GPSocio: A Transformer-based General-purpose Social Network Representation System

Anonymous ACL submission

Abstract

The paper contributes to social network representation (i.e., embedding) techniques for the purpose of supporting diverse downstream analytics and allowing knowledge transfer to data-scarce target domains. Social network analysis covers multiple types of tasks, including user preference detection (Pereira et al., 2018), information propagation (Zhao et al., 2012), influencer detection (Huynh et al., 2020), community detection (Bedi and Sharma, 2016) and sentiment analysis (SA) (Pozzi et al., 2016). One line of prior work focuses on improving social network representations for specific downstream tasks. While these approaches can achieve high performance on the specific intended tasks, their representations may not generalize well to support other purposes.

The emergence of Graph Foundation Models (GFMs) inspires new approaches for social network representation. Specifically, with the development of Large Language Models (LLMs), graph-based properties and text-based information can be jointly utilized for representation learning (Zhao et al., 2024; Ye et al., 2023; Zhao et al., 2023). To enhance reasoning over social media data, a key challenge of LLM-based GFMs is to align the original graph data with the sequential input language format needed for LLMs to understand it. In the environment of social networks, we leverage assertion propagation sequences as token sequences and represent nodes along each propagation sequence by their textural information to generate natural language paragraphs.

1 Introduction

The paper contributes to social network representation (i.e., embedding) techniques for the purpose of supporting diverse downstream analytics and allowing knowledge transfer to data-scarce target domains. Social network analysis covers multiple types of tasks, including user preference detection (Pereira et al., 2018), information propagation (Zhao et al., 2012), influencer detection (Huynh et al., 2020), community detection (Bedi and Sharma, 2016) and sentiment analysis (SA) (Pozzi et al., 2016), among others. One line of prior work focuses on improving social network representations for specific downstream tasks. While these approaches can achieve high performance on the specific intended tasks, their representations may not generalize well to support other purposes.

The emergence of Graph Foundation Models (GFMs) inspires new approaches for social network representation. Specifically, with the development of Large Language Models (LLMs), graph-based properties and text-based information can be jointly utilized for representation learning (Zhao et al., 2024; Ye et al., 2023; Zhao et al., 2023). To enhance reasoning over social media data, a key challenge of LLM-based GFMs is to align the original graph data with the sequential input language format needed for LLMs to understand it. In the environment of social networks, we leverage assertion propagation sequences as token sequences and represent nodes along each propagation sequence by their textural information to generate natural language paragraphs.

Using the above technique, we propose a novel transformer-based general-purpose representation framework, GPSocio, that aims to learn user and assertion representations to solve multiple downstream tasks. Following the GFMs framework, GPSocio is pre-trained with large-scale assertion propagation sequences and updated via small-scale domain-specific data. The framework of GPSocio contains four parts, (1) data pre-processing, which generates assertion propagation sequences in a unified natural language format that LLMs can understand; (2) representation generation, where sequences are processed by multiple embedding layers and then fed into the Longformer to generate representations; (3) model pre-training, which pre-trains GPSocio with large-scale sequences through

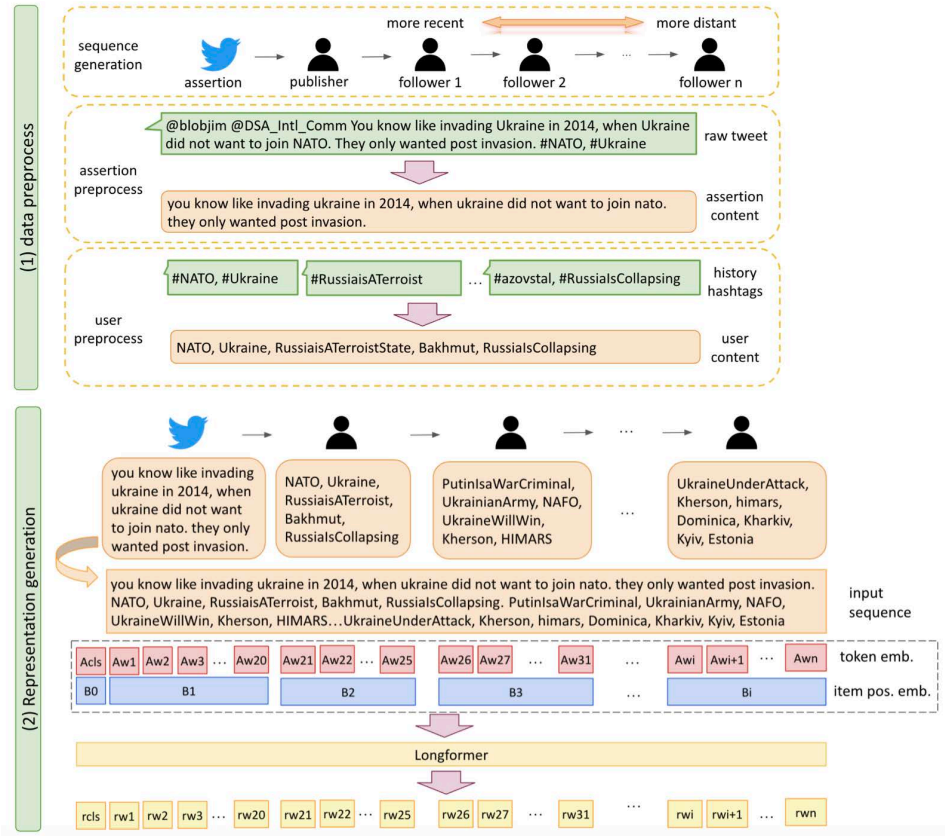


Figure 1: Data preprocess and representation generation of GPSocio.

a next user prediction task via contrastive learning; (4) domain-specific representation adaptation, which adapts the representation by using a small-scale user-assertion graph. Part (1) and (2) are in Figure 1. Part (3) and (4) are in Figure 2.

