TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations

Anonymous ACL submission

Abstract

We present TwHIN-BERT, a multilingual language model trained on in-domain data from the popular social network Twitter. TwHIN-BERT differs from prior pre-trained language models as it is trained with not only text-based self-supervision, but also with a social objective based on the rich social engagements within a Twitter heterogeneous information network (TwHIN). Our model is trained on 7 billion tweets covering over 100 distinct languages providing a valuable representation to model short, noisy, user-generated text. We evaluate our model on a variety of multilingual social recommendation and semantic understanding tasks and demonstrate significant metric improvement over established pre-trained language models. We will freely open-source TwHIN-BERT and our curated hashtag prediction and social engagement benchmark datasets to the research community.

1 Introduction

The proliferation of pre-trained language models (PLMs) (Devlin et al., 2019; Conneau et al., 2020) based on the Transformer architecture (Vaswani et al., 2017) has pushed the state of the art across many tasks in natural language processing (NLP). As an application of transfer learning, these models are typically trained on massive text corpora and, when fine-tuned on downstream tasks, have demonstrated state-of-the-art performance.

Despite the success of PLMs in general-domain NLP, fewer attempts have been made in language model pre-training for user-generated text on social media. In this work, we pre-train a language model for Twitter – a prominent social media platform where users post short messages called Tweets. Tweets contain informal diction, abbreviations, emojis, and topical tokens such as hashtags. As a result, PLMs designed for general text corpora may struggle to understand Tweet semantics accurately. Existing works (Nguyen et al., 2020; Barbieri et al., 2021) on Twitter LM pre-training do not address these challenges and simply replicate general domain pre-training on Twitter corpora.

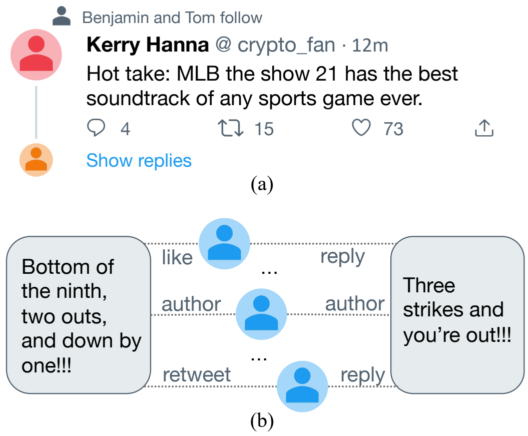

Figure 1: (a) This mock-up shows a short-text Tweet and social engagements such as Faves, Retweets, Replies, Follows that create a social context to Tweets and signify Tweet appeal to engaging users. (b) Co-engagement is a strong indicator of Tweet similarity.

A distinctive feature of Twitter social media is the user interactions through Tweet engagements. As seen in Figure 1, when a user visits Twitter, in addition to posting Tweets, they can perform a variety of *social* actions such as "Favoriting", "Replying" and "Retweeting" Tweets. The wealth of such engagement information is invaluable to Tweet content understanding. For example, the post "bottom of the ninth, two outs, and down by one!!" would be connected to baseball topics by its co-engaged Tweets, such as "three strikes and you're out!!!". Without the social contexts, a conventional text-only PLM objective would struggle to build this connection. As an additional benefit, a socially-enriched language model will also vastly benefit common applications on social media, such as social recommendations (Ying et al., 2018) and information diffusion prediction (Cheng et al., 2014; Sankar et al., 2020).

We introduce TwHIN-BERT– a multilingual language model for Twitter pre-trained with social engagements. The key idea of our method is to leverage *socially similar Tweets* for pre-training. Building on this idea, TwHIN-BERT has the following features. (1) We construct a Twitter Heterogeneous Information Network(TwHIN) (El-Kishky et al., 2022) to unify the multi-typed user engagement logs. Then, we run scalable embedding and approximate nearest neighbor search to sift through hundreds of billions of engagement records and mine socially similar Tweet pairs. (2) In conjunction with masked language modeling, we introduce a contrastive social objective that enforces the model to tell if a pair of Tweets are socially similar or not. Our model is trained on 7 billion Tweets from over 100 distinct languages, of which 1 billion have social engagement logs.

We evaluate the TwHIN-BERT model on both social recommendation and semantic understanding downstream evaluation tasks. To comprehensively evaluate on many languages, we curate two large-scale datasets, a social engagement prediction dataset focused on social aspects and a hashtag prediction dataset focused on language aspects. In addition to these two curated datasets, we also evaluate on established benchmark datasets to draw direct comparisons to other available pre-trained language models. TwHIN-BERT achieves state-of-the-art performance in our evaluations with a major advantage in the social tasks.

In summary, our contributions are as follows:

- We build the first ever socially-enriched pre-trained language model for noisy user-generated text on Twitter.
- Our model is the strongest multilingual Twitter PLM so far, covering 100 distinct languages.
- Our model has a major advantage in capturing social appeal of Tweets.
- We open-source TwHIN-BERT as well as two new Tweet benchmark datasets: (1) hashtag prediction and (2) social engagement prediction.

2 TwHIN-BERT

In this section, we outline how we construct training examples for our social objectives and subsequently train TwHIN-BERT with social and text pre-training objectives. As seen in Figure 2, we first construct and embed a user-Tweet engagement network. The resultant Tweet embeddings are then used to mine pairs of socially similar Tweets. These Tweet pairs and others are then used to pre-train TwHIN-BERT, which can then be fine-tuned for various downstream tasks.

2.1 Mining Socially Similar Tweets

With abundant social engagement logs, we (informally) define socially similar Tweets as *Tweets that are co-engaged by a similar set of users*. The challenge lies in how to implement this social similarity by (1) fusing heterogeneous engagement types, such as “Favorite”, “Reply” , “Retweet”, and (2) efficiently mining billions of similar Tweet pairs.

To address these challenges, TwHIN-BERT first constructs a **T**witter **H**eterogeneous **I**nformation **N**etwork (TwHIN) from the engagement logs, then runs a scalable heterogeneous network embedding method to capture co-engagement and map Tweets and users into a vector space. With this, social similarity translates to embedding space similarity. Subsequently, we mine similar Tweet pairs via ANN search on the Tweet embeddings.

