Learning Invariant Representations of Social Media Users

Nicholas Andrews and Marcus Bishop
Human Language Technology Center of Excellence
Johns Hopkins University
{noa,marcus.bishop}@jhu.edu

Abstract

The evolution of social media users' behavior over time complicates user-level comparison tasks such as verification, classification, clustering, and ranking. As a result, naïve approaches may fail to generalize to new users or even to future observations of previously known users. In this paper, we propose a novel procedure to learn a mapping from short episodes of user activity on social media to a vector space in which the distance between points captures the similarity of the corresponding users' invariant features. We fit the model by optimizing a surrogate metric learning objective over a large corpus of unlabeled social media content. Once learned, the mapping may be applied to users not seen at training time and enables efficient comparisons of users in the resulting vector space. We present a comprehensive evaluation to validate the benefits of the proposed approach using data from Reddit, Twitter, and Wikipedia.

1 Introduction

Social media presents a number of challenges for characterizing user behavior, chief among them that the topics of discussion and their participants evolve over time. This makes it difficult to understand and combat harmful behavior, such as election interference or radicalization (Thompson, 2011; Mihaylov and Nakov, 2016; Ferrara et al., 2016; Keller et al., 2017).

This work focuses on the fundamental problem of learning to compare social media users. We propose a procedure to learn embeddings of small samples of users' online activity, which we call *episodes*. This procedure involves learning the embedding using a metric learning objective that causes episodes by the same author to map to nearby points. Through this embedding users may be efficiently compared using cosine similarity. This representation immediately enables several tasks:

Verification. Determining if two episodes have the same author.

Classification. Labeling authors via their k-nearest neighbors.

Clustering. Grouping users via off-the-shelf methods like k-means or agglomerative clustering.

Ranking and retrieval. Sorting episodes according to their distances to a given episode.

The problem considered in this paper is most closely related to author attribution on social media. However, prior work in this area has primarily focused on classifying an author as a member of a closed and typically small set of authors (Stamatatos, 2009; Schwartz et al., 2013; Shrestha et al., 2017). In this paper, we are concerned with an *open-world* setting where we wish to characterize an *unbounded* number of users, some observed at training time, some appearing only at test time. A further challenge is that the episodes being compared may be drawn from different time periods. With these challenges in mind, the primary contributions described in this paper are as follows:

§**3** A training strategy in which a user's history is dynamically sampled at training time to yield multiple short episodes drawn from different time periods as a means of learning invariant features of the user's identity;

§**4** A user embedding that can be trained end-to-end and which incorporates text, timing, and context features from a sequence of posts;

§**5** Reddit and Twitter benchmark corpora for open-world author comparison tasks, which are substantially larger than previously considered;

§**6** Large-scale author ranking and clustering experiments, as well as an application to Wikipedia sockpuppet verification.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1684–1695,
Hong Kong, China, November 3–7, 2019.

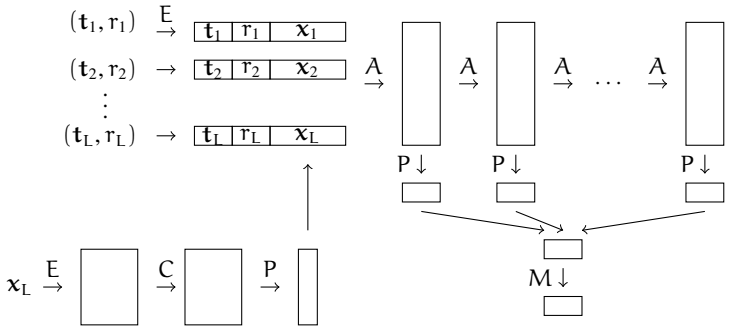


Figure 1: The map f_θ takes an episode as input and outputs a vector. Here A denotes a multi-head self-attention layer, C a stack of 1D convolutions, E an embedding lookup, M an MLP, and P a pooling layer.

2 Preliminaries

Broadly speaking, a corpus of social media data consists of the *actions* of a number of users. Each action consists of all available information from a given platform detailing what exactly the user did, which for purposes of this work we take to include: (1) a timestamp recording when the action occurred, from which we extract a tuple $\mathbf{t}$ of temporal features, (2) unstructured text content $\mathbf{x}$ of the action, and (3) a categorical feature r specifying the context of the action. Thus an action is a tuple of the form $(\mathbf{t}, \mathbf{x}, \mathrm{r})$. This formulation admits all three platforms considered in this work and therefore serves as a good starting point. However, incorporating features specific to particular platforms, such as image, network, and moderation features, might also provide useful signal.

