Enhancing Rumor Detection Methods with Propagation Structure Infused Language Model

Chaoqun Cui, Siyuan Li, Kunkun Ma, Caiyan Jia[*]

School of Computer Science and Technology & Beijing Key Lab of Traffic Data
Analysis and Mining Beijing Jiaotong University, Beijing 100044, China
{ccqun19990728,pratearon}@gmail.com
{siyuanli,cyjia}@bjtu.edu.cn

Abstract

Pretrained Language Models (PLMs) have excelled in various Natural Language Processing tasks, benefiting from large-scale pretraining and self-attention mechanism's ability to capture long-range dependencies. However, their performance on social media application tasks like rumor detection remains suboptimal. We attribute this to mismatches between pretraining corpora and social texts, inadequate handling of unique social symbols, and pretraining tasks ill-suited for modeling user engagements implicit in propagation structures. To address these issues, we propose a continue pretraining strategy called Post Engagement Prediction (PEP) to infuse information from propagation structures into PLMs. PEP makes models to predict root, branch, and parent relations between posts, capturing interactions of stance and sentiment crucial for rumor detection. We also curate and release large-scale Twitter corpus: TwitterCorpus (269GB text), and two unlabeled claim conversation datasets with propagation structures (UTwitter and UWeibo). Utilizing these resources and PEP strategy, we train a Twitter-tailored PLM called SoLM. Extensive experiments demonstrate PEP significantly boosts rumor detection performance across universal and social media PLMs, even in few-shot scenarios. On benchmark datasets, PEP enhances baseline models by 1.0-3.7% accuracy, even enabling it to outperform current state-of-the-art methods on multiple datasets. SoLM alone, without high-level modules, also achieves competitive results, highlighting the strategy's effectiveness in learning discriminative post interaction features.

1 Introduction

Recent years have seen Pretrained Language Models (PLMs) based on Transformer (Devlin et al., 2018; Liu et al., 2023; Brown et al., 2020) excel in various Natural Language Processing (NLP) tasks like machine translation (Vaswani et al., 2017; Edunov et al., 2018), sentiment analysis (Sun et al., 2019; Xu et al., 2019), and question-answering systems (Devlin et al., 2018; Yang et al., 2019). The success is largely due to the parallel computation of self-attention mechanism, enabling long-range dependency capture in texts and intricate semantic learning. Furthermore, PLMs benefit from large-scale pretraining on unlabeled corpora with increased model capacity and depth. In specialized domains, pretraining with extensive unlabeled professional corpora (Lee et al., 2020; Chalkidis et al., 2020; Yang et al., 2020) allows models to absorb domain-specific knowledge and concepts, thus improving performance on related tasks.

The text in social media platforms like Weibo and Twitter originates from user-generated posts and comments. This type of corpus, differing from most text corpora used for language model pretraining (such as book corpora, Wikipedia corpora, etc.), tends to be shorter, highly emotive, and possesses explicit directional relations. In other words, texts on social media contain interactions among users, where comments are specifically directed at other users' posts or comments. Current universal PLMs predominantly utilize token-level pretraining tasks like Causal Language Modeling (CLM) (Radford et al., 2018), which evidently struggle to model text interaction. In this study, we focus on rumor detection, a typical social media application task, to explore how to enhance the universal PLMs performance in such applications.

In prevalent rumor detection methods (Bian et al., 2020; Cui and Jia, 2024), learning from propagation structures of claims (a claim refers to the source post and its comments) is a common strategy, emphasizing semantics, stance, sentiment, and post/user interactions. However, these strategies have not significantly benefited from prevalent universal PLMs. This is reflected in the limited performance boost in rumor detection models employ-

[*] Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 7165–7179
January 19–24, 2025. ©2025 Association for Computational Linguistics

ing universal PLMs for initial feature extraction, as compared to traditional methods like word2vec (Mikolov et al., 2013) and tf-idf (Sparck Jones, 1972). Universal PLMs and word2vec lack inherent understanding of sentiment and stance engagements among posts, necessitating further training via high-level models such as GNNs.

We investigate this underperformance of universal PLMs. To enhance their performance in rumor detection, we propose a continue pretraining strategy called Post Engagement Prediction (PEP). PEP aims to integrate user engagement information, inherent in propagation structures, into PLMs. Additionally, we have collected and open-sourced high-quality data resources, including a large-scale Twitter corpus named TwitterCorpus and two large-scale conversation dataset with propagation structures called Unlabeled Twitter (UTwitter) and Unlabeled Weibo (UWeibo). Using these corpora and PEP strategy, we trained a BERT architecture PLM tailored for social media application tasks (for Twitter platform), named Social Language Model (SoLM). We believe PEP can not only improve PLMs' performance in rumor detection but also offer insights for other social media application tasks such as content recommendation, social network analysis, and user behavior analysis.

In summary, this study contributes as follows:

- We ran extensive experiments, demonstrating the poor performance of universal PLMs in rumor detection and analyzing the reasons.
- We proposed the PEP strategy to integrate user interaction information into PLMs.
- We collected multiple corpora and trained SoLM. We released all our resources.
- Experiments indicate that PLMs trained with PEP enhance the performance of existing rumor detection methods, with even more pronounced improvements in few-shot scenarios.

2 Related Work

In this section, we will review the related works.

2.1 Rumor Detection

Among the existing studies, early rumor detection methods mainly take advantage of traditional classification methods by using hand-crafted features (Castillo et al., 2011; Kwon et al., 2013; Yang et al., 2012). Deep learning has greatly promoted the development of rumor detection methods. These approaches generally fall into four categories: time-series based techniques (Yu et al., 2017; Shu et al., 2017; Liu and Wu, 2018) modeling text content or user profiles as time series; propagation structure learning methods (Ma et al., 2018; De Silva and Dou, 2021; Wei et al., 2021; Qiao et al., 2024) accounting for propagation structures of initial rumors and their replies; multi-source integration approaches (Karimi et al., 2018; Yuan et al., 2019; Birunda and Devi, 2021) combining various rumor resources, such as post content, user profiles, and relations between posts and users; and multi-modal fusion techniques (Jin et al., 2017; Wang et al., 2018; Singhal et al., 2019) that use both post content and associated images for efficient rumor debunking.

In the literature, the significance of propagation structure has been increasingly recognized. Numerous state-of-the-art (SOTA) models employ GNNs to model propagation trees. BiGCN (Bian et al., 2020) implemented a bidirectional Graph Convolutional Network (GCN) (Kipf and Welling, 2016) along with a root node feature enhancement technique. PLAN (Khoo et al., 2020) established a Transformer cognizant of the propagation tree structures. ClaHi-GAT (Lin et al., 2021) used GAT on undirected graphs with sibling relations to model user interactions. GACL (Sun et al., 2022) employed contrastive loss with adversarial training to learn noise-resilient representations of rumors. RAGCL (Cui and Jia, 2024) designed an adaptive graph contrastive learning method considering the structural characteristics of propagation trees. Together, these studies highlight the crucial role of propagation structures and post texts.

