action is to approach the task as regression (Lampos et al., 2014). However, if the exact value of the target variable is not relevant, it is common to transform the problem into classification by defining thresholds for popularity categories (Fiok et al., 2020).

Regression. Treating the problem as regression (REG) preserves more information from the target variable as we avoid the lossy transformation into a categorical variable. Due to large differences in scale of the output variables, we first scale the target variable by applying the natural logarithm and use the mean squared error (MSE) as the criterion. A task trained this way is still evaluated as a classification task by performing the same transformation into discrete classes on the outputs of the regression model.

Classification. In our case, we follow (Fiok et al., 2020) and opt for the three-class approach (CLF), where the classes: "not popular", "popular" and "very popular" are determined as the lower quartile (bottom 25%), the middle 50% and the top quartile (top 25%). We compute the values for the thresholds on the training split of the dataset.

Ordinal classification. Apart from the information lost in the transformation, another downside of the classification approach is that discrete classes do not retain ordinal information. To this end, we adopt the approach from Frank and Hall (ORD; 2001) and transform the discrete classes into ordinal labels. In this approach, N classes are encoded as a binary vector of length $N-1$, where each bit being set indicates that the target variable is greater than the treshold for that class. Thus, if a bit is set, all the less significant bits also have to be set[3]. Using this approach, the model will learn to model the order between classes – as the popularity increases, the model has to set that many more bits in the output prediction.

4.2 Preprocessing

When preprocessing text inputs, we use spaCy[4] for tokenization, filter punctuation tokens, replace hyperlinks with `<URL>` and separate posts in a thread with `<SEP>`. We consider only the 10000 most frequent word tokens for the models which do not use a pre-trained vocabulary and truncate sequences longer than 512 tokens. The numeric features are scaled to the $[0, 1]$ interval using scikit-learn's[5] `MinMaxScaler`.

We split the dataset in proportions of $0.7 : 0.1 : 0.2$ for the train, validation and test set, respectively. When splitting, we ensure that each user exists in only one of the splits to prevent information leakage via profile information. We attempt to ensure that the distribution of target variables is as similar as possible by running 10000 random splits with different seeds and choosing the one where the means and standard deviations have minimal difference between the splits.

4.3 Models

We consider three model families of text encoders with increasing complexity: an IDF-weighted averaging approach (AVG; Ramos et al., 2003), a GRU-based encoder model (RNN; Cho et al., 2014) and a pretrained RoBERTa-large model (BERT; Liu et al., 2019). For simplicity, we always use the same text encoder to encode all textual input features. In the AVG and RNN models, the word inputs are initialized to 300-dimensional GloVe embeddings (Pennington et al., 2014). Due to the small scale of the dataset, we do not fine-tine the ROBERTA encoder, but use the encodings from the last layer as-is. To obtain a fixed-size representation, we consider averaging the embeddings, pooling them using the

[3]Concretely, for our three-class approach, the vector $[00]$ would correspond to the "not popular" class, the lowermost bit $[01]$ would indicate that the instance is "popular", while both bits being set $[11]$ corresponds to the "very popular" class.

[4]https://spacy.io/

[5]https://scikit-learn.org/stable/

	# Likes			# Retweets		
Feature groups	AVG	RNN	BERT	AVG	RNN	BERT
NUM	39.14			37.10		
BIO	36.90	**40.58**	40.19	35.45	34.59	37.34
PAPER	29.92	29.32	39.06	40.12	24.24	42.52
THREAD	46.65	23.17	54.43	41.19	21.96	**53.14**
NUM, BIO	40.82	37.22	42.29	35.06	**41.07**	40.11
NUM, THREAD	**49.25**	37.90	53.78	**46.59**	31.22	51.68
THREAD, PAPER	41.44	24.56	54.28	38.72	24.52	50.91
BIO, THREAD	47.82	39.36	55.93	39.28	34.53	50.85
NUM, BIO, THREAD	47.13	39.40	56.23	42.03	37.17	52.35
ALL	44.82	40.12	**58.59**	45.88	31.40	51.69

Table 2: Overall best performing models across all considered training tasks for different feature sets. Scores reported are $100\times$macro-F1. Best scores in each column are **boldfaced**, best scores in each row are underlined.

pretrained pooler or taking the encodings of the `SEP` or `CLS` tokens. The encoded outputs of each considered input feature group are concatenated and used as inputs to a MLP classifier. For the sake of space, we detail considered hyperparameters of all models in Appendix C.

5 Results

When reporting results, we will mainly be looking to answer the following questions: (1) do more complex text encoders improve prediction performance?; (2) which feature groups improve the performance the most?; (3) which task type suits the problem the most?; and (4) in which cases do the models make mistakes? We do not report exhaustive ablation combinations for the sake of space and as the unreported combinations perform worse.

To answer the first two questions, we perform an ablation study and report the results in Table 2. Here, we can immediately notice that BERT-based models perform the best, indicating that content does matter for popularity. Secondly, we can see that the RNN model performs the worst. We believe this is caused by the relatively small size of the dataset and the fact that the recurrent encoders need to be trained from scratch, which causes the model to frequently overfit.