In our experiments we use a data-driven subword representation (Kudo, 2018) of $\mathbf{x}$, which admits multilingual and non-linguistic content, as well as misspellings and abbreviations, all of which useful in characterizing authors. We use a simple discrete time feature for $\mathbf{t}$, namely the hour of the day, although others might be helpful, such as durations between successive actions. In our Reddit experiments we take r to be the subreddit to which a comment was posted. On Twitter we take r to be a flag indicating whether the post was a tweet or a retweet.

3 Learning Invariant Representations

We organize the actions of each user into short sequences of chronologically ordered and ideally contiguous actions, which we call *episodes*. This paper is concerned with devising a notion of distance between episodes for which episodes by the same author are closer to one another than episodes by different authors. Such a distance function must necessarily be constructed on the basis of *past* social media data. But in the future, authors' behavior will evolve and new authors will emerge.

We would like episodes by the same author to be nearby, irrespective of when those episodes took place, possibly *future* to the creation of the distance function. A given user will discuss different topics, cycle through various moods, develop new interests, and so on, but distinctive features like uncommon word usage, misspellings, or patterns of activity will persist for longer and therefore provide useful signal for the distance function.

We would also like the distance to be meaningful when applied to episodes by users who didn't exist when the distance function was created. To this end, the features it considers must necessarily generalize to new users. For example, common stylometric features will be shared by many users, including new users, but their particular combination is distinctive of particular users (Orebaugh and Allnutt, 2009; Layton et al., 2010).

Rather than heuristically defining such a distance function, for example, based on word overlap between the textual content of the episodes, we instead introduce a parameterized embedding f_θ shown in Figure 1 that provides a vector representation of an episode. Then the desired distance between episodes can be taken to be the distance between the corresponding vectors. We fit the embedding f_θ using *metric learning* to simultaneously decrease the distance between episodes by the same user and increase the distance between

episodes by different users (Bromley et al., 1994; Wang et al., 2014).

But doing so requires knowledge of the true author of an episode, something which is not generally available. Therefore we take account names to be an approximation of latent authorship. Of course, account names are not always a reliable indicator of authorship on social media, as the same individual may use multiple accounts, and multiple individuals may use the same account. As such, we expect a small amount of label noise in our data, to which neural networks have proven robust in several domains (Krause et al., 2016; Rolnick et al., 2017).

We fit f_θ to a corpus of social media data using stochastic gradient descent on batches of examples, where each example consists of an episode of a given length drawn uniformly at random from the *full history* of each user's actions.[1] By construction, a metric learning objective with this batching scheme will encourage the embedding of episodes drawn from the same user's history to be close. In order to accomplish this, the model will need to distinguish between ephemeral and invariant features of a user. The invariant features are those that enable the model to consistently distinguish a given users' episodes from those of *all other users*.

4 The Model

We now describe a mapping f_θ parameterized by a vector θ from the space of user episodes to $\mathbb{R}^D$. The model is illustrated in Figure 1. This embedding induces a notion of distance between episodes that depends on which of the two proposed loss functions from §4.2 is used to train f_θ. We illustrate the embeddings resulting from both losses in Figure 2.

4.1 The encoder

One approach to define f_θ might be to manually define features of interest, such as stylometric or surface features (Solorio et al., 2014; Sari et al., 2018). However, when large amounts of data are available, it is preferable to use a data-driven approach to representation learning. Therefore we define f_θ using a neural network as follows. The network is illustrated in Figure 1.

[1]Different metric learning methods will sample users in different ways, for example to ensure a given ratio of examples of the same class. In this work we simply sample users uniformly at random.

Encoding actions. First, we embed each action (t, x, r) of an episode. We encode the time features t and the context r, both assumed to be discrete, using a learned embedding lookup. We next embed every symbol of x, again using a learned embedding lookup, and apply one-dimensional convolutions of increasing widths over this list of vectors, similar to Kim (2014); Shrestha et al. (2017). We then apply the relu activation and take the componentwise maximum of the list of vectors to reduce the text content to a single, fixed-dimensional vector. We optionally apply dropout at this stage if training. Finally, we concatenate the time, text, and context vectors to yield a single vector representing the action.

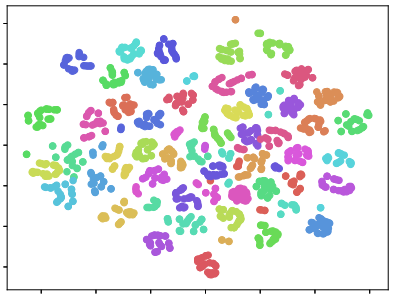
Embedding episodes. Next we combine the vector representations of the actions of an episode. For this purpose, one option is a recurrent neural network (RNN). However, recurrent models are biased due to processing inputs sequentially, and suffer from vanishing and exploding gradients. Therefore we propose the use of self-attention layers, which avoid the sequential biases of RNNs and admit efficient implementations.