With the generated representation, GPSocio is able to resolve various downstream tasks. For illustration, this paper showcases three downstream tasks: (a) user prediction, (b) assertion prediction, and (c) sentiment analysis. Our main contributions can be summarized as follows:

1. We develop a general-purpose social networks representation system, GPSocio, that can capture various aspects of social information to resolve multiple downstream tasks.

2. We convert social networks to LLMs' understandable natural language format by leveraging assertion propagation sequences, user-used hashtags and the cleaned version of assertions to identify users and assertions semantically.

3. GPSocio alleviates the data sparsity problem by adopting the GFMs framework, where GPSocio is pre-trained with large-scale data and updated with small-scale domain-specific target data.

4. GPSocio designs the pre-training task as next user prediction and solves it via contrastive learning. This task intuitively retains nodes' semantic content by using textual identifications to represent nodes, while capturing social relations contained in assertion propagation sequences.

5. Extensive experimental results demonstrate that GPSocio significantly outperforms competitors in both user and assertion prediction tasks, achieving multiplicative gains of 198.21% in HIT@10 and 167.39% in NDCG@10, as well as 167.27% in HIT@5 and 129.36% in NDCG@5, on average. In sentiment analysis, GPSocio consistently outperforms DeBerta across all datasets.

2 Related Work

Social Networks Representations

Many inference tasks are carried out on social network data, including user preference detection (Pereira et al., 2018), information propagation (Zhao et al., 2012), influencer detection (Huynh et al., 2020), community detection (Bedi and Sharma, 2016), and sentiment analysis (SA) (Pozzi et al., 2016). One line of research applies representation techniques to help solve these problems by embedding social network data into a lower-dimensional space with key features maintained. In the domain of user preference prediction, an example is to convert users'

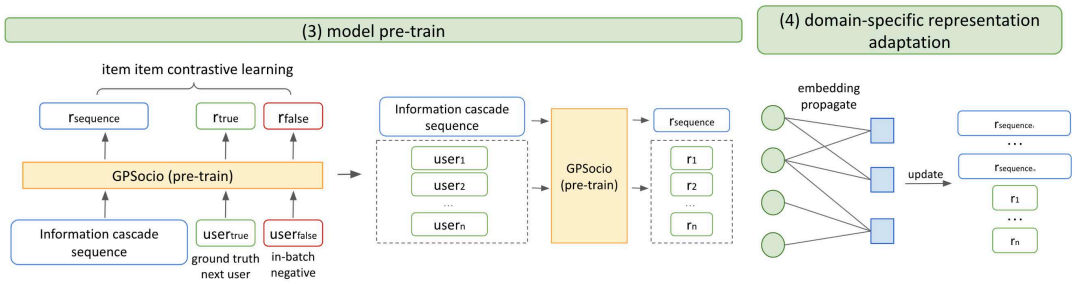


Figure 2: Model pre-train and domain-specific representation adaptation part of GPSocio.

purchase history into key-value pairs and apply transformers (Li et al., 2023). In information propagation area, one branch of work models social networks as a graph and uses Graph Neural Networks (GNNs) for information propagation prediction (Sankar et al., 2020; Sun et al., 2022). For influencer detection, an example is to detect influencers using graph-based long short-term memory (GLSTM) (Kumar et al., 2023). In community detection, an example is unsupervised attributed network embedding (Zhou et al., 2023). LLMs and attention mechanisms were also used for sentiment analysis (Liu et al., 2020; Xu et al., 2019). The above examples of social network representation approaches are designed for specific downstream tasks and are not intended for generalization to multiple purposes. Our work aims to design a general-purpose social network representation system that can help with multiple downstream tasks and enables representation transfer across datasets.

Graph Foundation Models (GFMs)

Foundation Models (FMs) attain knowledge transfer by using large amounts of data for self-supervised pre-training, then fine-tuning (some part of) the model using small-scale task-specific datasets to accommodate various downstream tasks (Bommasani et al., 2021). The capabilities of FMs ushered in a new paradigm in graph learning and motivated the emergence of GFMs. Based on backbone model types, GFMs can be divided into three categories, GNN-based, GNN+LLM-based, and LLM-based (Liu et al., 2023). GNN-based approaches aim to enhance existing graph learning paradigms. Many (Huang et al., 2024; Kipf and Welling, 2016b) adopt Graph Convolutional Networks (GCNs) (Kipf and Welling, 2016a) as the backbone model due to its simplicity and effectiveness. Others aim to improve the expressive power of traditional Graph Neural Networks (GNNs), for example, by applying fully-connected self-attention mechanisms and designing graph transformers (Ying et al., 2021). GNN+LLM-based approaches explore the synergy between GNNs and Large Language Models (LLMs) to empower each other with enhanced capabilities. One example (Chien et al., 2021) utilizes graph-aware self-supervised learning to fine-tune Language Models. Another (Mavromatis et al., 2023) optimizes a GNN teacher over the nodes of the graph using shared LLMs. With the recent surge of LLMs, LLM-based models emerge, which leverage LLMs as backbone models for graph learning. The key challenge becomes to convert graph structure into understandable natural language. For example, recent work (Zhao et al., 2024) integrates graph data with text data by treating node representations as tokens. Other work (Zhao et al., 2023) introduces a "Graph-syntax Tree" to describe graph data using natural language. Inspired by LLM-based GFMs, our research utilizes the nature of social networks and translate social graphs into assertion propagation sequences. To effectively process long cascades, we leverage Longformer's long document process ability and generate general purpose representations for social networks.