Analysing the effect of feature groups, we can see that the THREAD itself performs the best in isolation for both target variables, except for RNN models – indicating that a good summary influences the popularity the most. When analysing the THREAD features in combination with other feature groups for the LIKE prediction case, the BIO offers the most improvement, with PAPER the second most important group, indicating that paper content matters for popularity. For the RETWEET case, surprisingly, adding any feature group diminishes the performance of the BERT model, emphasizing the fact that the content of the thread is the most discriminative feature for determining popularity.

	# Likes			# Retweets		
Task	AVG	RNN	BERT	AVG	RNN	BERT
REG	38.4	35.5	48.4	43.5	29.1	45.6
CLF	**49.3**	**40.6**	**58.6**	**46.6**	**34.6**	**53.1**
ORD	40.8	37.2	54.6	44.0	31.9	52.4

Table 3: Overall best performing models for different task types. Scores reported are $100\times$macro-F1. Best results in each column **boldfaced**, best overall underlined.

Analysing the effect of the task formulation, in Table 3 we can see that the classification task performs best overall, although ordinal classification is the close second for BERT-based models.

Finally, we aim to understand whether the models are able to understand the class boundaries. To this end, we will take a look at the confusion matrices of the best performing models for the `#likes` (Table 4) and `#retweets` (Table 5) prediction tasks. In both tables, we can immediately see that the models generally only make mistakes in neighboring classes – indicating that although some cases might be borderline, the notion of popularity can be estimated from the input features. Furthermore, we can notice that the majority of the errors made are on the boundary between the first two classes, where the distinction between classes is made for a comparatively smaller value of the target variable. We believe that the fuzzy boundary between the two classes causes issues to the model, and in future work we aim to explore whether it is possible to set a clearer boundary.

6 Conclusion

We have introduced NLPOP: a novel dataset for popularity prediction which combines Twitter thread data, academic paper content and biographical user

	$y=0$	$y=1$	$y=2$
$\hat{y}=0$	61	49	9
$\hat{y}=1$	53	171	19
$\hat{y}=2$	11	35	48

Table 4: The confusion matrix of the best performing model (BERT-CLF) on the # Likes prediction task. True classes (y) are represented in columns, predicted classes $\hat{y}$ in rows. Class 0 corresponds to "not popular", 1 to "popular" and 2 to "very popular", respectively.

	$y=0$	$y=1$	$y=2$
$\hat{y}=0$	57	68	7
$\hat{y}=1$	40	156	28
$\hat{y}=2$	5	52	43

Table 5: The confusion matrix of the best performing model (BERT-CLF) on the # Retweets prediction task. True classes (y) are represented in columns, predicted classes $\hat{y}$ in rows. Class 0 corresponds to "not popular", 1 to "popular" and 2 to "very popular", respectively.

features. After carrying out ablation studies on input feature sets we have determined that, while the thread text is the most discriminative input, the content of the academic paper is also indicative of popularity measured in the number of likes. We believe that our dataset will grow at a significant pace over time and that in the future, it could be used to augment data in citation recommendation, as well as an evaluation dataset for abstractive summarization systems.

For future work, we aim to widen the pool of considered users by automating the manual validation process and plan on ensuring that the person promoting the work is an author of the paper – which could improve the quality of the summary. In scope of the paper we focused on presenting a proof-of-concept study, aiming to determine whether it is feasible to predict popularity of Twitter posts based on content, and whether such a dataset of significant size can be collected. We believe we have sufficiently demonstrated the quality of the dataset and the feasibility of the task to indicate its value for related NLP research areas.

References

Marcel Bollmann and Desmond Elliott. 2020. On forgetting to cite older papers: An analysis of the acl anthology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7819–7827.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Krzysztof Fiok, Waldemar Karwowski, Edgar Gutierrez, and Tareq Ahram. 2020. Predicting the volume of response to tweets posted by a single twitter account. *Symmetry*, 12(6):1054.

Eibe Frank and Mark Hall. 2001. A simple approach to ordinal classification. In *European conference on machine learning*, pages 145–156. Springer.

Anna Gorska, P Korzynski, G Mazurek, and F Pucciarelli. 2020. The role of social media in scholarly collaboration: an enabler of international research team's activation? *Journal of Global Information Technology Management*, 23(4):273–291.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Liangjie Hong, Ovidiu Dan, and Brian D Davison. 2011. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58.

Aaron Jaech, Victoria Zayats, Hao Fang, Mari Ostendorf, and Hannaneh Hajishirzi. 2015. Talking to the crowd: What do people react to in online discussions? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2026–2031.

Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. 2013. Analyzing and predicting viral tweets. In *Proceedings of the 22nd international conference on world wide web*, pages 657–664.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709.