2.1.1 Constructing TwHIN

We define and construct TwHIN as follows:

Definition 1 (TwHIN) *Our Twitter Heterogeneous Information Network is a directed bipartite graph $G = (U, T, E, \phi)$, where U is the set of user nodes, T is the set of Tweet nodes, $E = U \times T$ is the set of engagement edges. $\phi : E \mapsto \mathcal{R}$ is an edge type mapping function. Each edge $e \in E$ belongs to a type of engagement in $\mathcal{R}$.*

Our curated TwHIN (Figure 3) consists of approximately 200 million distinct users, 1 billion Tweets, and over 100 billion edges. We posit that our TwHIN encodes not only user preferences but also Tweet social appeal. We perform scalable network embedding to derive a social similarity metric from TwHIN. The network embedding fuses the heterogeneous engagements into a unified vector space that’s easy to operate on.

2.1.2 Embedding TwHIN Nodes

While our approach is agnostic to the exact methodology used to embed TwHIN, we follow the approach outlined in (El-Kishky et al., 2022; El-Kishky et al., 2022b). We perform training with the TransE embedding (Bordes et al., 2013) objective to co-embed users and Tweets using the PyTorch-Biggraph (Lerer et al., 2019) framework for scalability. Following previous approaches, we train for 10 epochs and perform negative sampling

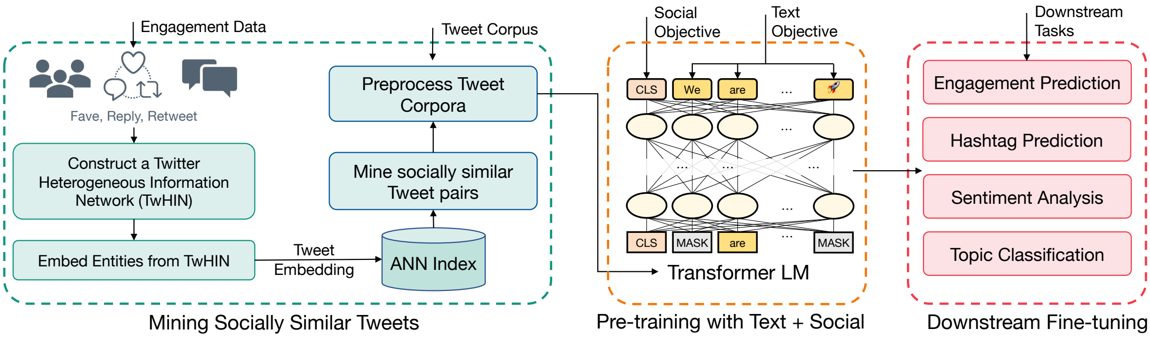

Figure 2: We outline the end-to-end TwHIN-BERT process. This three-step process involves (1) mining socially similar Tweet pairs by embedding a Twitter Heterogeneous Information Network (2) training TwHIN-BERT using a joint social and MLM objective and finally (3) fine-tuning TwHIN-BERT on downstream tasks.

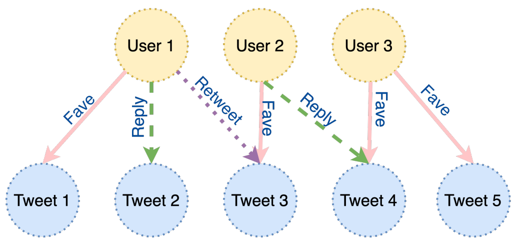

Figure 3: Twitter Heterogeneous Information Network (TwHIN) capturing social engagements between users and Tweets.

both uniformly and proportional to entity prevalence in TwHIN (Bordes et al., 2013; Lerer et al., 2019). Optimization is performed via Adagrad.

Upon learning dense representations of nodes in TwHIN, we utilize the learned Tweet representations to mine socially similar Tweets.

2.1.3 Mining Similar Tweet Pairs

Given the learned TwHIN Tweet embeddings, we seek to identify pairs of Tweets with *similar social appeal* – that is, Tweets that appeal to (i.e., are likely to be engaged with) similar users. We will use these socially-similar Tweet pairs as self-supervision when training TwHIN-BERT. To identify these pairs, we perform an approximate nearest neighbor (ANN) search in the TwHIN embedding space. To efficiently perform the search over 1B+ Tweets, we use the optimized FAISS[1] toolkit (Johnson et al., 2019) to create a compact index of Tweets keyed by their engagement-based TwHIN embeddings. As each Tweet embedding is 256-dimensional, storing billion-scale Tweet embeddings would require more than one TB of memory. To reduce the size of the index such that it can fit on a 16 A100 GPU node with each GPU possessing 40GB of memory, we apply product quantization (Jegou et al., 2010) to discretize and reduce embeddings size. The resultant index corresponds to `OPQ64,IVF65536,PQ64` in the FAISS index factory terminology.

After creating the FAISS index and populating it with TwHIN Tweet embeddings, we search the index using Tweet embedding queries to find pairs of similar Tweets (t_i, t_j) such that $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are close in the embedding space as defined by their cosine distance. To ensure high recall, we query the FAISS index with 2000 probes. Finally, we select the k closet Tweets defined by the cosine distance between the query Tweet and retrieved Tweets' embeddings. These Tweet pairs are used in our social-objective when pre-training TwHIN-BERT.

2.2 Pre-training Objectives

Given the mined *socially similar* Tweets, we describe our language model training process. To train TwHIN-BERT, we first run the Tweets through the language model and then train the model with a joint contrastive social loss and masked language model loss.

Tweet Encoding with LM. We use a Transformer language model to encode each Tweet. Similar to BERT (Devlin et al., 2019), given the tokenized text $\boldsymbol{w}_t = [w_1, w_2, ..., w_n]$ of a Tweet t, we add special tokens to mark the start and end of the Tweet: $\hat{\boldsymbol{w}}_t = [\text{CLS}]\boldsymbol{w}_t[\text{SEP}]$. As the Tweets are usually shorter than the maximum sequence length of a language model, we group multiple Tweets and feed them together into the language model when possible. We then apply *CLS-pooling*, which takes the [CLS] token embedding of each Tweet. These Tweet embeddings are passed through an MLP pro-

[1] `https://github.com/facebookresearch/faiss`

jection head for the *social loss* computation.

$$[\boldsymbol{e}_{t_1}, \boldsymbol{e}_{t_2}, ...] = \text{Pool}\left(\text{LM}([\hat{\boldsymbol{w}}_{t_1}, \hat{\boldsymbol{w}}_{t_2}, ...])\right) \quad (1)$$

$$\boldsymbol{z}_t = \text{MLP}(\boldsymbol{e}_t) \quad (2)$$

Contrastive Social Loss. We use a contrastive loss to let our model learn whether two Tweets are socially similar or not. For each batch of B socially similar Tweet pairs $\{(t_i, t_j)\}_B$, we compute the *NT-Xent* loss (Chen et al., 2020) with in-batch negatives:

$$\mathcal{L}_{\text{social}}(i,j) = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j))/\tau}{\sum_{\mathcal{N}_B(i)} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)} \quad (3)$$

The negatives $\mathcal{N}_B(i)$ of Tweet t_i are the $(2B-1)$ other Tweets in the batch that are not paired with t_i. We use cosine similarity for function $\text{sim}(\cdot,\cdot)$. τ is the loss temperature.