3 Concepts and Problem Statement

Concepts: We consider a pre-training input comprised of social media posts by the respective sources from a large data set that ideally covers multiple locales, topics, and time-periods. This input comprises *users* and *assertions*. We denote the set of all assertions by $\mathcal{A}$ and the set of all users by $\mathcal{U}$, where $|\mathcal{A}| = m$ and $|\mathcal{U}| = n$. Let us further denote the lth assertion by $a_l \in \mathcal{A}, 1 \leq l \leq m$, and the kth user by $u_k \in \mathcal{U}, 1 \leq k \leq n$. An assertion is said to be *published* by the initial user who posted it. It can then be *forwarded* by other users. We denote by t_{kl} the timestamp of posting assertion a_l by user u_k. Let $a_{ki} \in \mathcal{A}$ represent the i_{th} assertion in the history sequence of user u_k. Similarly, let user $u_{lj} \in \mathcal{U}$ represent the j_{th} user

in the propagation sequence of assertion a_l. In the assertion propagation sequence, the first user who publishes the assertion is called the *publisher* u_{l1}, followed by forwarders $u_{lj}, j > 1$.

More specifically, an *assertion propagation sequence* represents a series of users propagating an assertion. We use p_l to represent the assertion propagation sequence of a_l. The first node of the sequence is the assertion a_l itself. The second node is publisher u_{l1}. Based on the intuition that the most recent posting user will have closer connection with the next user, the remaining nodes $u_{lj}, j > 1$ will be arranged in a reverse-chronological order, as is shown in module (2) in Figure 1. A *User history sequence* represents a user's history of posted assertions. We use f_k to represent such sequence for user $u_k, 1 \leq k \leq n$. The representation f_k starts with user u_k, followed by the propagated assertions $a_{k1}, a_{k2}, ...$ in chronological order.

Problem definition: Consider a set of n users, $\mathcal{U} = \{u_1, u_2, ...u_n\}$, a set of m assertions, $\mathcal{A} = \{a_1, a_2, ...a_m\}$, a set of user history sequences, $\mathcal{F} = \{f_1, f_2, ..., f_k, ..., f_n\}$, where f_k represents the user history sequence of u_k, and (equivalently) a set of assertion propagation sequences, $\mathcal{P} = \{p_1, p_2, ..., p_l, ..., p_m\}$, where p_l represents the assertion propagation sequence of a_l. The goal of GPSocio is to generate representations for every item in $\mathcal{U}, \mathcal{A}, \mathcal{P}, \mathcal{F}$, denoted by $r_{u_k}, r_{a_l}, r_{p_l}, r_{f_k}$ for user u_k, assertion a_l, assertion propagation sequence p_l, and user history sequence f_k, respectively. These representations are then used to solve multiple downstream inference tasks.

4 GPSocio Framework

We design four modules for the transformer-based GPSocio framework:

Data preprocessing: To re-format social networks data into an LLM-understandable format, we adopt semantic identification for users and assertions, and concat the items with certain chronological order as they propagate. This design captures the semantic information of users and assertions while retaining the information propagation patterns and hidden social relations.

Representation generation: Due to the existence of long information cascades, we adopt modified Longformer (Beltagy et al., 2020), which can process long documents, as our backbone transformer.

Model pre-training: We choose GPSocio's pre-training task to be *next user prediction* along assertion propagation sequences, because they contain semantic information, social relations and behaviour patterns. These features will jointly decide the next user in social networks.

Domain-specific representation adaptation: Though different social networks share common features, domain-specific adaptation helps update representations more towards target domain knowledge. We design domain-specific adaptation based on the target task.

4.1 Data Preprocessing: Generating LLMs Understandable Assertion Propagation Sequences

The data preprocessing module contains (1) assertion propagation sequence generation, (2) assertion text preprocessing, and (3) user text preprocessing shown in Figure 1. Details are in Appendix A.1.

The **assertion propagation sequence** generation creates a sequence containing an assertion and users who posted this assertion, as is introduced in Session 3. The **assertion text preprocessing** function aims to clean up assertion text by excluding non-informative textural information, as detailed in Appendix A.1. The **user text preprocessing** function extracts each user's history hashtags as a semantic identification of users' interests. The number of hashtags are usually limited but the combination of them can nearly uniquely identify a user, making them a concise vehicle for user identification.

4.2 Representation Generation: Utilizing Longformer for Social Sequence Process

The representation generation module of GPSocio contains (1) input sequence generation, (2) sequence embedding generation, (3) Longformer implementation, and (4) representation generation.

Input sequence generation: Based on the data preprocessing procedure in Appendix A.1, we generate the assertion propagation sequence by concatenating assertion's semantic identification and users' semantic identification with the order of assertion propagation sequence. For assertion propagation sequence p_l, the input sequence is denoted as:

$$X = \{[CLS], s_{a_l}, s_{u_{l1}}, s_{u_{l2}}, ..., s_{u_{lj}}, ...\}, \quad (1)$$

where s_{a_l} and $s_{u_{lj}}$ are the semantic identifications of a_l and the j_{th} user in the sequence.

Embedding generation: To represent the propagation patterns and related semantic information, we

design a representation model that combines embedding generated from language models to utilize semantic information (Devlin et al., 2018; Zhuang et al., 2021), and self-attention sequential models to capture sequential patterns (Sun et al., 2019). The embedding contains the following parts:

Token embedding represent the embedding of each token. The token embedding is stored in $A \in R^{D_w \times d}$, D_w is the total number of tokens and d token embedding dimension. The token embedding conveys the semantic information of users and assertions.

Item position embedding represents the position of an item in the "cascade sentence". We use $B_k \in R^d$ to represent the k_{th} item in "cascade sentence" X. The item position embedding of every sequence is denoted as $B^{n \times d}$, where n represents the maximum length of words among assertion propagation sequences. This embedding term records information cascade sequences and is important in capturing cascade patterns.