Mohammad Reza Karimi, Erfan Tavakoli, Mehrdad Farajtabar, Le Song, and Manuel Gomez Rodriguez. 2016. Smart broadcasting: Do you want to be seen? In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1635–1644.

Vasileios Lampos, Nikolaos Aletras, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2014. Predicting and characterising user impact on twitter. In *14th conference of the European chapter of the Association for Computational Linguistics 2014, EACL 2014*, pages 405–413.

Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, pages 577–586.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dhiraj Murthy and Jeremiah P Lewis. 2015. Social media, collaboration, and scientific organizations. *American behavioral scientist*, 59(1):149–171.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 1, pages 29–48. Citeseer.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

Shivashankar Subramanian, Timothy Baldwin, and Trevor Cohn. 2018. Content-based popularity prediction of online petitions using a deep regression model. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 182–188.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 175–185.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ali Zarezade, Utkarsh Upadhyay, Hamid R Rabiee, and Manuel Gomez-Rodriguez. 2017. Redqueen: An online algorithm for smart broadcasting in social networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 51–60.

Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1513–1522.

Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2021. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

A Dataset construction details

To start the dataset construction process, we needed to create a set of Twitter accounts which we knew belonged to NLP researchers. Our initial set of users consisted of the Twitter followings of one of the authors (175 users). We then expanded this set by fetching users whose Twitter biographies contained a NLP keyword (NLP, CL or their expansions) and a general AI keyword (ML, AI or their expansions; to ensure that we avoid neuro-linguistic programming), which yielded 608 new users. We then manually validated the collected users to ensure quality, where after removing 26 users a total of 757 remained in the initial set.

We further expanded this initial set of users by fetching all the followers and followings of each user in the initial set (yielding a pool of 1.14M users). We then applied a similar filtering procedure, but retaining users which had a NLP keyword in their Twitter biography, resulting in 7851 new candidate users. This candidate set was once more manually verified, resulting in 7079 new users and a total of 7836 accounts in the final set (USERS).

In the next step we aimed to retrieve the posts of USERS which promote research work. To do this, we fetched only the posts which contained a link leading to arXiv[6], where it is categorized in the Computation and Language (cs.CL) category, either as the primary or secondary category. From these posts, we selected only the ones that formed

[6] https://arxiv.org/

Name	Value(s)
Max epochs	100
Optimizer	Adam
Patience	15
Batch size	32
AVG	
Vocabulary size	10000
Learning rate	$[1e^{-4}, 5e^{-4}, 1e^{-5}, 5e^{-5}]$
Freeze embeddings	True
Classifier hidden	$[512, 256]$
RNN	
Vocabulary size	10000
Learning rate	$[1e^{-4}, 5e^{-4}]$
Max seq length	512
Freeze embeddings	[True, False]
GRU hidden	$[128, 300]$
GRU dropout	0.3
GRU layers	2
Bidirectional	True
Classifier hidden	[300]
BERT	
Learning rate	$[1e^{-4}, 5e^{-4}, 1e^{-5}, 5e^{-5}]$
Max seq length	512
Classifier hidden	$[512, 256]$
Freeze model	True
Pooling strategy	[AVG, POOL, CLS, SEP]

Table 6: Hyperparameters of Considered Models

a thread (had more than one comment) in order to attempt to ensure that a brief description is provided by the person posting the link, and to avoid automated accounts which merely share the links to newly submitted papers on arXiv. We selected threads as the root post and all consecutive replies by the original poster to themselves. This selection process resulted in 2292 threads written by 858 distinct users. For each of these threads, we also retrieve the title and abstract of the preprint on arXiv. We further augment the dataset with Twitter biographical user data and thread metadata retrieved via the Twitter API[7]. The dataset was last updated on the 19th of October 2021.

B Dataset Splits

When splitting the dataset, there is a number of options we considered. We started with a completely random split as an initial step to be able to determine whether more intelligent ways of splitting the dataset improve performance by reducing bias (RANDOM). Our next step was to ensure that there is no user overlap between the dataset splits, attempting to minimize information leakage and the models overfitting to user data (USERS). This procedure, however, yielded imbalanced splits with respect to the target variables. To mitigate this issue, we resorted to a random search, where we ran the same splitting procedure 10000 times with different random seeds and selected the splits with minimal difference between the mean and standard deviations of the target variables (USERS-DIST). The determined thresholds for classes are $[0, 9)$ for "not popular", $[9, 71)$ for "popular" and $[71, \infty)$ for the "very popular" class in the like prediction scenario, while the respective thresholds are $[0, 2)$, $[2, 16)$ and $[16, \infty)$ for the retweet prediction scenario.

C Model Hyperparameters

When running our models, we fix some hyperparameters using manual tuning to reduce the search space and perform an exhaustive search over the remaining combinations. The full set of hyperparameters for all models is listed in Table 6. The best hyperparameters were selected with respect to model performance on the validation split, where $F1$ was the metric for classification models and MSE for regression models. All experiments were ran on four Nvidia GTX 1080 graphics cards.

[7] https://developer.twitter.com/en/docs/twitter-api