In our particular formulation, we use several layers of multi-head self-attention, each taking the output of the previous layer as input; architectural details of the encoder layers follow those of the Transformer architecture proposed by Vaswani et al. (2017). We apply mean pooling after every layer to yield layer-specific embeddings, which we concatenate. We project to the result to the desired embedding dimension D using an MLP, both its input and output batch normalized (Ioffe and Szegedy, 2015).

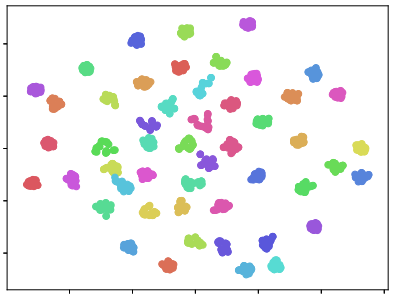
4.2 The loss function

For the purpose of training the embedding f_θ we compose it with a discriminative classifier $g_\phi : \mathbb{R}^D \to \mathbb{R}^Y$ with parameters ϕ predicting the author of an episode, where Y is the number of authors in the training set. We estimate θ and ϕ jointly using a standard cross-entropy loss on a corpus of examples with their known authors. Once the model is trained, the auxiliary projection g_ϕ is discarded. Two possibilities for g_ϕ are proposed below.

Softmax (SM). We introduce a weight matrix $W \in \mathbb{R}^{Y \times D}$ and define the map $g_\phi(z) = \mathrm{softmax}(Wz)$ with parameters $\phi = W$. When using this loss function, one compares embed-



(a) Embeddings obtained using **SM** loss.



(b) Embeddings obtained using **AM** loss.

Figure 2: Projections of embeddings of user episodes. Each point is the result of mapping an episode to a single point in $\mathbb{R}^{512}$ and projected to $\mathbb{R}^2$ using t-SNE. The colors of the points correspond with the 50 different authors of the underlying episodes. We emphasize that the episodes shown here were not seen by the model at training time.

dings using Euclidean distance.

Angular margin (AM). Following Deng et al. (2019) we again introduce a weight matrix $\boldsymbol{W} \in \mathbb{R}^{Y \times D}$ whose rows now serve as *class centers* for the training authors. Given the embedding $\boldsymbol{z} \in \mathbb{R}^D$ of an episode, let $\boldsymbol{z}' = \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$ be the normalization of $\boldsymbol{z}$ and let $\boldsymbol{W}'$ be obtained from $\boldsymbol{W}$ by normalizing its rows. Then the entries of $\boldsymbol{w} = \boldsymbol{W}'\boldsymbol{z}'$ give the cosines of the angles between $\boldsymbol{z}$ and the class centers. Let $\boldsymbol{w}'$ be obtained from $\boldsymbol{w}$ by modifying the entry corresponding with the correct author by adding a fixed margin $m > 0$ to the corresponding angle.[2] Finally, define $g_\Phi(\boldsymbol{z}) = \text{softmax}(s\boldsymbol{w}')$ where $s > 0$ is a fixed *scale constant*. When using this loss function, one compares embeddings using cosine similarity.

5 Corpora for Large-Scale Author Identification

5.1 Reddit benchmark

Reddit is a large, anonymous social media platform with a permissive public API. Using Reddit consists of reading and posting *comments*, which consist of informal text, primarily in English, each appearing within a particular *subreddit*, which we treat as a categorical feature providing useful contextual signal in characterizing users.

We introduce a new benchmark author identification corpus derived from the API (Gaffney and Matias, 2018) containing Reddit comments by 120,601 active users for training and 111,396 held-out users for evaluation. The training split contains posts published in `2016-08` while the evaluation split contains posts published in `2016-09`. In both cases, we restrict to users publishing at least 100 comments but not more than 500. The lower bound ensures that we have sufficient evidence for any given user for training, while the upper bound is intended to mitigate the impact of bots and atypical users. The evaluation split is disjoint from the training split and contains comments by 42,121 novel authors not contributing to the training split.

Validation. For model selection, we use the first 75% of each user's chronologically ordered posts from the training set, with the final 25% reserved for validation. For example, in our ranking experiments described in §6.3 we use these held-out comments as candidate targets, using ranking performance to inform hyper-parameter choice.

5.2 Twitter benchmark

The microblog domain is sufficiently distinct from Reddit that it is suitable as an additional case study. For this purpose, we sample 169,663 active Twitter users from three months of 2016 as separate training, development, and test sets (`2016-08` through `2016-10`). We use three months because we rely on a sub-sampled collection of Twitter, as little as 1% of all posts published, resulting in significantly fewer posts by each user than on Reddit. Another consequence of this sub-sampling is that the collection violates our assumptions regarding *contiguous* user actions.

[2] One way to calculate $\cos(\theta + m)$ from $\cos\theta$ is $\cos\theta\cos m - \sin\theta\sin m$ where $\sin\theta$ is calculated as $\sqrt{1-\cos^2\theta}$. Note that this calculation discards the sign of θ.