Longformer implementation: To resolve traditional transformer's limits in processing long sequences, LongFormer (Beltagy et al., 2020) introduced a new attention mechanism that scales linearly with sequence length, thus making it easy to process long documents, which suits the social data scenario where long propagation sequences widely exist. In Longformer, the [CLS] token is used to extract global attention and other tokens are used to extract local window attention. The attention pattern of Longformer consists of four parts: (i) *Sliding window*, which is a fixed-size window attention surrounding each token; (ii) *Dilated sliding window*, which further increases the receptive field without increasing computation; (iii) *Global attention*, which is a symmetric attention operation that can attend to all tokens; and (iv) *Linear Projections for Global Attention*, which applies two sets of projections that provide flexibility to model different types of attention. The detailed introduction is attached in Appendix A.2.

Representation generation: Given the word embedding, the representation of an assertion propagation sequence generated by Longformer can be formulated as:

$$
\begin{aligned}
&E_w = LayerNorm(A_w + B_w), \\
&E_{p_l X} = [E_{p_l[CLS]}, E_{p_l w_1}, ..., E_{p_l w_h}], \\
&[r_{p_l[CLS]}, r_{p_l w_1}, ..., r_{p_l w_h}] \\
&\quad = Longformer([E_{p_l[CLS]}, E_{p_l w_1}, ..., E_{p_l w_h}]),
\end{aligned}
$$

in which w is a word from p_l, $E_{p_l X} \in R^{(h+1) \times d}$ is the embedding of p_l with the maximum token length h, LayerNorm is the layer normalization technique introduced in (Ba et al., 2016).

In GPSocio, we generate representations for assertions, users, assertion propagation sequences and user history sequences. The specific generation procedure is as follows:

Assertion propagation sequences: We use the word representation $r_{p_l[CLS]}$ of token [CLS] to represent the assertion propagation sequence p_l.

Assertions: Since the assertion propagation sequence p_l reflects the propagation path of a_l, we use $r_{p_l[CLS]}$ to denote the representation of a_l.

User: To embed each user u_k, we first generate a "assertion propagation sequence" $X = \{[CLS], s_{u_k}\}$. Then we apply Longformer to generate the representation $[r_{u_k[CLS]}, r_{u_k}]$ and use $r_{u_k[CLS]}$ as the representation of user u_k.

User post history sequences: Given a user post history sequence $f_k = [u_k, a_{k1}, a_{k2}, ...a_{ki}, ..., a_{kq}]$, the user post history sequence representation is the weighted sum of all nodes, λ represents the weight of user representation, as is in Equation 2.

$$
r_{f_k} = \lambda r_{u_k} + \frac{\Sigma_{i=1}^{q} i^2 * r_{a_k i}}{q} \tag{2}
$$

4.3 Model Pre-training: Utilizing Assertion Propagation Sequence to Mine Hidden Information

Since in social networks, the semantic information, user behavior (post) patterns, and information propagation patterns constitute import knowledge latent within assertion propagation sequence, we define our pre-training task as the next user prediction task along the assertion propagation sequence. We apply item-item contrastive (IIC) learning to solve this pre-training task. Instead of negative sampling (Kang and McAuley, 2018), we adopt in-batch next item as negative samples (Chen et al., 2017). In-batch negative samples are the ground truth next items of other assertion propagation sequences in the same batch. Since GPSocio encodes items directly, it is infeasible to re-encode items from traditional negative sampling approach. One potential issue of in-batch negative sampling is that it is likely to generate false negative samples. To avoid this phenomenon from happening, after the in-batch negative samples are generated, we iterate the whole dataset to see if the in-batch negative samples is true negative. We would regenerate a in-batch negative sample for at most 2 more times

if the false negative issue is detected. In this way, we ensure the quality and efficiency of negative sampling. The IIC loss is defined as Equation 3 (τ is a constant):

$$sim_{p_l,u_k} = \frac{r_{p_l}^T r_{u_k}}{\|r_{p_l}\|\|r_{u_k}\|},$$
$$L_{iic} = -log\frac{e^{\frac{sim(r_{p_l},r_{true})}{\tau}}}{\Sigma_{u_k \in G_{batch}} e^{\frac{sim(r_{p_l},r_{u_k})}{\tau}}} \quad (3)$$

4.4 Domain-specific Representation Adaptation: Adopting Domain-specific Social Patterns

After pre-training, GPSocio have perceived high quality initialized parameters for downstream tasks. To achieve domain-specific information of target dataset, we adapt representations based on the user-assertion graph of target dataset. Given the training data from target domain, the bipartite graph depicts users' posting assertions. As is shown in Figure 2, the left-side nodes represent users and right-side nodes represent assertions, each link represents a user's positing an assertion. We update users' representations by propagating assertion representation to the connected users multiply by propagation weights, following Equation 4 (α represents propagation weight.).

$$r_{u_k} = r_{u_k} + \alpha\Sigma_{a_l \in G^{u_k}_{connected}} r_{a_l} \quad (4)$$

5 Experiments

We evaluate GPSocio over three downstream tasks (Figure 3): user prediction, assertion prediction and sentiment analysis, to showcase GPScio's generalization and knowledge transfer ability. We only include results of second-best compared models for task 1 and 2. Full results are in the Appendix.

5.1 Baselines and Metrics

Baselines: Task 1 and Task 2 compare six graph representation models, including the current state of the art (SOTA), namely Node2Vec, NDM, VGAE, Inf-VAE, MS-HGAT, and RotDiff (SOTA). Among these models, NDM, Inf-VAE, MS-HGAT, and RotDiff are specifically designed for information propagation, while Node2Vec and VGAE are general graph representation techniques. Task 3 compares three text representation techniques: Doc2vec, BERT, and DeBERTa. A detailed introduction is provided in Appendix A.3.

Metrics: Task 1 and task 2 are evaluated over HIT@K and NDCG@K. Task 3 is evaluated over micro-f1 and macro-f1 (Appendix A.5).