Our overall pre-training objective is a combination of the contrastive social loss and the masked language model loss (Devlin et al., 2019):

$$\mathcal{L} = \mathcal{L}_{\text{social}} + \lambda \mathcal{L}_{\text{MLM}} \quad (4)$$

λ is a hyperparameter that balances the social and language loss.

2.3 Pre-training Setup

Model Architecture. We use the same Transformer architecture as BERT (Devlin et al., 2019) for our language model. We adopt the XLM-R (Conneau et al., 2020) tokenizer, which offers good capacity and coverage in all languages. The model has a vocabulary size of 250K. The max sequence length is set to 128 tokens. The detailed model setup can be found in Appendix B. Note that although we have chosen this specific architecture, our social objective can be used in conjunction with a wide range of language model architectures.

Pre-training Data. We collect 7 billion Tweets in 100 languages from Jan. 2020 to Jun. 2022. Additionally, we collect 100 billion user-Tweet social engagement data covering 1 billion of our Tweets. We re-sample the data based on language frequency raised to the power of 0.7 to mitigate under-representation of low-resource languages.

Training Procedure. Our training has two stages. In the first stage, we train the model from scratch using the 6 billion Tweets without user engagement. The model is trained for 500K steps on 16 Nvidia A100 GPUs (a2-megagpu-16g) with a total batch size of 6K. In the second stage, the model is trained for another 500K steps on the 1 billion Tweets with the joint MLM and social loss. We use mixed precision during training. Overall pre-training takes approximately five days for the base model and two weeks for the large model. We refer readers to Appendix B for the detailed hyperparameter setup.

3 Experiments

In this section, we discuss baseline model specifications, evaluation setup, and results from two families of downstream evaluation tasks.

3.1 Evaluated Methods

We evaluate TwHIN-BERT against the following baselines. **mBERT** (Devlin et al., 2019) and **XLM-R** (Conneau et al., 2020) are two popular general domain multilingual language models trained with gigantic datasets. **BERTweet** (Nguyen et al., 2020) is the previous state-of-the-art English Tweet language model. **XLM-T** (Barbieri et al., 2021) is a multilingual Twitter language model based on XLM-R (Conneau et al., 2020).

We include three variations of our model trained on the same corpus: *base* and *large* sizes, and an ablated *base-MLM* trained with only an MLM objective. All baselines are *base* variants (with between 135M to 278M parameters depending on the size of the tokenizer). Our large model has around 550M parameters.

3.2 Social Engagement Prediction

Our first benchmark task is *social engagement prediction*. This task aims to evaluate how well the pre-trained language models capture social aspects of user-generated text. In our task, we predict whether users modeled via a user embedding vector will perform a certain social engagement on a given Tweet.

We use different pre-trained language models to generate representations for Tweets, and then feed these representations into a simple prediction model alongside the corresponding user representation. The model is trained to predict whether a user will engage with a specific Tweet.

Dataset. To curate our Tweet-Engagement dataset, we select the 50 popular languages on Twitter and sample 10,000 (or all if the total number is less than 10,000) Tweets of each language from a fixed time period. All Tweets are available via the Twitter public API. We then collect the user-Tweet engage-

Table 1: Engagement prediction HITS@10 on high, mid, low-resource, and average of all languages.

	High-Resource			Mid-Resource			Low-Resource			All
Method	**en**	**ja**	**ar**	**el**	**ur**	**nl**	**no**	**da**	**ps**	**Avg.**
mBERT	.0633	.0227	.0532	.0496	.0437	.0616	.0731	.1060	.0522	.0732
XLM-R	.0850	.0947	.0546	.0628	.0315	.0650	.1661	.1150	.0727	.0849
XLM-T	.1181	.1079	.1403	.0562	.0352	.0762	.1156	.1167	.0662	.1043
TwHIN-BERT										
- Base-MLM	.1400	.1413	.1640	.0801	.0547	.0965	.1502	.1334	.0600	.1161
- Base	.1552	.2065	**.2206**	.0944	.0627	**.1346**	.1920	.1470	.0799	.1436
- Large	**.1585**	**.2325**	.1989	**.1065**	**.0667**	.1248	**.2118**	**.1475**	**.0817**	**.1497**

ment records associated with these Tweets. There are, on average, 29K engagement records per language. We ensure that there is no overlap between the evaluation and pre-training datasets.

Each engagement record consists of a pre-trained 256-dimensional user embedding (El-Kishky et al., 2022) and a Tweet ID that indicates the user has engaged with the given Tweet. To ensure privacy, each user embedding appears only once, however each tweet may be engaged by multiple users. We split the Tweets into train, development, and test sets with a 0.8/0.1/0.1 ratio, and then collect the respective engagement records for each subset.

Prediction Model. Given a pre-trained language model, we use it to generate an embedding for each Tweet t given its content $\boldsymbol{w}_t$: $\boldsymbol{e}_t = \text{Pool}\left(\text{LM}(\boldsymbol{w}_t)\right)$

We apply the following pooling strategies to calculate the Tweet embedding from the language model. First, we take [CLS] token embedding as the first part of overall embedding. Then, we take the average token embedding of non-special tokens as the second part. The two parts are concatenated to form the *Combined* embedding of a Tweet.