6 Experiments

In the experiments described below, we refer to our method as IUR for Invariant User Representations.

6.1 Baseline methods

In order to validate the merit of each of our modeling contributions, we compare against three baseline models described below. To the best of our knowledge, we are the first to consider using metric learning to learn embeddings from episodes of user activity. We are also the first to consider doing so in open-world and large-scale settings. As such, the neural baseline described below uses the training scheme proposed in this paper, and was further improved to be more competitive with the proposed model.

Neural author identification. We use the architecture proposed by Shrestha et al. (2017) for closed-set author attribution in place of our $\mathtt{f}_\theta$. At the level of individual posts this architecture is broadly similar to ours in that it applies 1D convolutions to the text content. To extend it to episodes of comments, we simply concatenate the text content into a single sequence with a distinguished end-of-sequence marker. Note that the timing and context features may also be viewed as sequences, and in experiments with these features we run a separate set of one-dimensional filters over them. All max-over-time pooled features are concatenated depthwise. By itself, this model failed to produce useful representations; we found it necessary to apply the batch-normalized MLP described in §4.1 to the output layer before the loss. To train the model, we follow the procedure described in §4.2 to compose the embedding with the **SM** loss function, optimize the composition using cross-entropy loss, and discard the **SM** factor after training.

Document vectors. By concatenating all the textual content of an episode we can view the episode as a single document. This makes it straightforward to apply classical document indexing methods to the resulting pseudo-document. As a representative approach, we use TFIDF with cosine distance (Robertson, 2004). We note that TFIDF is also well-defined with respect to arbitrary bags-of-items, and we make use of this fact to represent a user according to the sequence of subreddits to which they post as a further baseline in §6.3.

Author verification models. We use the SCAP $\mathfrak{n}$-gram profile method of Frantzeskou et al. (2007). Two episodes are compared by calculating the size of the intersection of their $\mathfrak{n}$-gram profiles. We use profiles of fixed length 64 in our experiments.

6.2 Model hyperparameters and training

Below we list our hyperparameter choices for the IUR model, which we define in §4.

For both Twitter and Reddit, we estimate the sub-word vocabulary on training data using an inventory of 65,536 word pieces, including a distinguished end-of-sequence symbol. We truncate comments to 32 word pieces, padding if necessary.[3] We restrict to the 2048 most popular subreddits, mapping all others to a distinguished `unk` symbol. We encode word pieces and subreddits as 256-long vectors. The architecture for the text content uses four convolutions of widths 2, 3, 4, 5 with 256 filters per convolution. We use two layers of self-attention with 4 attention heads per layer, and hidden layers of size 512. Other details such as use of layer normalization match the recommendations of Vaswani et al. (2017).

We train all variations of the IUR for a fixed budget of 200,000 iterations of stochastic gradient descent with momentum 0.9 and a piecewise linear learning rate schedule that starts at 0.1 and is decreased by a factor of 10 at 100,000 and 150,000 iterations. The final MLP has one hidden layer of dimension 512 with output also of dimension $\mathrm{D} = 512$. For the angular margin loss we take $\mathfrak{m} = 0.5$ and $\mathrm{s} = 64$ as suggested in Deng et al. (2019).

6.3 Reddit ranking experiment

Given a query episode by a known user, our author ranking experiment consists of returning a list of target episodes ranked according to their similarity to the query. The problem arises in the moderation of social media content, when say, a user attempts to circumvent an account ban by using another account.

Experimental setup. Recall that we train all Reddit models on the `2016-08` split. In this experiment we draw episodes from the first half of `2016-09` as *queries* and the second half of `2016-09` as *targets*. Specifically, for

[3] In experiments not reported here, we have found that increasing the number of subwords per action increases performance but at the cost of slower training.

Input Features	Method	MRR (⇑)	MR (⇓)	R@1 (⇑)	R@2	R@4	R@8
text only	SCAP	0.0057	31292	0.0035	0.004	0.0075	0.0085
	TF-IDF (word)	0.071	5548	0.048	0.065	0.084	0.11
	TF-IDF (char trigram)	0.07	6264	0.05	0.066	0.081	0.1
	Shrestha et al. (2017)	0.081	660	0.052	0.071	0.094	0.12
	IUR	**0.2**	**88**	**0.15**	**0.19**	**0.24**	**0.29**
subreddit only	TF-IDF	0.1	305	0.068	0.091	0.12	0.16
	Shrestha et al. (2017)	0.18	110	0.12	0.16	0.21	0.26
	IUR	**0.21**	**64**	**0.15**	**0.2**	**0.24**	**0.3**
text, subreddit, time	Shrestha et al. (2017)	0.39	8	0.31	0.38	0.45	0.51
	IUR (softmax loss)	0.38	9	0.31	0.38	0.44	0.49
	IUR (recurrent encoder)	0.34	17	0.27	0.33	0.39	0.44
	IUR (without time)	0.48	3	0.41	0.48	0.55	0.61
	IUR	**0.52**	**2**	**0.44**	**0.52**	**0.59**	**0.65**