5.2 Datasets

In our experiments, we pre-trained GPSocio using the "Russia Ukraine War" dataset, and then tested GPSocio using the "Attack Zelensky," "Brics Superiority," and "Ukraine Nazi Claims" datasets. The data were collected from Twitter using keywords from May 1, 2022, to April 15, 2023. We maintained users with at least 5 assertions and assertions with at least 10 users. The statistics can be found in Table 1. We tested the cosine similarity of topic keywords between the pre-trained dataset and the test dataset, as shown in Table 7 in Appendix A.4. This demonstrates GPSocio's ability to handle domain shift issues.

5.3 Experimental Setting

We set epochs, batch size and temperature to 20, 8 and 0.05, window size to 512 and representation dimension to 768 for task 1 and 2. For each sequence, the first node to the second last node are used as training data and the last node is test data. For task 1, candidate users are all users in target dataset. K is set to $10, 20, 30, 50$ for HIT@K and NDCG@K. For task 2, the candidate assertions are all assertions in the test set of target dataset, which corresponds to the *#test asser* column in Table 1. K is set to $5, 10, 20$ for HIT@K and NDCG@K. For task 3, representation dimension is set to 768. Training data ratio is 8% and test data retio is 92%. The assertion labels are generated by GPT-4 (Achiam et al., 2023) and then double-checked by human. The labeling criteria is in Appendix A.9. We conducted tests to measure the pre-training time of GPSocio using various pre-training data scales, as well as the inference time for downstream tasks. The results can be found in Table 8 and Table 9 in Appendix A.6. These findings demonstrate the stability of GPSocio's pre-training stage and its efficiency in handling downstream tasks.

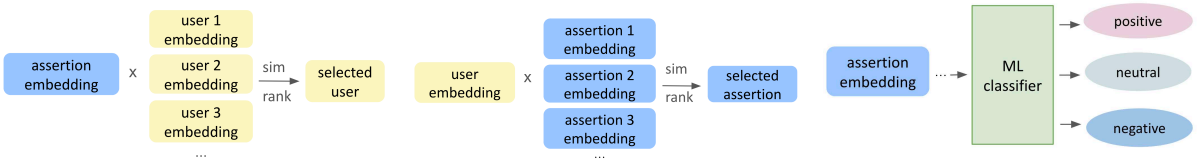
5.4 Task 1: User Prediction Task

Task formulation

We formulate the user prediction task as a representation similarity ranking task. We obtain r_{u_k} of each user $u_k, 1 \leq k \leq n$ and r_{p_l} for each assertion propagation sequence $p_l, 1 \leq l \leq m$. Then we calculate the cosine similarity between r_{p_l} and r_{u_k} following 3 and rank the scores. u_k with highest $sim_{r_{p_l},r_{u_k}}$ will be predicted as the next user.

Overall Performance

To define an appropriate propagation weight, we



(a) User prediction. (b) Assertion prediction. (c) Sentiment analysis.

Figure 3: Three downstream tasks of GPSocio.

Table 1: Statistics of pre-train dataset and target datasets.

	dataset name	keywords	#user	#asser	Avg. Seq.	#test asser	#common users
Pre-train	Russia Ukraine War	Russian Invasion, Russia Ukraine Conflict, Russia Ukraine War	62901	37754	45.81	-	-
Target	Attack Zelensky	KievRegime, Zelensky Regime, ZelenskyWarCriminal	12171	1668	32.67	731	3182
	Brics Superiority	BRICS bloc, BRICS G7, Russia global South	1845	481	18.36	124	1183
	Ukraine Nazi Claims	Naziukraine, Ukronazism, UkraineNazis,DenazifyUkraine, UkraineTerroristState, Ukronazi, Ukroreich, Azov Nazi	3181	633	12.67	221	1684

Table 2: Performance comparison of next user prediction.

Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
Dataset: Attack Zelensky										
VGAE	0.1384	0.1384	0.14021	0.1439	0.1519	0.1366	0.1366	0.1370	0.1377	0.1390
node2vec	0.1426	0.1616	0.1882	0.2110	0.2662	0.0980	0.1027	0.1083	0.1126	0.1217
GPSocio	**0.2500**	**0.2815**	**0.3053**	**0.3393**	**0.3946**	**0.2026**	**0.2106**	**0.2157**	**0.2221**	**0.2310**
Dataset: Brics Superiority										
MS-HGAT	0.0794	0.0966	0.1168	0.1511	0.2083	0.0892	0.0954	0.1017	0.1109	0.1242
rotdiff (SOTA)	0.1069	0.1471	0.1834	0.2295	0.3364	0.075	0.0851	0.0928	0.1015	0.1187
GPSocio	**0.2466**	**0.3007**	**0.3480**	**0.4020**	**0.4966**	**0.1650**	**0.1795**	**0.1895**	**0.1996**	**0.2149**
Dataset: Ukraine Nazi Claims										
MS-HGAT	0.1209	0.1644	0.1996	0.24748	0.3249	0.1111	0.1267	0.1376	0.1505	0.1685
rotdiff (SOTA)	0.1546	0.2103	0.2543	0.3280	0.4697	0.1029	0.1169	0.1262	0.14	0.1629
GPSocio	**0.2916**	**0.3532**	**0.3963**	**0.4353**	**0.5051**	**0.1811**	**0.1967**	**0.2061**	**0.2136**	**0.2250**

tested NDCG@10 over weights ranging from 0 to 2. As is shown in Figure 4 in Appendix A.7.1, the representation propagation can benefit NDCG@10 when weight is from 0.1 to 0.5. We use the best performance weight to tested other metrics. Table 2 shows GPSocio outperforms all baselines. Full result is attached to Appendix A.7.2.

Zero-shot Performance

To test the quality of initialized parameters of GPSocio, we perform experiments when GPSocio is without domain-specific adaptation. Table 3 shows zero-shot representation can already outperform baseline models, and domain-specific adaptation can further improve the performance. This is because the pre-trained model can capture general semantic information and common text connection patterns. Domain-specific adaptation can update representations with target domain knowledge, thus can further improve performance. Full result is in Appendix A.7.3.