With LM-derived Tweet embeddings, pre-trained user embeddings, and the user-Tweet engagement records, we build an engagement prediction model $\boldsymbol{\Theta} = (\boldsymbol{W}_t, \boldsymbol{W}_u)$. Given a user u and a Tweet t, the model projects the user embedding $\boldsymbol{e}_u$ and the Tweet embedding $\boldsymbol{e}_t$ into the same space, and then calculates the probability of engagement:

$$\boldsymbol{h}_u = \boldsymbol{W}_u^T \boldsymbol{e}_u, \quad \boldsymbol{h}_t = \boldsymbol{W}_t^T \boldsymbol{e}_t$$

$$P(t \mid u) = \sigma\left(\boldsymbol{h}_u^T \boldsymbol{h}_t\right)$$

We optimize a negative sampling loss on the training engagement records R. For each engagement pair $(u, t) \in R$, the loss is defined as:

$$\log \sigma\left(\boldsymbol{h}_u^T \boldsymbol{h}_t\right) + \mathbb{E}_{t' \sim P_n(R)} \log \sigma\left(-\boldsymbol{h}_u^T \boldsymbol{h}_{t'}\right)$$

where $P_n(R)$ is a negative sampling distribution. We use the frequency of each Tweet in R raised to the power of $3/4$ for this distribution.

Our prediction model closely resembles classical link prediction models such as (Tang et al., 2015b). We keep the model simple, making sure it will not overpower the language model embeddings.

Evaluation Setup and Metrics. We conduct hyperparameter search on the English development dataset and use these hyperparameters for the other languages. The prediction model projects user and Tweet embedding to 128 dimensions. We set batch size to 512, learning rate to 1e-3. The best model on validation set is selected for test set evaluation.

In the test set, we pair each user with 1,000 Tweets: one Tweet they have engaged with and the rest are randomly sampled negatives. The model ranks the Tweets by the predicted probability of engagement, and we evaluate with HITS@10. We report median results from 6 runs with different initialization.

Results. We show results for high, mid, and low-resource languages (determined by language frequency on Twitter) in Table 1. Language abbreviations are ISO language codes[2]. We also show the average results from all 50 languages in the evaluation dataset, and leave the details in Appendix D. Our TwHIN-BERT model demonstrates significant improvement over the baselines on the social engagement task. Comparing our model to the ablation without the social loss, we can see the contrastive social pre-training provides significant lift over just MLM pre-training for social engagement

[2]https://www.iso.org/iso-639-language-codes.html

Table 2: Text classification dataset statistics. *Statistics for Hashtag shows the numbers for each language.

Dataset	Lang.	Label	Train	Dev	Test
SE2017	en	3	45,389	2,000	11,906
SE2018-en	en	20	45,000	5,000	50,000
SE2018-es	es	19	96,142	2,726	9,969
ASAD	ar	3	137,432	15,153	16,842
COVID-JA	ja	6	147,806	16,394	16,394
SE2020-hi	hi+en	3	14,000	3,000	3,000
SE2020-es	es+en	3	10,800	1,200	3,000
Hashtag	multi	500*	16,000*	2,000*	2,000*

prediction. An analysis on all 50 evaluation languages shows the *large* model to perform better than the *base* model on average, with more wins than losses. Additionally, we also observe that our method yields the most improvement when using the *Combined* [CLS] token and average non-special token embedding. We believe the [CLS] token embedding from our model captures social aspects of the Tweet, while averaging the other token embeddings captures the semantic aspects of the Tweet. Naturally, utilizing both aspects is essential to better model a Tweet's appeal and a user's inclination to engage with a Tweet.

3.3 Tweet Classification

Our second collection of downstream tasks is Tweet classification. In these tasks, we take as input the Tweet text and predict discrete labels corresponding to the label space for each task.

Datasets. We curate a multilingual Tweet hashtag prediction dataset (available via Twitter public API) to comprehensively cover the popular languages on Twitter. In addition, we evaluate on five external benchmark datasets for tasks such as sentiment classification, emoji prediction, and topic classification in selected languages. We show the dataset statistics in Table 2.

- **Tweet Hashtag Prediction** dataset is a multilingual hashtag prediction dataset we collected from Tweets. It contains Tweets of 50 popular languages on Twitter. For each language, 500 most popular hashtags were selected and 100k Tweets that has those hashtags were sampled. We made sure each Tweet will only contain one of the 500 candidate hashtags. Similar to work proposed in Mireshghallah et al. (2022), the task is to predict the hashtag used in the Tweet.
- **Sentiment Analysis**. The English dataset **SemEval2017** task 4A (Rosenthal et al., 2019), Arabic dataset **ASAD** (Alharbi et al., 2020), code mixed Hindi/Spanish+English datasets **SemEval2020** task 9 (Patwa et al., 2020) are three-point sentiment analysis tasks with labels of "positive", "negative", "neutral".
- **Emoji Prediction**. **SemEval2018** task 2 (Barbieri et al., 2018) is an emoji prediction dataset in both English and Spanish. The objective is to predict the most likely used emoji in a Tweet.
- **Topic Classification**. **COVID-JA** (Suzuki, 2019) is a Japanese Tweets classification dataset. The objective is to classify each Tweet into one of the six pre-defined topics around COVID-19.

Setup and Evaluation Metrics. We use the standard language model fine-tuning method as described in (Devlin et al., 2019) and apply a linear prediction layer on top of the pooled output of the last transformer layer. Each model is fine-tuned for up to 30 epochs, and we evaluate the best model from the training epochs on the test set based on development set performance. The fine-tuning hyperparameter setup can be found in Appendix B. We report the median results from 3 fine-tuning runs with different random seeds. Results are the evaluation metrics recommended for each benchmark dataset or challenge (Appendix C). For hashtag prediction datasets, we report macro-F1 scores.

Multilingual Hashtag Prediction. In Table 3, we show macro F1 scores on selected languages from our multilingual hashtag prediction dataset. We also report the average performance of all 50 languages in the dataset, and leave detailed results in Appendix E. We can see that TwHIN-BERT significantly outperforms the baseline methods at the same *base* size. Our *large* model is slightly better than or on par with the *base* model, with a better overall performance. On the English dataset, our model outperforms the BERTweet monolingual language model trained exclusively on English Tweets and with a dedicated English tokenizer. Comparing our model to the ablation with no social loss, the two models demonstrate similar performance with our model being slightly better. These results show that while our model has a major advantage on social tasks, it retains high performance on semantic understanding applications.

External Classification Benchmarks. As shown in Table 4, TwHIN-BERT matches or outperforms the multilingual baselines on the established classification benchmarks. BERTweet fares better than

Table 3: Multilingual hashtag prediction Macro-F1 on high, mid, low resource, and average of all languages.