Table 1: Reddit author ranking results with 111,396 possible targets. The best results for each feature group are in printed in **bold**. The proposed Invariant User Representations are denoted IUR, with variations of the full model noted in parenthesis. MRR stands for the mean reciprocal rank, MR for median rank, and R@k stands for recall at the top k ranked episodes. Larger numbers are better (⇑) except for MR where lower rank is better (⇓). Metrics are computed over 25,000 queries.

each of 25,000 randomly selected users from the `2016-09` split we randomly draw a query episode of length 16 from among those posts published by that user before `2016-09-15`. Then for each of the 111,396 users in the `2016-09` split we randomly draw a target episode of length 16 from among those posts published by that user on or after `2016-09-15`. For each query, the goal of the experiment is to rank the targets according to their likelihoods of being the *unique* target composed by the author of the query.

We compare models using mean reciprocal rank (MRR), median rank (MR), and recall-at-k (R@k) for various k. The MRR is the mean over all 25,000 queries of the reciprocal of the position of the correct target in the ranked list. The MR is the median over the queries of the position of the correct target. The R@k is the proportion of the queries for which the correct target appears among the first k ranked targets.

Results. The results of this experiment are shown in Table 1. For each combination of features considered, the rankings based on the proposed IUR embeddings consistently outperform all methods considered, both neural and classical. We also report results on several variations of our model, noted in parenthesis. First, using the proposed architecture for f_θ but the softmax loss results in ranking performance comparable to the baseline system. Second, using a recurrent architecture rather than self-attention to aggregate information across an episode results in significantly worse performance.[4] Finally, omitting time features results in worse performance.

Performance on novel users. As described above, the experiments presented in Table 1 involved ranking episodes by test authors, some of whom had been seen during training, and some new to the model. To better understand the ability of the proposed embedding to generalize to new users, we performed a further evaluation in which authors were restricted to those *not* seen at training time. For the IUR incorporating all features, this yielded a MRR of 0.50, while our extension of Shrestha et al. (2017) obtains 0.38 for the same queries. Both methods produce salient embeddings of novel users, but IUR retains an edge over the baseline.

Varying episode length. As described above, the experiments presented in Table 1 involved episodes of length exactly 16. In Figure 3, we report results of a further ranking experiment in which we vary the episode length, both at training time and at ranking time. For both the proposed IUR and our extension of Shrestha et al. (2017), performance increases as episode length increases. Furthermore, even for the shortest episodes considered, the proposed approach performs better. This illustrates that the choice of episode length should be decided on an application-specific basis. For example, for social media moderation, it

[4]We choose RNN hyper-parameters such that the numbers of parameters of both models are on the same order of magnitude.

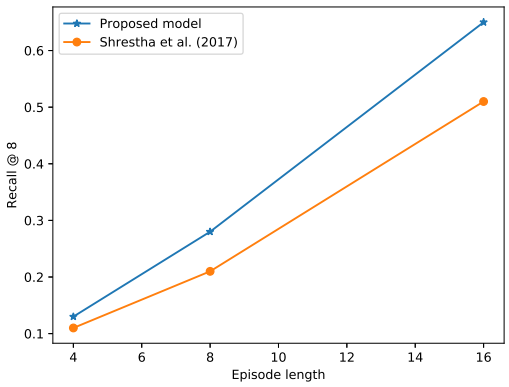


Figure 3: We report Recall@8 for different episode lengths using all features.

may be desirable to quickly identify problematic users on the basis of as few posts as possible.

6.4 Twitter ranking experiment

We repeat the experiment described in §6.3 using data from Twitter in place of Reddit, and with the further difference that the queries were drawn from `2016-08` and the targets from `2016-10` as a mitigation of Twitter's 1% censorship. Unlike the Reddit dataset, all three data splits contain posts by exactly the same authors. The results are shown in Table 2.

6.5 Wikipedia sockpuppet verification

In this section we describe an experiment for the task of sockpuppet verification on Wikipedia using the dataset collected by Solorio et al. (2014). Wikipedia allows editors to open cases against other editors for using suspected *sockpuppet* accounts to promote their contributions. We have reorganized the dataset into pairs of episodes by different accounts. Half of our examples contain a pair deemed by the community to have the same author, while half have been deemed to have different authors. The task is to predict whether a pair of episodes was composed by the same author.