2.2 Social Media Language Models

There exists various language models specifically designed for social media. For instance, BERTweet (Nguyen et al., 2020) replicated RoBERTa on 850 million tweets. TimeLMs (Loureiro et al., 2022) utilized a set of RoBERTa models (Liu et al., 2019) to learn from English tweets across various time ranges. Another example is XLM-T (Barbieri et al., 2022), which extended the pretraining process from a XLM-R checkpoint (Conneau et al., 2019) utilizing 198 million multilingual tweets. Additionally, TwHIN-BERT (Zhang et al., 2022) models user engagements as a heterogeneous graph and then utilizes user interaction information on the graph during training process. These PLMs model texts from social corpora, alleviating some issues of universal PLMs in rumor detection. However, their pretraining methods overlook the learning of post engagements and semantic association between multiple posts characterized by propagation structures, which are crucial for rumor identification.

3 Problem Analysis

In this section, we discuss the suboptimal performance exhibited by universal PLMs and its reasons.

3.1 Inefficacy of Universal PLMs

The claim propagation process follows a tree structure, with source post as root and comments as other nodes. The reply relation between comments serve as edges. This tree is the primary data structure processed by rumor detection methods based on propagation structure. See Appendix A for examples of propagation tree. Typically, the interaction relation between comments of rumor and non-rumor claims is markedly different. This is manifested specifically as comments to rumor claims having more intense stances and sentiments, while those to non-rumor claims tend to be more moderate. Propagation structure based methods focus on learning stance and sentiment interaction among posts. These methods generally use common text feature extraction methods for initial post feature vectors, which are then processed by high-level models like GNNs to learn inter-post relations.

We examined the impact of feature initialization methods on rumor detection model including PLAN, BiGCN and GACL across five datasets: Weibo (Ma et al., 2016), DRWeibo (Cui and Jia, 2024), Twitter15, Twitter16 (Ma et al., 2017), and PHEME (Zubiaga et al., 2017). These datasets originate from two large platforms, Twitter and Weibo. We reported macro F1 score on the class-imbalanced dataset PHEME, and accuracy on the other class-balanced datasets. The dataset statistics and the results are presented in Table 1 and 2. In experiments, we involved traditional methods such as tf-idf and word2vec (skip-gram), as well as autoencoding language models like BERT, RoBERTa, BERTweet and TwHIN-BERT, and the generative large language model Baichuan2 (Yang et al., 2023) and LLaMA2 (Touvron et al., 2023). For Baichuan2 and LLaMA2, we utilized the embedding of the last token in a tweet as its representation.

The results indicate: (1) Universal PLMs do not exhibit significant improvement over traditional methods, such as word2vec; (2) Among autoencoding PLMs, TwHIN-BERT and BERTweet models pretrained on Twitter corpus outperforms universal PLMs in most scenarios; (3) While SOTA generative large models (like Baichuan2 and LLaMA2) have several orders of magnitude more parameters compared to traditional PLMs, they do not lead to noticeably better performance. Given the outstanding performance of universal PLMs in other domains (Devlin et al., 2018; Radford et al., 2018), their suboptimal results in rumor detection becomes a question worth investigating.

3.2 Cause Analysis

We attribute the underperformance of universal PLMs primarily to three factors. (1) The training corpora of universal PLMs do not align with social media texts. (2) Universal PLMs are not equipped to properly process symbols unique to social media texts. (3) The pretraining tasks employed by universal PLMs are ill-suited for rumor detection tasks. We will expound on these points.

3.2.1 Training Corpus Mismatch

Universal PLMs are usually trained on corpora such as books and articles (like BooksCorpus (Zhu et al., 2015) or Project Gutenberg[1]), Wikipedia corpora, and web-crawled data (like Common Crawl[2]), with language that is generally more formal, grammatically correct, and skewed towards the written form. However, texts in posts on social platforms is usually colloquial, expressive in a more spoken style, and includes uncivilized language, slang, abbreviations (like U, IC, OIC, THX), emojis, and unique internet terms.

Universal PLMs are mainly trained on long texts, while social media posts are usually very short. We counted the length distribution of 2.8 billion tweets in TwitterCorpus, as shown in Figure 1. We found that posts tend to be very brief, with most (57.92%) having fewer than 20 tokens, and virtually none (0.01%) exceeding 100 tokens. This indicates that the length distribution of texts from social media platforms is significantly different from corpora like BooksCorpus and Wikipedia. This may lead to problems for universal PLMs when dealing with short texts. First, although models pretrained on longer texts excel at capturing long-distance contextual relations, this strength may not be crucial

[1] https://www.gutenberg.org

[2] https://commoncrawl.org

Statistic	Weibo	DRWeibo	Twitter15	Twitter16	PHEME	UWeibo	UTwitter
language	zh	zh	en	en	en	zh	en
labeled	True	True	True	True	True	False	False
# claims	4664	6037	1490	818	6425	209549	204922
# non-rumors	2351	3185	374	205	4023	-	-
# false rumors	2313	2852	370	205	638	-	-
# true rumors	-	-	372	207	1067	-	-
# unverified rumors	-	-	374	201	697	-	-
avg num posts	803.5	61.8	31.1	25.9	15.4	50.5	82.5

Table 1: Statistics of the datasets.

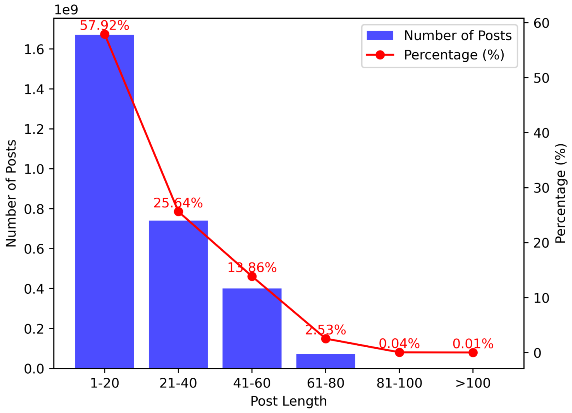


Figure 1: Post lengths distribution on TwitterCorpus.

for short texts, leading to potential misalignment with the characteristics of short-text tasks. Second, there could be disparities in vocabulary and grammatical features between long and short texts. For example, short texts (e.g., tweets, text messages) might contain more informal language, slang, emojis, and abbreviations (as mentioned before). If the PLMs haven't thoroughly learned these features, they could struggle with short texts.