Performance after User Filtering

To test if GPSocio relies on common users for good performance, we perform experiments by including common users with ratios ranging from 0% to 100%. When the ratio is not 0% or 100%, we random sample included common users three times and calculate the average. Figure 5 in Appendix A.7.3 didn't show fixed performance variation pattern, which implies GPSocio does not rely on common users. We test all metrics when excluding all common users. Table 3 shows that HIT scores improved after removing users due to the drop of candidate user size. However, there is no fixed variance direction of NDCG@K, which implies GPSocio does not rely on common users for high quality representations. Full result is in Appendix A.7.3.

5.5 Task 2: Assertion Prediction Task

Task formulation

Similar to task 1, we obtain r_{f_k} of user post history $f_k, 1 \leq k \leq n$ and r_{a_l} of each assertion $a_l, 1 \leq l \leq m$. Then we calculate the cosine similarity between r_{f_k} and r_{a_l} following 3 and rank assertions with scores. a_l with highest $sim_{r_{f_k}, r_{a_l}}$ will be the next assertion prediction.

Overall Performance

To define an appropriate propagation weight, we tested GPSocio over propagation weights ranging from 0 to 10. The NDCG@5 of different datasets is

Table 3: Ablation study of GPSocio for the user prediction task.

Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
Dataset: Attack Zelensky										
w adaptation, w/o common users	**0.3018**	**0.3418**	**0.3703**	**0.40803**	**0.4650**	**0.2443**	**0.2546**	**0.2606**	**0.2677**	**0.2769**
GPSocio	0.2500	0.2815	0.3053	0.3393	0.3946	0.2026	0.2106	0.2157	0.2221	0.2310
Dataset: Brics Superiority										
w adaptation, w/o common users	0.2290	**0.3469**	**0.4172**	**0.4966**	**0.6122**	0.1420	0.1720	**0.1875**	**0.2020**	**0.2202**
GPSocio	**0.2466**	0.3007	0.3480	0.4020	0.4966	**0.1650**	**0.1795**	**0.1895**	0.1996	0.2149
Dataset: Ukraine Nazi Claims										
w adaptation, w/o common users	0.2857	0.3479	0.3805	0.4300	**0.5318**	0.1753	0.1913	0.1979	0.2075	0.2237
GPSocio	**0.2916**	**0.3532**	**0.3963**	**0.4353**	0.5051	**0.1811**	**0.1967**	**0.2061**	**0.2136**	**0.2250**

Table 4: Performance of assertion prediction.

Methods	H@5	H@10	H@20	N@5	N@10	N@20
Dataset: Attack Zelensky						
MS-HGAT	0.0642	0.0865	0.1306	0.0627	0.0708	0.0834
rotdiff (SOTA)	0.0685	0.0967	0.1529	0.0459	0.053	0.0641
GPSocio	**0.1305**	**0.2059**	**0.3208**	**0.0847**	**0.1088**	**0.1377**
Dataset: Brics Superiority						
InfVAE	0.1703	0.2102	0.2757	0.1073	0.1195	0.1365
GPSocio	**0.2021**	**0.3184**	**0.5018**	**0.1263**	**0.1640**	**0.2101**
Dataset: Ukraine Nazi Claims						
MS-HGAT	0.1278	0.1594	0.2207	0.1237	0.1351	0.1525
rotdiff (SOTA)	0.1477	0.1998	0.2723	0.1148	0.1316	0.1499
GPSocio	**0.2925**	**0.4059**	**0.4989**	**0.1812**	**0.2178**	**0.2414**

shown in Figure 6 in Appendix A.8.1. The dotted line represents the score when $\lambda = 0$ in Equation 2. We use the best performance weight and tested the remaining metrics. Table 4 shows that after domain-specific adaptation, GPSocio can outperform baseline models over all evaluation metrics. Full results are shown in Appendix A.8.2.

Zero-shot Performance

Table 5: Ablation study of the assertion prediction task.

Methods	H@5	H@10	H@20	N@5	N@10	N@20
Dataset: Attack Zelensky						
w adaptation, w/o common users	**0.1471**	**0.2445**	**0.3871**	**0.0903**	**0.1217**	**0.1576**
GPSocio	0.1305	0.2059	0.3208	0.0847	0.1088	0.1377
Dataset: Brics Superiority						
w adaptation, w/o common users	0.1928	0.2771	0.4277	0.1127	0.1425	0.1798
GPSocio	**0.2021**	**0.3184**	**0.5018**	**0.1263**	**0.1640**	**0.2101**
Dataset: Ukraine Nazi Claims						
w adaptation, w/o common users	**0.4048**	**0.5397**	**0.6032**	**0.2534**	**0.2953**	**0.3111**
GPSocio	0.2925	0.4059	0.4989	0.1812	0.2178	0.2414

We analyze two scenarios: when user representation is not updated through propagation, and when it's excluded from the history sequence. Table 5 shows propagating assertion representations improves performance. Including assertion representation in user history is crucial, as it captures detailed information from past posts, beyond general hashtag usage. Full results are in Appendix A.8.3.

Performance after User Filtering

Similar to task 1, we test NDCG@5 with different ratio of included common users. The results are shown in Figure 7 in Appendix A.8.3. Result shows that GPSocio's doesn't rely on common users for its performance, as the performance doesn't have a clear dropping trend after common users are removed. We test all metrics when removing all common users. The result in Table 5 confirms the finding that GPSocio does not rely on common users. Full result is attached in Appendix A.8.3.

5.6 Task 3: Sentiment Analysis

Task formulation

Table 6: Ablation study of sentiment analysis task.