We are interested in whether the text-only version of our IUR model, trained on Reddit data, is able to transfer effectively to this task. This domain is challenging because in many cases sockpuppet accounts are trying to hide their identity, and furthermore, Wikipedia talk pages contain domain-specific markup which is difficult to reliably strip or normalize. Naturally we expect that the identities of Wikipedia editors do not overlap with Reddit authors seen at training time, since the data is drawn from different time periods and from different platforms.

As a baseline, we compare to BERT, a generic text representation model trained primarily on Wikipedia article text (Devlin et al., 2018). While BERT is not specifically trained for author recognition tasks, BERT has obtained state-of-the-art results in many pairwise text classification tasks including natural language inference, question pair equivalence, question answering, and paraphrase recognition. The BERT model used here has 110 million parameters compared to 20 million for our embedding.

Setup. Because many comments are short, we preprocess the data to ensure that each comment has at least 5 whitespace-separated tokens. We restrict to users contributing at least 8 such comments. This left us with 180 cases which we split into 72 for training, and 54 each for validation and testing. We fine-tune both the cased and uncased pretrained English BERT models for our sockpuppet detection task using public models and software.[5] In order to combine the comments comprising an episode for BERT, we explored different strategies, including encoding each comment separately. We found that simply combining comments together and using a long sequence length of 512 gave the best validation performance. For our model, we fine-tune by fitting an MLP on top of our embeddings using binary cross entropy and keeping other parameters fixed. Both methods are tuned on validation data, and the best hyperparameter configuration is then evaluated on held-out test data.

Results. Results are reported in Table 3. The best validation performance is obtained by the cased BERT model. However, both BERT models appear to overfit the training data as test performance is significantly lower. Regarding the proposed IUR model, we see that its performance on validation data is comparable to BERT while generalizing better to held-out test data. For reference, Solorio et al. (2013) report accuracy of 68.83 using the same data using a SVM with hand-crafted features; however, neither their experimental splits nor their model are available for purposes of a direct comparison.

[5] https://github.com/google-research/bert

Method	MRR (⇑)	MR (⇓)	R@1 (⇑)	R@2	R@4	R@8	R@16	R@32
TF-IDF (word)	0.060	4447	0.048	0.057	0.067	0.077	0.092	0.110
TF-IDF (char trigram)	0.070	1622	0.052	0.064	0.078	0.095	0.120	0.140
SCAP	0.049	3582	0.037	0.044	0.053	0.065	0.08	0.098
Shrestha et al. (2017)	0.056	577	0.030	0.050	0.070	0.090	0.130	0.140
IUR (text, time, context)	0.113	179	0.073	0.100	0.130	0.170	0.224	0.287
IUR (text only)	0.117	161	0.077	0.100	0.133	0.176	0.228	0.293
IUR (text, time)	**0.119**	**154**	**0.078**	**0.103**	**0.137**	**0.178**	**0.234**	**0.305**

Table 2: Twitter ranking results with 25,000 queries and with 169,663 possible targets.

	Validation	Test
Majority baseline	0.5	0.5
BERT (uncased)	0.72	0.65
BERT (cased)	**0.76**	0.61
IUR (text-only)	0.74	**0.72**

Table 3: Validation and test accuracy for the Wikipedia sockpuppet task. Best results in **bold**.

	NMI	H	C
Shrestha et al. (2017)	0.54	0.39	0.74
IUR	**0.76**	**0.70**	**0.84**

Table 4: Clustering performance on Reddit episodes using embeddings obtained with different methods.

6.6 Clustering users

For certain tasks it is useful to identify groups of accounts shared by the same author or to identify groups of accounts behaving in a similar fashion (Solorio et al., 2013; Tsikerdekis and Zeadally, 2014). To this end, we experiment with how well a clustering algorithm can partition authors on the basis of the cosine similarity of their IUR episode embeddings.

Procedure. Using the pre-trained Reddit IUR model, we embed five episodes of length 16 by 5000 users selected uniformly at random, all drawn from the held-out `2016-09` split. The embeddings are clustered using affinity propagation, hiding both the identities of the users as well as the true number of users from the algorithm (Frey and Dueck, 2007). Ideally the algorithm will arrive at 5000 clusters, each containing exactly five episodes by same author. Clustering performance is evaluated using mutual information (NMI), homogeneity (H), and completeness (C) (Rosenberg and Hirschberg, 2007). NMI involves a ratio of the mutual information of the clustering and ground truth. Homogeneity is a measure of cluster purity. Completeness measures the extent to which data points by the same author are elements of the same cluster. All three measures lie in interval $[0, 1]$ where 1 is best. The results are shown in Table 4.