3.2.2 Symbol Processing Shortfalls

Social media posts contain special symbols that represent specific interactive behaviors, mainly including user mentions (like @someone), web/url links, topic tags (like #Covid19), and emojis. These symbols may affect the text semantics learned by PLMs, while some universal PLMs (like BERT, RoBERTa) lack the ability to handle these special symbols properly. Some necessary processes include: (1) Mitigating the impact of user mentions and web links, which usually do not affect the text content; (2) Identifying topic tags in the texts, as they often delineate the subject matter of posts and hold significant semantic value; (3) Recognizing emojis in texts, as the emojis often convey abundant emotional information, crucial for rumor detection reliant on stance and sentiment recognition.

3.2.3 Auxiliary Task Limitations

PLMs typically utilize various pretraining auxiliary tasks to enhance abilities in several aspects, including understanding sentence relations, recognizing entities, and managing grammatical rules. Mainstream auxiliary tasks encompass Next Sentence Prediction (Devlin et al., 2018), Sentence Order Prediction (Lan et al., 2019), Replaced Token Detection (Xiao et al., 2020), etc. These tasks carry out pretraining by learning semantic relations within documents, while rumor detection tasks focus more on interactive relations between documents (posts), particularly stance-related semantic relations. This is mainly because the content of a claim's comment is typically not independent but directional. Users generally express their opinions in response to content posted by other users.

4 Method

In this section, we introduce the datasets curated, and describe how we utilize PEP to train SoLM.

4.1 Pretraining Corpora

TwitterCorpus is a pure text Twitter corpus, which uses The Twitter Stream Grab publicly available on the Archive Team[3] as its data source. It has extracted 2.8 billion English tweets from 2015 to 2022, totaling 269GB of uncompressed texts.

UTwitter contains trending claims from the past two years, collected from Twitter using a web crawler. It comprises about 200,000 unlabeled claims, each with a source post, multiple replies, and its propagation structure, totaling about 17 million tweets. Besides PLM pre-training, it can also be used for semi-supervised rumor detection.

[3] https://archive.org/details/twitterstream

Method	Initialization	Parameters	Dataset				
			Weibo	DRWeibo	Twitter15	Twitter16	PHEME
PLAN	TF-IDF	-	90.8	74.3	80.2	82.0	65.3
	Word2Vec	-	91.5	78.8	81.9	84.3	68.6
	BERT	110M	91.2	77.9	82.7	83.7	68.7
	RoBERTa	125M	91.8	78.3	82.4	83.0	67.8
	BERTweet	110M	-	-	83.2	**84.5**	68.5
	TwHIN-BERT	280M	-	-	82.8	84.3	69.5
	Baichuan2	7B	**92.5**	**79.4**	-	-	-
	LLaMA2	7B	-	-	**83.4**	84.0	**70.2**
BiGCN	TF-IDF	-	93.1	84.2	81.8	84.7	66.7
	Word2Vec	-	94.2	86.6	84.4	**88.0**	70.8
	BERT	110M	**94.4**	86.1	83.5	87.9	70.3
	RoBERTa	125M	93.8	87.2	83.8	87.3	70.5
	BERTweet	110M	-	-	84.9	87.8	71.2
	TwHIN-BERT	280M	-	-	**85.2**	87.2	71.8
	Baichuan2	7B	94.0	**88.7**	-	-	-
	LLaMA2	7B	-	-	85.0	87.0	**72.0**
GACL	TF-IDF	-	92.8	85.7	84.9	85.9	66.9
	Word2Vec	-	93.0	**87.4**	85.0	89.5	71.2
	BERT	110M	93.8	87.0	84.6	89.1	71.1
	RoBERTa	125M	93.4	86.4	85.3	89.4	70.3
	BERTweet	110M	-	-	85.5	**90.2**	71.7
	TwHIN-BERT	280M	-	-	85.8	88.8	71.4
	Baichuan2	7B	**94.3**	87.1	-	-	-
	LLaMA2	7B	-	-	**86.0**	89.8	**72.3**

Table 2: The impact of feature initialization methods. BERT and RoBERTa are employed on Chinese and English datasets respectively, using their corresponding Chinese and English models.

UWeibo contains about 200,000 trending claims from Weibo over the past two years, with about 11 million posts.

TwitterCorpus, UTwitter, and UWeibo datasets are all available at https://mega.nz/folder/wZwFGTzR#eAg4o-xJw3SBxfd2R3AmwQ, https://github.com/CcQunResearch/UTwitter, and https://github.com/CcQunResearch/UWeibo. The statistics and construction process are in Table 1 and Appendix B. See Appendix C.2 for dataset preprocessing and SoLM architecture.

4.2 Post Engagement Prediction

A claim conversation or propagation structure can be seen as a graph or, more specifically, a tree (Ma et al., 2018). This structure is characterized by a canonical node sorting within the tree, which proceeds either top-down or bottom-up (Bian et al., 2020). Existing rumor detection methods based on propagation structures take advantage of the reply relation within the structures to learn the interaction of stance and sentiment between posts, thus identifying discriminative patterns to detect rumors. Thus, it is critical for PLMs to capture these interactive features between nodes in the trees. Yet, this is an aspect that current universal PLMs typically lack, as they tend to focus more on semantic connections within lengthy documents rather than modeling correlations between short ones, which is essential for social media application tasks such as rumor detection. Recognizing this issue, we propose the PEP strategy to assist PLMs in integrating the interaction information in propagation trees.

We found that nodes within a rumor propagation tree share certain connections, including: (1) All nodes are intrinsically linked to the root node, as all claim replies tend to revolve around the source post, discussing specific topics; (2) Nodes on the same branch form a conversation thread with closely related content, where deeper successor nodes are semantically dependent on the shallower prefix nodes; (3) Directly connected nodes exhibit a clear reply relation, with child nodes often stating explicit stances or sentiments towards parent nodes.

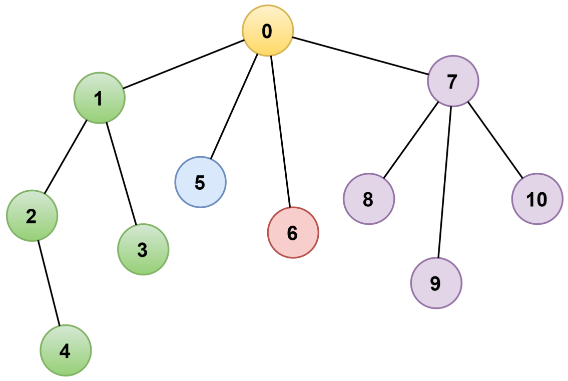


Figure 2: An example of rumor propagation tree. Different colors correspond to different conversation threads.