	High-Resource			Mid-Resource			Low-Resource			All
Method	**en**	**ja**	**ar**	**el**	**ur**	**nl**	**no**	**da**	**ps**	**Avg.**
BERTweet	59.01	-	-	-	-	-	-	-	-	-
mBERT	54.56	68.43	38.48	44.00	36.44	39.75	46.09	59.54	29.41	50.05
XLM-R	53.90	69.07	37.85	43.94	37.56	40.85	48.94	60.35	34.92	50.86
XLM-T	55.08	70.55	42.27	44.15	39.22	41.01	49.22	59.97	33.27	51.74
TwHIN-BERT										
- Base-MLM	58.38	72.66	43.08	46.89	41.53	42.36	49.60	61.00	35.37	53.66
- Base	59.31	**73.03**	44.24	**47.59**	42.81	42.69	51.11	60.33	36.21	54.62
- Large	**60.07**	72.91	**45.41**	47.43	**43.39**	**44.80**	**51.34**	**61.56**	**38.24**	**55.23**

Table 4: External classification benchmark results.

	SE2017	SE2018		ASAD	COVID-JA	SE2020		Avg.
Method	**en**	**en**	**es**	**ar**	**ja**	**hi+en**	**es+en**	
BERTweet	72.97	33.27	-	-	-	-	-	-
mBERT	66.17	27.73	19.19	69.08	80.57	66.55	45.31	53.51
XLM-R	71.15	30.94	21.05	79.09	81.67	69.59	48.97	57.49
XLM-T	72.01	31.97	21.49	80.70	81.48	70.94	51.06	58.52
TwHIN-BERT								
- **Base-MLM**	72.10	32.44	21.79	80.48	82.12	72.42	51.67	59.00
- **Base**	72.30	32.41	22.23	80.73	82.37	71.30	54.32	59.38
- **Large**	**73.10**	**33.31**	**22.80**	**81.19**	**82.50**	**73.08**	**54.47**	**60.06**

our *base* model with its dedicated large English tokenizer and monolingual training. Our *large* model outperforms all the baselines. Similar to hashtag prediction, TwHIN-BERT performs on par with or slightly better than the MLM-only ablation.

3.4 Varying Downstream Supervision

In this set of experiments, we study how TwHIN-BERT performs when the amount of downstream supervision changes. We fine-tune our model and baseline models on the hashtag prediction dataset (Section 3.3). We select English and Japanese as they are the most popular languages on Twitter. We change the number of training examples given to the models during fine-tuning. It is varied from 2 to 32 labeled training examples per class. We follow the same protocols as Section 3.3 and report macro F1 scores on the test set.

Figure 4 shows the results. TwHIN-BERT holds significant performance gain across different amount of downstream supervision. Note that when supervision is scarce, e.g., two labeled training examples per class given, our model has a even larger relative performance improvement over the baselines. The results indicate that our model may empower weakly supervised applications on Tweet natural language understanding.

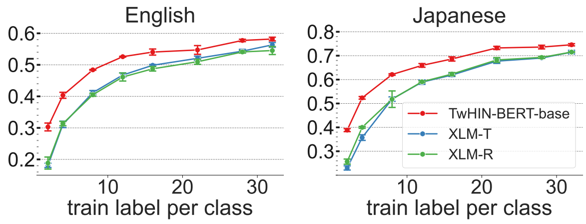

Figure 4: Macro-F1 score on English and Japanese hashtag prediction datasets w.r.t. number of labeled training examples per class.

3.5 Feature-based Classification

In addition to language model fine-tuning experiments, we evaluate TwHIN-BERT's performance as a feature extractor. We use the hashtag prediction datasets (Section 3.3) and select three popular languages with different scripts. We use our model and the baseline models to embed each Tweet into a feature vector and train a Logistic Regression classifier with the fixed feature vectors as input.

Table 5: Feature-based classification on hashtag prediction datasets (Macro-F1).

Method	en	ja	ar
XLM-R	30.88	41.14	21.55
XLM-T	41.66	51.56	32.46
TwHIN-BERT-base	51.16	**64.12**	37.20
TwHIN-BERT-large	**54.12**	64.03	**38.78**

Table 5 shows the results of our feature-based classification experiments. TwHIN-BERT outperforms the baselines with a wide margin on all languages. The results demonstrate a clear advantage of our model for unsupervised Tweet representations, and shows its potential in other feature-based downstream applications.

4 Related Works

Pre-trained Language Models: Since their introduction (Peters et al., 2018; Devlin et al., 2019), pre-trained language models have enjoyed tremendous success in all aspects of natural language processing. Follow up research has advanced PLMs by scaling them with respect to number of parameters and training data (Micheli and Fleuret, 2021; Raffel et al., 2020; Shoeybi et al., 2019), and by improving the training objectives (Yang et al., 2019; Clark et al., 2020; Meng et al., 2021). Despite these innovations in scaling and pre-training objectives, the vast majority of the work has focused on text-only training objectives applied to general domain corpora. In this paper, we deviate from most previous works by explore PLM training using solely Twitter in-domain data and training our model based on both text-based and social-based objectives.

Tweet Language Models: While a majority of PLMs are trained on general domain corpora, a few language models have been proposed specifically for Twitter and other social media platforms. BERTweet (Nguyen et al., 2020) mirrors BERT training on 850 million English Tweets. TimeLMs (Loureiro et al., 2022) trains a set of RoBERTa (Liu et al., 2019) models for English Tweets on different time ranges. XLM-T (Barbieri et al., 2021) continues the pre-training process from an XLM-R (Conneau et al., 2020) checkpoint on 198 million multilingual Tweets. These methods mostly replicate existing general domain PLM methods and simply substitute the training data with Tweets. However, our approach utilizes additional social engagement signals to enhance the pre-trained Tweet representations.

Enriching PLMs with Additional Information: Several existing works introduce additional information for language model pre-training. ERNIE (Zhang et al., 2019) and K-BERT (Liu et al., 2020) inject entities and their relations from knowledge graphs to augment the pre-training corpus. OAG-BERT (Liu et al., 2022) appends metadata of a document to its raw text, and designs objectives to jointly predict text and metadata. These works focus on bringing additional metadata and knowledge by injecting training instances, while our work leverage the rich social engagements embedded in the social media platform for text relevance. Recent work (Yasunaga et al., 2022) has utilized document hyperlinks for LM pre-training, but does so with a simple three way classification objective.