Methods	Attack Zelensky micro-f1	Attack Zelensky macro-f1	Brics Superiority micro-f1	Brics Superiority macro-f1	Ukraine Nazi Claims micro-f1	Ukraine Nazi Claims macro-f1
random	0.43	0.25	0.4	0.24	0.4	0.33
majority	0.73	0.3	0.72	0.3	0.31	0.22
doc2vec	0.72	0.34	0.75	**0.38**	0.46	0.42
bert	0.73	0.34	**0.75**	0.35	**0.56**	**0.46**
deberta	0.73	0.3	0.72	0.3	0.43	0.38
GPSocio	**0.76**	**0.37**	0.73	0.35	0.48	0.38

We formulate sentiment analysis task as weakly-supervised classification, with very small ratio of data being labeled, namely 8% in our experiment. We use GPSocio representations (with domain-specific adaptation) as assertions' features and train a multi-layer perceptron classifier to classify each assertion based on its sentiment. The categories include "positive", "negative" and "neutral".

Overall performance

We compare GPSocio with five baselines: random, majority, Doc2vec, Bert, and Deberta. In addition to the baselines from Session 5.1, random assigns labels randomly, while majority assigns the most frequent label. Results in Table 6 show GPSocio consistently outperforms random, majority, and Deberta, and performs similarly to other LLMs, demonstrating its strength in capturing sentiment information.

6 Conclusion

We propose GPSocio, a framework pre-trained on large-scale data and fine-tuned with domain-specific information. It extracts assertion propagation sequences from social networks, concatenating each node's semantic identification to create LLM-friendly representations. Unlike existing methods, GPSocio offers a versatile representation for various tasks. Its pre-training transfers general social network knowledge to smaller datasets, mitigating data sparsity challenges.

7 Limitations

GPSocio introduces a pre-training and fine-tuning framework that can handle tasks like predicting the next user, predicting the next assertion, and analyzing sentiment. Although this paper only covers three specific tasks due to space constraints, it suggests that other tasks such as influencer detection and community clustering could also be explored.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Punam Bedi and Chhavi Sharma. 2016. Community detection in social networks. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 6(3):115–135.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–776.

Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S Dhillon. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.

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*.

Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef. 2020. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *applied sciences*, 10(21):7748.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy S Liang, and Jure Leskovec. 2024. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems*, 36.

Tai Huynh, Hien Nguyen, Ivan Zelinka, Dac Dinh, and Xuan Hau Pham. 2020. Detecting the influencer on social networks using passion point and measures of information propagation. *Sustainability*, 12(7):3064.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

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

Thomas N Kipf and Max Welling. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Sanjay Kumar, Abhishek Mallik, and BS Panda. 2023. Influence maximization in social networks using transfer learning via graph-based lstm. *Expert Systems with Applications*, 212:118770.

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

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. 2023. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*.

Wenbin Liu, Bojian Wen, Shang Gao, Jiesheng Zheng, and Yinlong Zheng. 2020. A multi-label text classification model based on elmo and attention. In *MATEC Web of Conferences*, volume 309, page 03015. EDP Sciences.

Costas Mavromatis, Vassilis N Ioannidis, Shen Wang, Da Zheng, Soji Adeshina, Jun Ma, Han Zhao, Christos Faloutsos, and George Karypis. 2023. Train your own gnn teacher: Graph-aware distillation on textual graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 157–173. Springer.

Fabíola SF Pereira, João Gama, Sandra de Amo, and Gina MB Oliveira. 2018. On analyzing user preference dynamics with temporal social networks. *Machine Learning*, 107:1745–1773.

Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu. 2016. *Sentiment analysis in social networks*. Morgan Kaufmann.

Hongliang Qiao, Shanshan Feng, Xutao Li, Huiwei Lin, Han Hu, Wei Wei, and Yunming Ye. 2023. Rotdiff: A hyperbolic rotation representation model for information diffusion prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2065–2074.

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 *Proceedings of the 13th international conference on web search and data mining*, pages 510–518.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. 2022. Ms-hgat: memory-enhanced sequential hypergraph attention network for information diffusion prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4156–4164.

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.

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*.

Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu, and Huanbo Luan. 2018. Neural diffusion model for microscopic cascade prediction. *arXiv preprint arXiv:1812.08933*.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888.

Haiteng Zhao, Shengchao Liu, Ma Chang, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. 2024. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in Neural Information Processing Systems*, 36.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Jichang Zhao, Junjie Wu, Xu Feng, Hui Xiong, and Ke Xu. 2012. Information propagation in online social networks: a tie-strength perspective. *Knowledge and Information Systems*, 32:589–608.

Xinchuang Zhou, Lingtao Su, Xiangju Li, Zhongying Zhao, and Chao Li. 2023. Community detection based on unsupervised attributed network embedding. *Expert Systems with Applications*, 213:118937.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th chinese national conference on computational linguistics*, pages 1218–1227.

A Appendix

The supplementary material is structured as follows:

- Appendix A.1 introduces the data preprocess procedure of GPSocio in detail;
- Appendix A.2 introduces the attention pattern of Longformer;
- Appendix A.3 introduces baseline models for showcased downstream tasks;
- Appendix A.4 discusses the cosine similarity of topics between the pre-training dataset and test datasets;
- Appendix A.5 introduces the evaluation metrics we use in this work;
- Appendix A.6 presents the pre-training time and performance of GPSocio with different pre-training data scales, as well as the inference time on downstream tasks;
- Appendix A.7 includes supplement experimental results of the user prediction task;
- Appendix A.8 includes supplement experimental results of assertion prediction task;

- Appendix A.9 introduces the assertion sentiment labeling rule for task 3.