Post1	Post2	RoP	BrP	PaP
0	1	T	T	T
0	2	T	T	F
1	2	F	T	T
1	4	F	T	F
1	7	F	F	F
4	7	F	F	F
..	..	..	..	..

Table 3: Root, branch and parent relation labels derived from the tree in Figure 2. T for True, F for False.

Such clear semantic connections reflected via the graph structure are due to the claim propagation tree's canonical node sorting.

PEP is a continue pretraining strategy that is conceptually straightforward in its formulation. It uses these node relations conveyed by the propagation structure as self-supervised information to assist PLM pretraining. Specifically, PEP includes Root Prediction (RoP), Branch Prediction (BrP), and Parent Prediction (PaP), which allow a PLM to predict the root, branch, and parent relations in propagation trees, respectively. For example, some RoP, BrP and PaP labels in Figure 2 can be illustrated in Table 3. It is worth noting that although we use BERT as basic architecture of SoLM, PEP can also assist to pretrain all mainstream PLM architectures for tree-structured tasks such as rumor detection.

Root Prediction. RoP promotes learning interactions between a source post and its comment posts by predicting if two nodes are in a root relation (i.e., whether one node is the root node of another). Specifically, for a propagation tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges. $\mathbf{H}_\mathcal{G} \in \mathbb{R}^{n \times d}$ is node feature matrix, where n is node number, and d is dimension of feature vectors. Each row vector in $\mathbf{H}_\mathcal{G}$ represents a sentence embedding extracted from a PLM for a corresponding post. This could be, for instance, the embedding vector of the `[CLS]` token in BERT, or the embedding vector of the final token of a post in an autoregressive model. Then, the loss of RoP is:

$$\mathcal{L}_{\text{RoP}} = -\frac{1}{|\mathbb{G}|} \sum_{\mathcal{G} \in \mathbb{G}} CE(\sigma(\mathbf{H}_\mathcal{G} \mathbf{H}_\mathcal{G}^T), \mathbf{Y}_{\text{RoP}}), \quad (1)$$

where $\mathbb{G}$ is the set of claim propagation trees corresponding to the claims in UTwitter or UWeibo, $CE(\cdot, \cdot)$ is the cross-entropy loss, $\sigma(\cdot)$ is the sigmoid activation function, and $\mathbf{Y}_{\text{RoP}} \in \mathbb{R}^{n \times n}$ is the self-supervised label matrix of root relations extracted from propagation trees.

Branch Prediction. BrP predicts whether two nodes come from the same conversation thread in a propagation tree (the nodes with the same color in Figure 2). Usually, posts in the same conversation thread discuss root post's content from the same perspective. BrP captures the interaction of nodes in the same branch by learning this kind semantic connection. Similarly, we obtain the loss $\mathcal{L}_{\text{BrP}}$ through $\mathbf{H}_\mathcal{G}$ and label matrix $\mathbf{Y}_{\text{BrP}} \in \mathbb{R}^{n \times n}$:

$$\mathcal{L}_{\text{BrP}} = -\frac{1}{|\mathbb{G}|} \sum_{\mathcal{G} \in \mathbb{G}} CE(\sigma(\mathbf{H}_\mathcal{G} \mathbf{H}_\mathcal{G}^T), \mathbf{Y}_{\text{BrP}}). \quad (2)$$

Parent Prediction. PaP is similar to link prediction (Fan et al., 2019; Zhang and Chen, 2018). In a propagation tree, parent-child nodes that are directly connected have clear stances and emotional relations semantically. PaP facilitates the model to learn node interaction that are directly connected in a propagation tree by predicting whether two nodes are directly connected (that is, whether one node is the parent of the other). We use $\mathbf{H}_\mathcal{G}$ and label matrix $\mathbf{Y}_{\text{PaP}} \in \mathbb{R}^{n \times n}$ to derive loss $\mathcal{L}_{\text{PaP}}$:

$$\mathcal{L}_{\text{PaP}} = -\frac{1}{|\mathbb{G}|} \sum_{\mathcal{G} \in \mathbb{G}} CE(\sigma(\mathbf{H}_\mathcal{G} \mathbf{H}_\mathcal{G}^T), \mathbf{Y}_{\text{PaP}}). \quad (3)$$

The final loss function of PEP is as follows:

$$\mathcal{L}_{\text{PEP}} = \alpha \cdot \mathcal{L}_{\text{RoP}} + \beta \cdot \mathcal{L}_{\text{BrP}} + \gamma \cdot \mathcal{L}_{\text{PaP}}. \quad (4)$$

We set $\alpha = \beta = \gamma = 1$ in our experiments.

4.3 Training Strategy

We train on TwitterCorpus with Masked Language Modeling (MLM) to learn basic knowledge (first stage). Then, we train on UTwitter using MLM and PEP (second stage). The process is in Algorithm 1.

Algorithm 1 Pretraining Strategy

```
Input: initial parameter $\theta^{(0)}$, training step $N$ of first stage, training step $M$ of second stage.
Output: optimized model parameter $\theta^{(N+M)}$.
// First stage: pretraining on TwitterCorpus.
for $n = 1$ to $N$ do
    Update $\theta^{(n)}$: minimize $\mathcal{L}_{\text{MLM}}$.
end for
// Second stage: pretraining on UTwitter.
for $m = 1$ to M do
    Update $\theta^{(N+m)}$: minimize $\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{PEP}}$.
end for
return $\theta^{(N+M)}$.
```

5 Experiments

This section presents a evaluation on performance.

5.1 Experimental Settings

We verify the enhancement effect on baseline methods (typical high-level rumor detection methods).

5.1.1 Datasets

We experimented on five benchmark datasets in Table 1. PHEME is a class-imbalanced dataset, while others are class-balanced. We reported macro F1 score on PHEME, and accuracy on the others.

5.1.2 Baselines

We replace the feature initialization modules of the following baseline methods with PLMs trained by PEP to verify its performance enhancement.

PLAN (Khoo et al., 2020) is based on Transformer architecture. Its StA-PLAN version uses rumor propagation structure information.

BiGCN (Bian et al., 2020) utilizes two bidirectional GCN encoders and root node feature enhancement strategy to classify rumor.

ClaHi-GAT (Lin et al., 2021) uses GAT on undirected graphs with sibling relations to model user interactions.

GACL (Sun et al., 2022) uses contrastive learning and adversarial training to classify rumor.