Network Embedding: Network embedding has emerged as a valuable tool for transferring information from relational data to other tasks (El-Kishky et al., 2022a). With the introduction of heterogeneous information networks (Sun and Han, 2013) as a formalism to model rich multi-typed, multi-relational networks, many heterogeneous network embedding approaches were developed (Chang et al., 2015; Tang et al., 2015a; Xu et al., 2017; Chen and Sun, 2017; Dong et al., 2017). However, many of these techniques are difficult to scale to very large networks. In this work, we apply knowledge graph embeddings (Wang et al., 2017; Bordes et al., 2013; Trouillon et al., 2016), which have been shown to be both highly scalable and flexible enough to model multiple node and edge types.

5 Conclusions

In this work we introduce TwHIN-BERT, a multilingual BERT-style language model trained on a large Tweet corpus. Unlike previous BERT-style language models, TwHIN-BERT is trained using two objectives: (1) a standard MLM pre-training objective and (2) a contrasting social objective. We utilize our pre-trained language model to perform a variety of downstream tasks on Tweet data. Our experiments demonstrate that TwHIN-BERT outperforms previously released language models on not only semantic tasks, but also on social engagement prediction tasks. We release this model to the academic community to further research in social media NLP.

References

Basma Alharbi, Hind Alamro, Manal Abdulaziz Alshehri, Zuhair Khayyat, Manal Kalkatawi, Inji Ibrahim Jaber, and Xiangliang Zhang. 2020. Asad: A twitter-based benchmark arabic sentiment analysis dataset. *ArXiv*, abs/2011.00578.

Francesco Barbieri, Luis Espinosa Anke, and José Camacho-Collados. 2021. Xlm-t: A multilingual language model toolkit for twitter. *ArXiv*, abs/2104.12250.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

S. Chang, W. Han, J. Tang, G. Qi, C. Aggarwal, and T. Huang. 2015. Heterogeneous network embedding via deep architectures. In *SIGKDD*, pages 119–128.

T. Chen and Y. Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*, pages 295–304.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Justin Cheng, Lada A. Adamic, P. Alex Dow, Jon M. Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted? In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 925–936. ACM.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Y. Dong, N. Chawla, and A. Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*, pages 135–144.

Ahmed El-Kishky, Michael Bronstein, Ying Xiao, and Aria Haghighi. 2022a. Graph-based representation learning for web-scale recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4784–4785.

Ahmed El-Kishky, Thomas Markovich, Kenny Leung, Frank Portman, and Aria Haghighi. 2022b. knn-embed: Locally smoothed embedding mixtures for multi-interest candidate retrieval. *arXiv preprint arXiv:2205.06205*.

Ahmed El-Kishky, Thomas Markovich, Serim Park, Chetan Verma, Baekjin Kim, Ramy Eskander, Yury Malkov, Frank Portman, Sofía Samaniego, Ying Xiao, and Aria Haghighi. 2022. Twhin: Embedding the twitter heterogeneous information network for personalized recommendation. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 2842–2850. ACM.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*.

Xiao Liu, Da Yin, Jingnan Zheng, Xingjian Zhang, P. Zhang, Hongxia Yang, Yuxiao Dong, and Jie Tang. 2022. Oag-bert: Towards a unified backbone language model for academic knowledge services. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

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. *CoRR*, abs/1907.11692.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and José Camacho-Collados. 2022. Timelms: Diachronic language models from twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 251–260. Association for Computational Linguistics.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: correcting and contrasting text sequences for language model pretraining. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23102–23114.

Vincent Micheli and François Fleuret. 2021. Language models are few-shot butlers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9312–9318. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Nikolai Vogler, Junxian He, Omar Florez, Ahmed El-Kishky, and Taylor Berg-Kirkpatrick. 2022. Non-parametric temporal adaptation for social media topic classification. *arXiv preprint arXiv:2209.05706*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 9–14. Association for Computational Linguistics.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING 2020, Barcelona (online), December 12-13, 2020*, pages 774–790. International Committee for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019. Semeval-2017 task 4: Sentiment analysis in twitter. *CoRR*, abs/1912.00741.

Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 510–518. ACM.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053.

Y. Sun and J. Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter*.

Yu Suzuki. 2019. Filtering method for twitter streaming data using human-in-the-loop machine learning. *J. Inf. Process.*, 27:404–410.

J. Tang, M. Qu, and Q. Mei. 2015a. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD*, pages 1165–1174.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015b. LINE: large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1067–1077. ACM.

T. Trouillon, J Welbl, S. Riedel, É. Gaussier, and G. Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *TKDE*, 29(12):2724–2743.

L. Xu, X. Wei, J. Cao, and P. Yu. 2017. Embedding of embedding (eoe) joint embedding for coupled heterogeneous networks. In *WSDM*, pages 741–749.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 974–983. ACM.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.

A Distribution of Languages in Training Dataset

Figure 5 shows the distribution of languages in our pre-training dataset. Some languages with different variations (e.g., Hindi and Hindi Romanized) are represented with the same ISO language code. We run fastText (Bojanowski et al., 2017) language identification model `lid.176.bin`[3] to detect languages.

We deem a language "high-resource" if we have more than 10^8 Tweets during pre-training *after* frequency-based re-sampling (Section 2.3); "mid-resource" if we have more than 10^7 and less than 10^8 Tweets; "low-resource" if we have less than 10^7 Tweets.

B Hyperparameters for Pre-training and Fine-Tuning

Table 6 shows the pre-training hyperparameters. The model architecture and hyperparameters not shown in the table are the same as RoBERTa (Liu et al., 2019).

Table 7 shows the hyperparameters for classification fine-tuning. We do hyperparameter selection on the development datasets and share the same set of hyperparameters for the base models, as we find them to perform well with this setting. The weight decay for base models is set to zero. A different set of hyperparameters were necessary for the large model because it behaves differently from the base models in terms of convergence.

C Evaluation Metrics for External Classification Benchmarks

The recommended evaluation metrics that we report in Table 4 are as follows. Average recall for ASAD, SemEval 2017 datasets; Macro-F1 for SemEval 2018 English and Spanish datasets; Accuracy for COVID-JA, SemEval 2020 datasets.

D Engagement Prediction Results on Additional Languages

Table 8 shows the engagement prediction results on all available evaluation languages. Some languages have more examples than other languages due to data availability.