7 Related Work

This work considers the problem of learning to compare users on social media. A related task which has received considerably more attention is predicting user attributes (Han et al., 2014; Sap et al., 2014; Dredze et al., 2013; Culotta et al., 2015; Volkova et al., 2015; Goldin et al., 2018). The inferred user attributes have proven useful for social science and public health research (Mislove et al., 2011; Morgan-Lopez et al., 2017). While author attributes like gender or political leaning may be useful for population-level studies, they are inadequate for identifying particular users.[6]

More generally, learning representations for downstream tasks using unsupervised training has recently emerged as an effective way to mitigate the lack of task-specific training data (Peters et al., 2018; Devlin et al., 2018). In the context of social media data, unsupervised methods have also been explored to obtain vector representations of individual posts on Twitter (Dhingra et al., 2016; Vosoughi et al., 2016). Our approach is distinguished from this prior work in several respects. First, we embed episodes consisting of multiple documents, which involves aggregating features. Second, for each document, we encode both textual features as well as associated meta-data. Finally, our training procedure is discriminative, embedding episodes into a vector space with an im-

[6]We leave as future work the question of whether the episode embeddings proposed in this paper are useful for attribute prediction.

mediately meaningful distance.

When social network structure is available, for example on Twitter via *followers*, it may be used to learn user embeddings (Tang et al., 2015; Grover and Leskovec, 2016; Kipf and Welling, 2016). Graph representations have successfully been combined with content-based features; for example, Benton et al. (2016) propose matrix decomposition methods that exploit complementary features of Twitter authors. Graph-based embeddings have proven useful in downstream applications such as entity linking (Yang et al., 2016). However, such methods are not applicable when network structure is unavailable or unreliable, such as with new users or on social media platforms like Reddit. In this work, we are motivated in part by adversarial settings such as moderation, where it is desirable to quickly identify the authorship of novel users on the basis of sparse evidence.[7]

The most closely related work is author identification on social media. However, previous work in this area has largely focused on distinguishing among small, closed sets of authors rather than the open-world setting of this paper (Mikros and Perifanos, 2013; Ge et al., 2016). For example, Schwartz et al. (2013) consider the problem of assigning single tweets to one of a closed set of 1000 authors. Overdorf and Greenstadt (2016) consider the problem of cross-domain authorship attribution and consider 100 users active on multiple platforms. In a different direction, Sari et al. (2018) seek to identify stylistic features contributing to successful author identification and consider a closed set of 62 authors. In contrast, the present work is concerned with problems involving several orders of magnitude more authors. This scale precludes methods where similarity between examples is expensive to compute, such as the method of Koppel and Winter (2014).

Prior work on detecting harmful behavior like hate speech has focused on individual documents such as blog posts or comments (Spertus, 1997; Magu et al., 2017; Pavlopoulos et al., 2017; Davidson et al., 2017; de la Vega and Ng, 2018; Basile et al., 2019; Zampieri et al., 2019). Recently, there have been some efforts to incorporate user-level information. For example, for the supervised task of abuse detection, Mishra et al. (2018) find consistent improvements from incorporating user-level features.

8 Conclusion

Learning meaningful embeddings of social media users on the basis of short episodes of activity poses a number of challenges. This paper describes a novel approach to learning such embeddings using metric learning coupled with a novel training regime designed to learn invariant user representations. Our experiments show that the proposed embeddings are robust with respect to both novel users and data drawn from future time periods. To our knowledge, we are the first to tackle open-world author ranking tasks by learning a vector space with a meaningful distance.

There are several natural extensions of this work. An immediate extension is to further scale up the experiments to Web-scale datasets consisting of millions of users, as has been successfully done for face recognition (Kemelmacher-Shlizerman et al., 2016). Sorting episodes according to their distances to a query can be made efficient using a number of approximate nearest neighbor techniques (Indyk and Motwani, 1998; Andoni and Indyk, 2006).

We are also considering further applications of the proposed approach beyond those in this paper. For example, by restricting the features considered in the encoder to text-alone or text and temporal features, it would be interesting to explore cross-domain author attribution (Stamatatos et al., 2018). It would also be interesting to explore community composition on the basis of the proposed embeddings (Newell et al., 2016; Waller and Anderson, 2019).

Finally, it bears mentioning that the proposed model presents a double-edged sword: methods designed to identify users engaging in harmful behavior could also be used to identify authors with legitimate reasons to remain anonymous, such as political dissidents, activists, or oppressed minorities. On the other hand, methods similar to the proposed model could be developed for such purposes and not shared with the broader community. Therefore, as part of our effort to encourage positive applications, we release source code to reproduce our key results.[8]

[7] Incorporating social network information in our model as additional features is in principle straightforward, requiring only minor architectural changes to the model; the metric learning procedure would otherwise remain the same.

[8] http://github.com/noa/iur.

References

Alexandr Andoni and Piotr Indyk. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 459–468. IEEE Computer Society.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

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)*, volume 2, pages 14–19.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Aron Culotta, Nirmal Ravi Kumar, and Jennifer Cutler. 2015. Predicting the demographics of twitter users from website traffic data. In *AAAI*, pages 72–78.