RAGCL (Cui and Jia, 2024) is the current SOTA method on the benchmark datasets. It uses contrast learning with adaptive data augmentation.

These baseline methods all follow a unified framework: (1) Extract initial text features using PLMs or word2vec; (2) Further encode the extracted features using high-level models such as GNNs or Graph Transformer; (3) Train the model using training strategies like contrastive learning or adversarial training. In the experiments conducted in Table 2 and Table 4, we only replaced the feature initialization module in (1) to explore its impact, keeping components in (2) and (3) unchanged.

We further access SoLM's capability to manage rumor detection tasks independently, without using any high-level model (*SoLM Only* in Table 4). Specifically, we follow the GNN approach for graph classification tasks, performing pooling on feature vectors of all posts related to a claim. Each post's feature vector is extracted from the `[CLS]` token representation in SoLM. A linear classifier is then applied. More details of experimental settings is shown in Appendix C. The source code of PEP are available at https://github.com/CcQunResearch/SoLM.

5.2 Results and Discussion

In our experiments, we evaluated the impact of various universal PLMs such as RoBERTa, Baichuan2, LLaMA2, the social media specific PLM TwHIN-BERT, and SoLM on rumor detection methods. We used UWeibo to continue pretraining Chinese PLMs (RoBERTa-base-Chinese and Baichuan2) and UTwitter for others, in order to separately process benchmark datasets in Chinese and English. For earlier PLMs like BERT and RoBERTa, their vocabularies struggle to effectively handle special symbols such as emojis in social text (as mentioned in Section 3.2.2). We utilize reserved tokens (`[unused]`) in BERT or infrequently used tokens in RoBERTa to represent these special symbols. The experimental results are presented in Table 4 (see Appendix D.1 for results on other baselines).

The experimental results indicate that the PEP strategy significantly enhances the performance of these PLMs in rumor detection tasks. Specifically, on the Weibo, DRWeibo, Twitter15, Twitter16, and PHEME datasets, performance improvements of 1.2-2.2%, 1.7-3.7%, 1.1-2.2%, 1.0-1.9%, and 1.5-2.4% were achieved, respectively. The peak performance achieved on multiple datasets even surpassed the latest SOTA results (Cui and Jia, 2024). Furthermore, the direct utilization of SoLM's features for rumor identification, without relying on high-level models, also yielded considerable results. These experimental findings highlight the critical importance of the text feature extraction module's ability to effectively learn the interactive features between posts for typical social media application tasks like rumor detection. Our PEP training strategy integrates the user engagement

Method	Initialization	Dataset				
		Weibo	DRWeibo	Twitter15	Twitter16	PHEME
PLAN	RoBERTa	91.8	78.3	82.4	83.0	67.8
	w/ PEP	94.0(↑2.2)	81.1(↑2.8)	84.2(↑1.8)	85.8(↑2.8)	69.0(↑1.2)
	TwHIN-BERT	-	-	82.8	84.3	69.5
	w/ PEP	-	-	84.7(↑1.9)	86.0(↑1.7)	71.7(↑2.2)
	Baichuan2	92.5	79.4	-	-	-
	w/ PEP	94.8(↑2.3)	83.2(↑3.8)	-	-	-
	LLaMA2	-	-	83.4	84.0	70.2
	w/ PEP	-	-	85.0(↑1.6)	86.6(↑2.6)	71.9(↑1.7)
	SoLM(MLM)	-	-	83.3	84.6	68.7
	SoLM	-	-	85.2(↑1.9)	87.0(↑2.4)	70.6(↑1.9)
BiGCN	RoBERTa	93.8	87.2	83.8	87.3	70.5
	w/ PEP	95.0(↑1.2)	89.7(↑2.5)	85.6(↑1.8)	88.5(↑1.2)	72.0(↑1.5)
	TwHIN-BERT	-	-	85.2	87.2	71.8
	w/ PEP	-	-	87.0(↑1.8)	88.7(↑1.5)	73.8(↑2.0)
	Baichuan2	94.0	88.7	-	-	-
	w/ PEP	95.6(↑1.6)	**90.4(↑1.7)**	-	-	-
	LLaMA2	-	-	85.1	87.0	72.0
	w/ PEP	-	-	87.3(↑2.2)	88.2(↑1.2)	74.0(↑2.0)
	SoLM(MLM)	-	-	85.0	87.3	70.8
	SoLM	-	-	86.6(↑1.6)	89.2(↑1.9)	73.2(↑2.4)
GACL	RoBERTa	93.4	86.4	85.3	89.4	70.3
	w/ PEP	95.2(↑1.8)	89.0(↑2.6)	86.4(↑1.1)	90.4(↑1.0)	72.1(↑1.8)
	TwHIN-BERT	-	-	85.8	88.8	71.4
	w/ PEP	-	-	86.9(↑1.1)	90.0(↑1.2)	73.5(↑2.1)
	Baichuan2	94.3	87.1	-	-	-
	w/ PEP	**96.5(↑2.2)**	90.8(↑3.7)	-	-	-
	LLaMA2	-	-	86.0	89.8	72.3
	w/ PEP	-	-	87.3(↑1.3)	**90.8(↑1.0)**	73.9(↑1.6)
	SoLM(MLM)	-	-	85.4	89.1	72.8
	SoLM	-	-	**87.4(↑2.0)**	90.6(↑1.5)	74.5(↑1.7)
SoLM Only	-	-	-	82.6	83.9	67.4
RAGCL(SOTA)	RoBERTa	96.2	89.4	86.7	90.5	**76.8**

Table 4: Experimental results on benchmark datasets. SoLM(MLM) refers to SoLM without second stage training.

information embedded in the propagation structure of claims into the semantics of PLMs in a straightforward manner, resulting in performance gains without altering the high-level model.

In addition, according to the result in Table 2, the performance of PLMs is comparable to word2vec. A possible explanation is that rumor detection models are not particularly sensitive to feature extraction method, with the performance being primarily driven by high-level model. However, the performance improvements observed in Table 4 underscore the equal importance of underlying feature extraction methods. Extracting better features can aid high-level models in learning more discriminative patterns, thereby achieving superior results.

5.3 Ablation Study

We investigated the impact of SoLM's two training stages and various training tasks on model performance using Twitter15 and Twitter16 with BiGCN. The outcomes, shown in Table 5, reveal the positive effect of TwitterCorpus on the PLM model's performance, likely due to resolving prior corpora mismatch issues. The impact of the second stage training on model performance is also significant. RoP and PaP notably outperform BrP in impacting model performance (order of importance: PaP >= RoP > BrP), implying rumor detection's reliance

	Twitter15	Twitter16
BiGCN w/ SoLM	86.6	89.2
w/o MLM pretraining	85.8(↓0.8)	88.1(↓1.1)
w/o PEP pretraining	85.0(↓1.6)	87.3(↓1.9)
w/o RoP	85.8(↓0.8)	88.3(↓0.9)
w/o BrP	86.3(↓0.3)	88.7(↓0.5)
w/o PaP	85.6(↓1.0)	88.1(↓1.1)

Table 5: Ablation study on corpora and PEP strategy.