E Hashtag Prediction Results on Additional Languages

Table 9 shows the hashtag prediction results on all available evaluation languages. A small number of languages have less examples than shown in Table 2 due to data availability. The Russian language is not evaluated as the XLM-T baseline fails on some Russian characters in our dataset.

[3] `https://fasttext.cc/docs/en/language-identification.html`

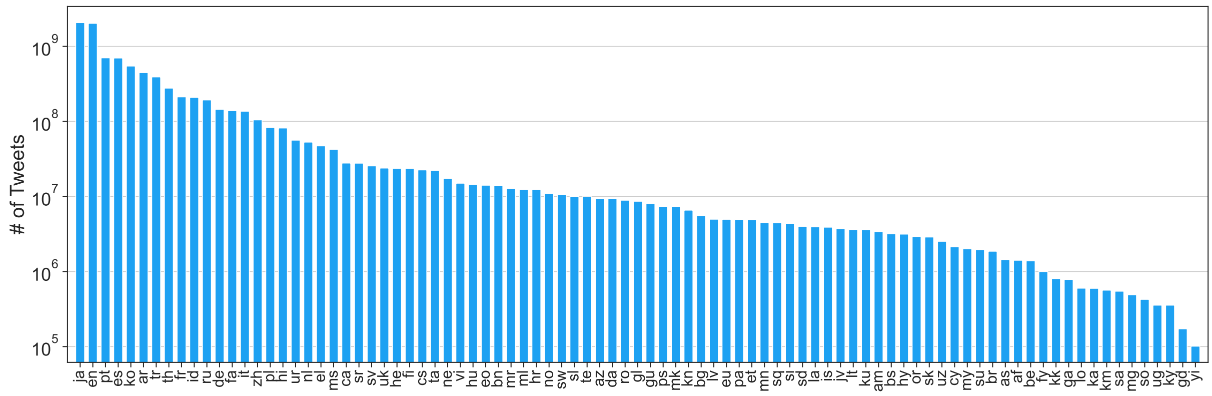

Figure 5: The number of Tweets in the pre-training dataset for each language. Languages are marked by ISO language codes.

Table 6: Hyperparameters for pre-training TwHIN-BERT.

Hyperparameter	**TwHIN-BERT-base**	**TwHIN-BERT-large**
Max sequence length	128	128
Precision	BF16	BF16
Stage 1: MLM		
Total batch size	6K	8K
Gradient accumulation steps	1	4
Peak learning rate	2e-4	2e-4
Warmup steps	30K	30K
Total steps	500K	500K
Stage 2: MLM + Social		
Total batch size	6K	6K
Gradient checkpointing	No	Yes
Peak learning rate	1e-4	1e-4
Warmup steps	30K	30K
Total steps	500K	500K
Contrastive projection head	[768, 768]	[1024, 512]
Contrastive loss temperature	0.1	0.1
Loss balancing λ	0.05	0.05

Table 7: Hyperparameters for fine-tuning TwHIN-BERT and the baselines for classification.

Hyperparameter	**Hashtag**	**SE2017**	**SE2018**	**ASAD**	**COVID-JA**	**SE2020**
Base models						
Learning rate	4e-5	4e-5	1e-5	1e-5	2e-5	2e-5
Batch size	128	128	128	128	128	128
TwHIN-BERT-large						
Learning rate	2e-5	2e-5	1e-5	1e-5	1e-5	1e-5
Weight decay	0	0	5e-4	5e-4	0	5e-4
Batch size	128	128	128	128	128	128

Table 8: Social engagement prediction results (HITS@10) on all evaluation Languages.

				TwHIN-BERT		
Language	**mBERT**	**XLM-R**	**XLM-T**	**Base-MLM**	**Base**	**Large**
English (en)	.0633	.0850	.1181	.1400	.1552	**.1585**
Japanese (ja)	.0227	.0947	.1079	.1413	.2065	**.2325**
Turkish (tr)	.0348	.0476	.1180	**.1268**	.1204	.0547
Spanish (es)	.0575	.0704	.1103	.1204	.1618	**.2055**
Arabic (ar)	.0532	.0546	.1403	.1640	**.2206**	.1989
Portuguese (pt)	.0731	.1285	.1709	.1201	**.1924**	.1915
Persian (fa)	.0556	.1621	.1754	.1903	.2065	**.2097**
Korean (ko)	.0275	.1105	.1446	.1675	.3611	**.3714**
French (fr)	.0488	.0635	.0805	.0700	.1030	**.1053**
Russian (ru)	.0889	.1482	.1530	.0990	**.1726**	.1704
German (de)	.0852	.1071	.3019	.2189	**.3020**	.2621
Thai (th)	.0659	.1027	.1056	.1196	**.2083**	.2004
Italian (it)	.0586	.0769	.1237	.1478	.1699	**.1706**
Hindi (hi)	.0870	.0838	.1140	.1054	.1737	**.1751**
Indonesian (id)	.0809	.0735	.0921	.1014	.1021	**.1115**
Polish (pl)	.0867	.0835	.1031	.1402	**.1696**	.1633
Urdu (ur)	.0437	.0315	.0352	.0547	.0627	**.0667**
Filipino (tl)	.0610	.0653	.0877	.1045	.1332	**.1400**
Egyptian Arabic (arz)	.0669	.0749	.1049	.0943	**.1159**	.1122
Greek (el)	.0496	.0628	.0562	.0801	.0944	**.1065**
Serbian (sr)	.1013	.1144	.1359	.1394	**.1647**	.1556
Dutch (nl)	.0616	.0650	.0762	.0965	**.1346**	.1248
Hebrew (he)	.0392	.0433	.0441	.0499	.0550	**.0577**
Ukrainian (uk)	.0497	**.0842**	.0669	.0711	.0811	**.0842**
Catalan (ca)	.1339	.1364	.1650	.1930	**.1955**	.1713
Swedish (sv)	.0942	.0716	.1161	.1342	**.1467**	.1462
Tamil (ta)	.0556	.0691	.0929	.1005	.1037	**.1060**
Finnish (fi)	.0876	.1067	.1317	.1529	.1710	**.1809**
Czech (cs)	.1155	.0904	.0766	.0997	.1062	**.1308**
Nepali (ne)	.0421	.0555	.0486	.0589	.0787	**.0851**
Azerbaijani (az)	.1561	.1148	.1702	.1576	.1712	**.1839**
Marathi (mr)	.0506	.0600	.0519	.0597	.0780	**.0906**
Bangla (bn)	.1361	.1350	.1320	.1601	.1649	**.1675**
Norwegian (no)	.0731	.1661	.1156	.1502	.1920	**.2118**
Telugu (te)	.0279	.0505	.0728	.0883	.1017	**.1654**
Pashto (ps)	.0522	.0727	.0662	.0600	.0799	**.0817**
Danish (da)	.1060	.1150	.1167	.1334	.1470	**.1475**
Vietnamese (vi)	.0929	.1060	.1085	.1216	.1417	**.1809**
Central Kurdish (ckb)	.0725	.0699	.0946	.1023	.1023	**.1185**
Gujarati (gu)	.0666	.0676	.0676	.0793	.1054	**.1057**
Macedonian (mk)	.0685	.0945	.0534	.0973	**.1089**	.1041
Cebuano (ceb)	.1222	.1267	.1767	.1900	.2003	**.2334**
Romanian (ro)	.1718	.1493	.1991	.2071	**.2264**	**.2264**
Kannada (kn)	.0552	.1355	.0814	.1098	.1282	**.2113**
Latvian (lv)	.0480	.0297	.0493	.0642	.0655	**.0750**
Bulgarian (bg)	.1953	.0448	.0702	.1790	.2248	**.2269**
Sinhala (si)	.0504	.0142	.0378	.0630	**.0709**	.0661
Icelandic (is)	.0319	.0341	.0466	.0364	.0387	**.0603**
Sindhi (sd)	.0619	.0288	.0553	.0885	.0951	**.0973**
Amharic (am)	.0293	.0663	.0491	.0543	.0698	**.0818**
Average	.0732	.0849	.1043	.1161	.1436	**.1497**