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

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274.

Mark Dredze, Michael J Paul, Shane Bergsma, and Hieu Tran. 2013. Carmen: A twitter geolocation system with applications to public health. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Emilio Ferrara, Onur Varol, Filippo Menczer, and Alessandro Flammini. 2016. Detection of promoted social media campaigns. In *tenth international AAAI conference on web and social media*.

Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis, Carole E Chaski, and Blake Stephen Howald. 2007. Identifying authorship by byte-level n-grams: The source code author profile (scap) method. *International Journal of Digital Evidence*, 6(1):1–18.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Devin Gaffney and J Nathan Matias. 2018. Caveat emptor, computational social science: Large-scale missing data in a widely-published reddit corpus. *PloS one*, 13(7):e0200162.

Zhenhao Ge, Yufang Sun, and Mark JT Smith. 2016. Authorship attribution using a neural network language model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Gili Goldin, Ella Rabinovich, and Shuly Wintner. 2018. Native language identification with user generated content. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Brussels, Belgium. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA. ACM.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Franziska B Keller, David Schoch, Sebastian Stier, and JungHwan Yang. 2017. How to manipulate social media: Analyzing political astroturfing using ground truth data from south korea. In *Eleventh International AAAI Conference on Web and Social Media*.

Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Moshe Koppel and Yaron Winter. 2014. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65(1):178–187.

Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. 2016. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, pages 301–320. Springer.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for twitter in 140 characters or less. In *2010 Second Cybercrime and Trustworthy Computing Workshop*, pages 1–8. IEEE.

Rijul Magu, Kshitij Joshi, and Jiebo Luo. 2017. Detecting the hate code on social media. In *Eleventh International AAAI Conference on Web and Social Media*.

Todor Mihaylov and Preslav Nakov. 2016. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 399–405.

George K Mikros and Kostas Perifanos. 2013. Authorship attribution in greek tweets using author's multilevel n-gram profiles. In *2013 AAAI Spring Symposium Series*.

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.

Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J Niels Rosenquist. 2011. Understanding the demographics of twitter users. In *Fifth international AAAI conference on weblogs and social media*.

Antonio A Morgan-Lopez, Annice E Kim, Robert F Chew, and Paul Ruddle. 2017. Predicting age groups of twitter users based on language and metadata features. *PloS one*, 12(8):e0183537.

Edward Newell, David Jurgens, Haji Mohammad Saleem, Hardik Vala, Jad Sassine, Caitrin Armstrong, and Derek Ruths. 2016. User migration in online social networks: A case study on reddit during a period of community unrest. In *Tenth International AAAI Conference on Web and Social Media*.

Angela Orebaugh and Jeremy Allnutt. 2009. Classification of instant messaging communications for forensics analysis. *The International Journal of Forensic Computer Science*, 1:22–28.

Rebekah Overdorf and Rachel Greenstadt. 2016. Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution. *Proceedings on Privacy Enhancing Technologies*, 2016(3):155–171.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Maarten Sap, Gregory Park, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Lyle Ungar, and Hansen Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1146–1151.

Yunita Sari, Mark Stevenson, and Andreas Vlachos. 2018. Topic or style? exploring the most useful features for authorship attribution. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 343–353.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship

attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 669–674.

Thamar Solorio, Ragib Hasan, and Mainul Mizan. 2013. A case study of sockpuppet detection in wikipedia. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 59–68.

Thamar Solorio, Ragib Hasan, and Mainul Mizan. 2014. Sockpuppet detection in wikipedia: A corpus of real-world deceptive writing for linking identities. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *AAAI/IAAI*, pages 1058–1065.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556.

Efstathios Stamatatos, Francisco Rangel, Michael Tschuggnall, Benno Stein, Mike Kestemont, Paolo Rosso, and Martin Potthast. 2018. Overview of pan 2018. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 267–285. Springer.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Robin Thompson. 2011. Radicalization and the use of social media. *Journal of strategic security*, 4(4):167–190.

Michail Tsikerdekis and Sherali Zeadally. 2014. Multiple account identity deception detection in social media using nonverbal behavior. *IEEE Transactions on Information Forensics and Security*, 9(8):1311–1321.

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.

Luis Gerardo Mojica de la Vega and Vincent Ng. 2018. Modeling trolling in social media conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1041–1044, New York, NY, USA. ACM.

Isaac Waller and Ashton Anderson. 2019. Generalists and specialists: Using community embeddings to quantify activity diversity in online platforms. In *The World Wide Web Conference*, WWW '19, pages 1954–1964, New York, NY, USA. ACM.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393.

Yi Yang, Ming-Wei Chang, and Jacob Eisenstein. 2016. Toward socially-infused information extraction: Embedding authors, mentions, and entities. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1452–1461.

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.