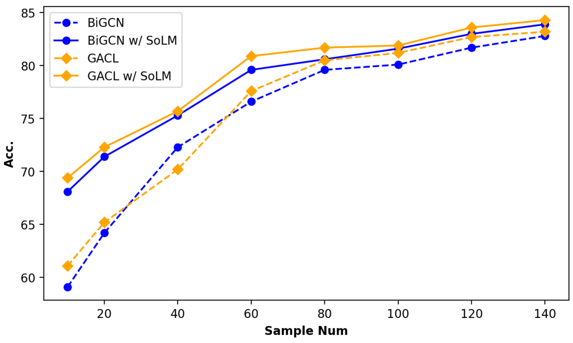


Figure 3: Results of few-shot experiments.

on interactions between replies and source posts, as well as between directly replied posts.

Training a domain specific PLM for social media from scratch using a large-scale corpus like TwitterCorpus demands substantial computational resources (e.g., SoLM requires eight A800 80GB SXM GPUs for 14 days of training). In contrast, employing PEP strategy to continue pretraining an existing universal PLM such as RoBERTa requires only a single A800 80GB SXM GPU for one day, while still achieving a similarly notable improvement in performance (see Table 4). Under conditions of limited computational resources, PEP can serve as an alternative strategy for training social media specific PLMs. Furthermore, the assumptions underlying the design of PEP are universally applicable to social text and rely solely on easily accessible claim texts and propagation structure.

5.4 Few-shot Performance

As shown in Figure 3, we use BiGCN and GACL to conduct few-shot learning experiments on Twitter15 to verify the enhancement effect of SoLM with only a few labeled samples. Because rumors are usually deleted after being detected, making it difficult to gain a large-scale labeled dataset, so the exploration of few-shot rumor detection is essential. We varied the number of labeled samples k between 10 and 140. The results highlight that SoLM significantly enhances baseline model performance with fewer labeled samples. As the number of samples escalates, this enhancement effect tapers off. This superior few-shot performance indicates that SoLM has played a significant role in mitigating the overfitting issue in rumor detection models. See Appendix D.3 for few-shot results on Twitter16.

6 Conclusion

In conclusion, this study identifies significant limitations in using universal PLMs for rumor detection and introduces a novel continue pretraining strategy, PEP, to address these issues. By pretraining on large-scale Twitter corpora and incorporating the PEP strategy focused on post interactions, our SoLM overcomes key deficiencies of traditional PLMs for this domain. Extensive experiments demonstrate that SoLM significantly enhances the performance of existing rumor detection methods, especially in few-shot scenarios.

Ethical Statement

We employed web crawling tools to gather data from publicly available content posted by users on the Weibo and Twitter platforms. This content is accessible to any user of these platforms. To protect privacy, we will process the final dataset by removing any personally identifiable information, ensuring that no individual can be identified. Our exclusive aim in collecting and analyzing this data is for academic research, specifically to enhance the quality of information on social media and curb the spread of misinformation. By leveraging semi-supervised learning methods, we can improve model performance even with limited labeled data, contributing valuable insights to the field of rumor detection. Throughout our research, we are committed to upholding ethical standards, complying with legal requirements, and respecting our data, participants, and society at large.

Limitations

In this study, we propose the PEP continue pretraining strategy and SoLM, which enhance the performance of PLMs in existing rumor detection models. However, these methods also have the following limitations: (1) Although the underlying assumptions of the PEP strategy exhibit a certain degree of generality across different social media application tasks, we have only validated its performance on the rumor detection task so far. Further research is needed to explore its effectiveness in other tasks. (2) The PEP strategy relies on data from specific platforms for pretraining. Although the PEP strategy exhibits a certain degree of cross-platform generalizability, the individual model trained does not possess this capability. (3) Rumor detection faces challenges such as rapid updates, fast dissemination, and significant harm. The performance of the PEP strategy in tasks that require timely updates of information needs further validation.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their help and insightful comments. This work is supported in part by the National Key R&D Program of China (2018AAA0100302) and the National Natural Science Foundation of China (61876016).

References

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

S Selva Birunda and R Kanniga Devi. 2021. A novel score-based multi-source fake news detection using gradient boosting algorithm. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 406–414. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Chaoqun Cui and Caiyan Jia. 2024. Propagation tree is not deep: Adaptive graph contrastive learning approach for rumor detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 73–81.

Nisansa De Silva and Dejing Dou. 2021. Semantic oppositeness assisted deep contextual modeling for automatic rumor detection in social networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 405–415.

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*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426.

Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. 2017. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 795–816.

Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. 2018. Multi-source multi-class fake news detection. In *Proceedings of the 27th international conference on computational linguistics*, pages 1546–1557.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8783–8790.

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

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th international conference on data mining*, pages 1103–1108. IEEE.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Hongzhan Lin, Jing Ma, Mingfei Cheng, Zhiwei Yang, Liangliang Chen, and Guang Chen. 2021. Rumor detection on twitter with claim-guided hierarchical graph attention networks. *arXiv preprint arXiv:2110.04522*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yang Liu and Yi-Fang Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Timelms: Diachronic language models from twitter. *arXiv preprint arXiv:2202.03829*.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Yuhan Qiao, Chaoqun Cui, Yiying Wang, and Caiyan Jia. 2024. A debiased self-training framework with graph self-supervised pre-training aided for semi-supervised rumor detection. *Neurocomputing*, 604:128314.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. 2019. Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE fifth international conference on multimedia big data (BigMM)*, pages 39–47. IEEE.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.

Tiening Sun, Zhong Qian, Sujun Dong, Peifeng Li, and Qiaoming Zhu. 2022. Rumor detection on social media with graph adversarial contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2789–2797.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857.

Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue, and Songlin Hu. 2021. Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection. *arXiv preprint arXiv:2107.11934*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gram: pre-training with explicitly n-gram masked language modeling for natural language understanding. *arXiv preprint arXiv:2010.12148*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*, pages 1–7.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification. In *IJCAI*, pages 3901–3907.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2019. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In *2019 IEEE international conference on data mining (ICDM)*, pages 796–805. IEEE.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

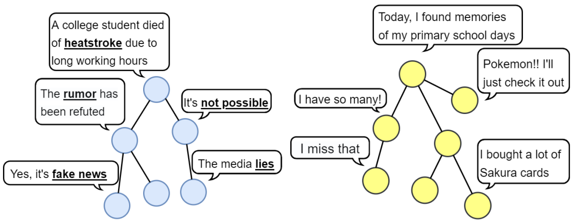
Xinyang Zhang, Yury Malkov, Omar Florez, Serim Park, Brian McWilliams, Jiawei Han, and Ahmed El-Kishky. 2022. Twhin-bert: A socially-enriched pre-trained language model for multilingual tweet representations at twitter. *arXiv preprint arXiv:2209.07562*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *International conference on social informatics*, pages 109–123. Springer.

A Claim Propagation Trees

Figure 4 shows two examples of claim propagation trees from the Twitter platform, where the replies of rumor and non-rumor claims exhibit distinct differences in stance and sentiment. These are key features for identifying rumors.


A college student died of **heatstroke** due to long working hours
The **rumor** has been refuted
It's **not possible**
Yes, it's **fake news**
The media **lies**
Today, I found memories of my primary school days
Pokemon!! I'll just check it out
I have so many!
I miss that
I bought a lot of Sakura cards

(a) Rumor (b) Non-rumor

Figure 4: Examples of claim propagation trees. Comments under rumor conversation typically express more heated stances and sentiments.

B Unlabeled Dataset Construction

For the UWeibo dataset, we employed web crawler techniques to randomly collect trending posts and their complete propagation structures from the homepage of popular Weibo posts[4]. To ensure the dataset's integrity and independence from platform recommendation algorithms, we utilized multiple newly created accounts to extract data. This approach aimed to mitigate potential biases that might arise from the platform's algorithms and to reflect the genuine domain distribution of social media content. The code for the web scraping program can be found at `https://github.com/CcQunResearch/WeiboPostAndCommentCrawl`.

[4] `https://weibo.com/hot/weibo/102803`

For UTwitter dataset, we initially utilized multiple newly created accounts to randomly follow high-follower count influencers. Subsequently, we conducted random crawling of posts and their propagation structures from the Twitter homepage[5]. Due to the fact that UTwitter dataset is exclusively sourced from users with a substantial number of followers, the authenticity of the posts is more likely to be ensured. The code for the web scraping program is available at `https://github.com/CcQunResearch/TwitterPostAndCommnetCrawl`.

Due to the stringent regulations imposed by platforms on the dissemination of rumors, acquiring a sufficiently large-scale labeled dataset for rumor detection proves to be exceptionally challenging. Conversely, obtaining extensive amounts of unlabeled data is relatively simpler, especially with the availability of platform data APIs offered by certain mainstream social media platforms (e.g., Twitter API). Consequently, we suggest that future research should place greater emphasis on semi-supervised rumor detection methods.

Regarding the issue of potential data leakage, we believe that its impact is minimal in our study. This is because the Twitter15 and Twitter16 datasets include both source tweets and their corresponding comments, with each source tweet typically associated with dozens to hundreds of varying comments. In contrast, the TwitterCorpus dataset comprises only source tweets from the years 2015-2022 and does not include any comments. Therefore, the majority of the texts in Twitter15 and Twitter16 are not present in TwitterCorpus. As for UTwitter, it contains data from only the past one to two years, which does not overlap temporally with the Twitter15 and Twitter16 datasets (from the years 2015 and 2016). Consequently, we believe that data leakage is not a significant concern for the methodology employed in our paper.

C Experimental Details

This section primarily details the experimental setup.

C.1 Main Setting

All models are implemented by PyTorch and the baseline methods are re-implemented. It should be noted that BiGCN and GACL utilize early stopping to observe the performance that models can achieve. However, due to oscillations in the early stages of model training, the observed model performance is unstable. In order to compare the performance of different models more fairly, we conduct experiments on multiple baseline methods with the same data, while all models are trained for 100 epochs until convergence. We consider the average results of the final 10 epochs out of these 100 as the stable outcome that the models can achieve.

C.2 Preprocessing and Architecture

For the texts in TwitterCorpus and UTwitter, we first standardize the different fonts present in the texts, then identify user mentions and web links as special tokens, `<@user>` and `<url>`. Next, we use the TweetTokenizer from the NLTK toolkit (Bird et al., 2009) to tokenize the raw texts. Then, we use the `emoji` package[6] to translate the emojis in the texts into text string tokens. Considering the vast scale of our corpora and the fact that tweets usually contain a lot of informal language, slang, internet jargon, and emojis, we set a larger vocabulary size of 52,000 for SoLM. Our SoLM adopts $\text{BERT}_{\text{base}}$ (Devlin et al., 2018) as the model architecture. In conjunction with the previous statistics on text length, we set the maximum positional encoding of the model to 128. In total, there are seven special tokens in the vocabulary: `[UNK]`, `[SEP]`, `[PAD]`, `[CLS]`, `[MASK]`, `<@user>`, and `<url>`.

C.3 Optimization

We use Huggingface Transformers (Wolf et al., 2020) to implement the basic architecture of SoLM. We set the maximum sequence length to 128 and use the AdamW optimizer (Loshchilov and Hutter, 2017) to optimize the model. In the first stage of model training, we set the batch size to 8,000 and the peak learning rate to 0.0004, and use the first 4 epochs out of 40 epochs to warm up the learning rate. In the second stage, we set the batch size to 64, the peak learning rate to 0.00005, and use the first 2 epochs out of 20 epochs to warm up the learning rate. The entire training process was conducted over a span of 14 days on eight A800 80GB SXM GPUs.

D Extended Experiments

This section will present additional evaluation experiments.

[5] `https://twitter.com/home`

[6] `https://pypi.org/project/emoji`

D.1 PEP Performance on Additional Baselines

Similar to Table 4, we validated the performance of the PEP strategy and SoLM on ClaHi-GAT (Lin et al., 2021) and RAGCL (Cui and Jia, 2024), shown in Table 6. Similarly, the PEP strategy enhances the performance of various PLMs in the rumor detection task, further confirming the generalizability of the PEP strategy. The performance improvement of the PEP strategy on ClaHi-GAT indicates that the language model trained with the PEP strategy is not only suitable for models of the Transformer architecture such as PLAN, as well as GCN architectures like BiGCN and GACL, but also effectively applicable to models of the GAT architecture. Our experiments in Appendix D.2 further corroborate this. Additionally, as a recent SOTA method, RAGCL has already achieved excellent performance, but the language model trained with the PEP strategy provided a performance enhancement of 0.7-2.0% across various datasets.