Table 9: Hashtag prediction results (Macro-F1) on all evaluation languages.

				TwHIN-BERT		
Language	**mBERT**	**XLM-R**	**XLM-T**	**Base-MLM**	**Base**	**Large**
English (en)	54.56	53.90	55.08	58.38	59.31	**60.07**
Japanese (ja)	68.43	69.07	70.55	72.66	**73.03**	72.91
Turkish (tr)	42.87	46.37	47.14	48.72	49.31	**51.12**
Spanish (es)	42.48	43.80	45.85	48.41	48.59	**49.88**
Arabic (ar)	38.48	37.85	42.27	43.08	44.24	**45.41**
Portuguese (pt)	47.81	50.33	51.98	52.15	52.98	**56.08**
Persian (fa)	43.39	45.04	45.25	46.02	47.46	**47.94**
Korean (ko)	75.46	77.73	78.45	79.49	79.11	**80.02**
French (fr)	40.37	40.81	41.89	44.43	45.40	**47.01**
German (de)	40.80	41.42	41.11	41.32	41.38	**42.59**
Thai (th)	44.10	56.27	57.40	58.25	58.80	**59.46**
Italian (it)	42.36	41.82	42.76	45.11	44.18	**45.72**
Hindi (hi)	49.84	51.92	52.58	55.17	55.28	**57.29**
Chinese (zh)	72.88	72.54	72.40	73.85	**73.94**	72.30
Polish (pl)	48.97	50.20	50.50	51.20	51.81	**54.49**
Urdu (ur)	36.44	37.56	39.22	41.53	42.81	**43.39**
Filipino (tl)	52.96	52.99	54.86	56.76	57.33	**59.43**
Greek (el)	44.00	43.94	44.15	46.89	**47.59**	47.43
Serbian (sr)	42.50	42.32	40.71	44.22	45.95	**47.45**
Dutch (nl)	39.75	40.85	41.01	42.36	42.69	**44.80**
Catalan (ca)	48.61	47.85	48.79	51.72	52.60	**52.90**
Swedish (sv)	47.79	47.80	47.31	49.39	51.28	**51.44**
Tamil (ta)	48.04	49.67	50.65	52.85	54.14	**54.92**
Finnish (fi)	45.28	45.28	44.03	43.98	45.59	**46.42**
Czech (cs)	53.03	52.60	52.89	55.01	55.93	**56.02**
Nepali (ne)	44.58	47.00	46.94	49.83	**51.57**	51.12
Marathi (mr)	50.85	48.40	51.44	54.18	**55.76**	55.31
Malayalam (ml)	38.43	42.20	42.77	44.72	**45.86**	44.36
Bangla (bn)	57.79	57.08	56.74	59.11	60.32	**60.92**
Hungarian (hu)	60.29	59.94	60.08	61.86	**63.81**	62.68
Slovenian (sl)	58.79	59.68	59.13	61.18	62.34	**62.74**
Norwegian (no)	46.09	48.94	49.22	49.60	51.11	**51.34**
Telugu (te)	49.54	51.47	52.45	55.13	56.66	**57.03**
Pashto (ps)	29.41	34.92	33.27	35.37	36.21	**38.24**
Danish (da)	59.54	60.35	59.97	61.00	60.33	**61.56**
Central Kurdish (ckb)	40.28	37.59	39.06	42.89	**45.65**	45.26
Gujarati (gu)	52.55	54.09	54.24	57.36	**58.59**	58.54
Romanian (ro)	71.24	71.53	72.34	73.17	73.25	**73.58**
Kannada (kn)	54.19	55.76	56.68	59.09	**61.34**	60.19
Estonian (et)	57.81	58.10	59.00	61.95	61.22	**62.61**
Latvian (lv)	58.03	55.18	57.53	58.43	59.52	**61.47**
Bulgarian (bg)	65.20	65.52	66.45	67.42	66.94	**68.35**
Sinhala (si)	37.71	40.17	42.77	45.72	**47.54**	47.06
Icelandic (is)	51.32	48.53	50.16	53.39	**55.53**	54.61
Sindhi (sd)	27.28	26.46	31.08	32.42	35.05	**35.28**
Basque (eu)	58.55	56.55	56.78	59.56	60.62	**61.04**
Amharic (am)	24.10	28.57	35.01	34.20	**37.47**	36.87
Lithuanian (lt)	71.31	69.50	69.65	72.26	72.43	**73.09**
Welsh (cy)	58.36	58.50	57.56	59.66	59.95	**60.24**
Haitian Creole (ht)	68.13	67.05	67.97	70.70	71.33	**71.41**
Average	50.05	50.86	51.74	53.66	54.62	**55.23**