D.2 Extended Ablation Study

We conducted experiments on three commonly used GNN encoders, namely Graph Convolutional Network (GCN) (Kipf and Welling, 2016), Graph Attention Network (GAT) (Veličković et al., 2017), and Graph Isomorphism Network (GIN) (Xu et al., 2018), to explore the generality of SoLM in enhancing the performance of various high-level models. The experimental results are presented in Table 7, indicating that SoLM consistently improves the performance of these generic GNNs in rumor detection tasks. This observation underscores the versatility of SoLM across different models for rumor detection.

D.3 Extended Few-shot Experiments

We present few-shot experiments on the Twitter16 dataset in Figure 5. Similar to the results in Figure 3, pre-training SoLM on large-scale data improves the performance of existing rumor detection models when confronted with a small number of labeled samples.

E Discussions

In this section, we will address some concerns that readers may have regarding the PEP strategy and the SoLM language model.

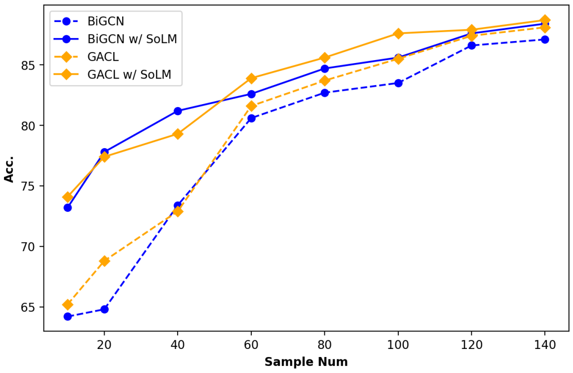


Figure 5: Results of few-shot experiments on Twitter16 dataset.

E.1 Consumption of Computing Resources

As previously discussed in Section 5.3, although pre-training on the TwitterCorpus requires substantial computational resources (8 A100 GPUs for 14 days), fortunately, fine-tuning an existing open-source language model using the PEP strategy requires only a single A100 GPU for one day. This is a manageable requirement for most developers and still results in significant performance improvements (as shown in Table 4 and Table 6). We also highly recommend utilizing the method of fine-tuning open-source models. In fact, the comparative experiments on the performance of SoLM(MLM) and SoLM in Tables 4 and 6 also indicate that the PEP continue training process in the second stage, which integrates propagation structure information, is more important than training a language model from scratch.

E.2 Platform Generalizability

It is important to note that the cross-platform generalizability we focus on refers to the ability of the PEP strategy to utilize data from different platforms for training and to be effective across those respective platforms, rather than training a single model that performs effectively on any platform's application task. This distinction is made because, in real-world applications, the latter is relatively meaningless.

Different platforms indeed exhibit noticeable differences, likely due to the varying user bases they cater to (across different age groups, ethnicities, languages, etc.), as well as the influence of platform-specific recommendation algorithms (Sun et al., 2022; Cui and Jia, 2024). However, these platforms (such as mainstream platforms like Twitter, Weibo, Reddit, YouTube and TikTok) typically

Method	Initialization	Dataset				
		Weibo	DRWeibo	Twitter15	Twitter16	PHEME
ClaHi-GAT	RoBERTa	93.4	86.4	85.0	88.5	70.3
	w/ PEP	94.8(↑1.4)	88.1(↑1.7)	86.6(↑1.6)	89.9(↑1.4)	72.5(↑2.2)
	TwHIN-BERT	-	-	85.3	88.6	70.9
	w/ PEP	-	-	87.1(↑1.8)	90.6(↑2.0)	72.6(↑1.7)
	Baichuan2	94.0	86.9	-	-	-
	w/ PEP	95.1(↑1.1)	89.4(↑2.5)	-	-	-
	LLaMA2	-	-	85.4	89.1	71.4
	w/ PEP	-	-	86.7(↑1.3)	90.7(↑1.6)	73.2(↑1.8)
	SoLM(MLM)	-	-	85.6	88.9	72.1
	SoLM	-	-	87.3(↑1.7)	90.9(↑2.0)	74.2(↑2.1)
RAGCL	RoBERTa	96.2	89.4	86.7	90.5	76.8
	w/ PEP	96.9(↑0.7)	90.8(↑1.4)	87.8(↑1.1)	91.4(↑0.9)	78.8(↑2.0)
	TwHIN-BERT	-	-	86.4	90.3	77.0
	w/ PEP	-	-	87.4(↑1.0)	91.6(↑1.3)	78.6(↑1.6)
	Baichuan2	95.9	89.9	-	-	-
	w/ PEP	96.8(↑0.9)	91.4(↑1.5)	-	-	-
	LLaMA2	-	-	86.6	90.3	77.1
	w/ PEP	-	-	87.7(↑1.1)	91.3(↑1.0)	78.9(↑1.8)
	SoLM(MLM)	-	-	86.5	91.0	77.0
	SoLM	-	-	87.5(↑1.0)	92.3(↑1.3)	78.6(↑1.6)
SoLM Only	-	-	-	86.7	90.5	76.8

Table 6: Experimental results of additional baseline models on benchmark datasets.

Method	Initialization	Twitter15	Twitter16
GCN	RoBERTa	81.5	83.3
	SoLM	83.5(↑2.0)	84.7(↑1.4)
GAT	RoBERTa	80.9	82.1
	SoLM	83.0(↑2.1)	83.8(↑1.7)
GIN	RoBERTa	81.9	82.9
	SoLM	84.2(↑2.3)	84.4(↑1.5)

Table 7: Performance enhancement on general GNNs.

organize post responses in a tree structure (Ma et al., 2018; Bian et al., 2020). Our PEP strategy does not make platform-specific assumptions but rather utilizes the topological relations among nodes within this tree structure. The PEP strategy leverages unlabeled data and self-supervised learning to capture language patterns, thus possessing a certain degree of adaptability across different platforms.

Our PEP strategy merely leverages the primary topological relations within the tree structure (Root, Branch, Parent Relation) for self-supervised LM continue pretraining. Our training approach only adds a linear layer to the model to predict the node topological relations and does not employ contrastive learning (Sun et al., 2022), attention mechanisms (Lin et al., 2021), or other strong prior assumption methods (De Silva and Dou, 2021; Wei et al., 2021; Cui and Jia, 2024). This is primarily to minimize unnecessary inductive biases and to make as few prior assumptions about the data as possible. We aim for the model to learn the general relations between replies rather than specific behaviors.

Experimental results have demonstrated that PEP consistently shows effectiveness on both Twitter and Weibo. Therefore, it can be concluded that the PEP strategy exhibits a certain degree of generalizability across different platforms. As for its performance beyond Twitter and Weibo, further research can be conducted in the future.