A.1 Data Preprocess of GPSocio

Assertion propagation sequence generation Given the raw social networks, we generate the assertion propagation sequences as is described in Session 3. Firstly, we group the dataset by assertions and get the publisher and followers for the assertion. Then we order the followers by time, from the most recent follower to the most distant follower. We use p_l to denote the assertion propagation sequence of a_l. The assertion propagation sequence starts with assertion a_l, followed by the publisher $u_l 1$ and the re-ordered followers $u_{l2}, ..., u_{lj}, ...$ p_l can be represented as:

$$p_l = a_l, u_{l1}, u_{l2}, ..., u_{lj}, ... \tag{5}$$

Assertion preprocess The raw content extracted from online social media might contain non-informative information including user names, new line symbols, etc. To exclude the semantically useless information, we conduct text cleaning task, including lowering all characters, excluding user names, excluding assertion format related content (e.g. "RT" in Twitter), and removing hyperlinks. The remaining cleaned content are the preprocessed assertions. We denote the semantic identification of the l_{th} assertion s_{a_l} as:

$$s_{a_l} = clean(text_{a_l}), \tag{6}$$

in which s_{a_l} denotes the semantic identification of a_l, $text_{a_l}$ denotes the raw text of a_l, $clean(.)$ represents the text cleaning operations mentioned in the assertion preprocess procedure.

User preprocess To effectively identify users, we use their used hashtags to represent them. Hashtags often convey users' attitudes of an event and usually contains limited expressions compared with full assertions, thus reduce the difficulty for transformers to process the information. At the same time, the combination of hashtags can nearly uniquely identify each user, which ensures the informativeness of the representation. We denote the semantic identification of the $k_t h$ user s_{u_k} as:

$$s_{u_k} = g_{u_{k1}}, g_{u_{k2}}, ..., g_{u_{kj}}, ... \tag{7}$$

in which $g_{u_{kj}}$ represents the j_{th} hashtag of user u_k.

A.2 Longformer Implementation

To resolve the problem of original Transformers that have $O(n^2)$ time and memory complexity with sequence length of n, Longformer introduces sparsify full self-attention matrix by specifying pairs of input locations attending to one another. This proposed "attention pattern" can scale linearly with the input sequence, which makes it efficient for longer sequences. The proposed "attention pattern" contains four parts: sliding window, dilated sliding window, global attention and linear projections for global attention.

Sliding window Sliding window applies a fixed-size window attention around each token. When multiple stacked layers of such fixed-size window attentions are used, the top layer can have access to all input locations and can build representations with access to information of the entire input. Given a fixed window size w, each token can attend to $\frac{1}{2}w$ tokens on each side, which reduces the computation complexity to $O(n \times w)$. This realizes the linear scale with an input sequence of length n and can generate the receptive field at the top layer with size lw with l layers of transformers.

Dilated sliding window The dilated sliding window can avoid increasing computation with the increase of receptive field. Assuming a fixed dimension d and window size w for all layers, the receptive field is $l \times d \times w$, which can generate a huge receptive field even with small d. In multi-head attention, when the head is set without dilation, it will focus on the local context. When the head is set with dilation, it will focus on the longer context. Therefore, setting different dilation configurations per head can improve the performance by allowing different focus of different heads.

Global Attention The windowed and dilated attentions are not flexible enough to generate task-specific representations. To resolve this problem, "global attention" can be added to pre-selected input locations. A token with global attention attends to all tokens across the sequence and all tokens can have access to the global attention. For example, in classification tasks, [CLS] is used as the global attention, while in QA tasks, global attention is provided on all question tokens.

Linear Projections for Global Attention The original transformers (Vaswani et al., 2017) computes attention scores as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (8)$$

In Longformer, two sets of projections are applied, where Q_s, K_s, V_s are used to compute attention scores of sliding window attention and Q_g, K_g, V_g are used to compute attention scores for the global attention. Introducing two sets of projections provides flexibility to model different types of attention, which is beneficial to resolve downstream tasks.

A.3 Baseline Models

Task 1 & Task 2

We compare our work with six models for user prediction and assertion prediction tasks. This methods includes general graph representation methods: Node2Vec and VGAE, and representation methods designed specifically for the social information diffusion task: NDM, Inf-VAE, MS-HGAT and RotDiff. Based on the structure of input data after data preprocessing, the methods can be divided into graph-only methods, which includes Node2Vec, NDM, VGAE and Inf-VAE, and graph-text methods, which includes MS-HGAT and RotDiff. Same as GPSocio, the embedding dimension of compared models are set to 768. Among the methods, Inf-VAE includes vectors as features, thus we use the Bert (Devlin et al., 2018) to generate 768 dimension embedding of every node as the semantic features. For other methods, we apply Bert to embed the text information into 768 dimensional space. Then we apply an average pooling layer to project the concatenation of graph embedding and text embedding to 768 dimension.

Graph-based methods

- **Node2Vec**(Grover and Leskovec, 2016): A framework learns low-dimensional representations for nodes in a network by optimizing a neighborhood preserving objective.
- **NDM**(Yang et al., 2018): A neural diffusion model which makes relaxed assumptions and employs deep learning techniques including attention mechanism and convolutional network for cascade modeling.
- **VGAE**(Kipf and Welling, 2016b): A graph autoencoder framework consists of graph convolutional layers to represent the network and node attributes, and an inner product decoder to predict the link existence.
- **Inf-VAE**(Sankar et al., 2020): A variational framework that combines a variation graph autoencoder and a co-attention network to jointly model social relations and sequence information.

Sequence-based methods

- **MS-HGAT**(Sun et al., 2022): A Memory-enhanced Sequential Hypergraph Attention Networks that integrates global dependencies of users through friendships and cascade-level interactions.
- **RotDiff**(Qiao et al., 2023): A hyperbolic rotation representation model which combines attention mechanism in the hyperbolic space with extra rotation transformations.

Task 3

We compare the sentiment analysis performance with three embedding techniques: Doc2vec(Le and Mikolov, 2014), Bert(Devlin et al., 2018) and Deberta(He et al., 2020). The representation dimension are set to 768. A Multi-layer Perceptron (MLP) classifier is used for sentiment analysis with 8% sequences are randomly selected as training data and 92% sequences are testing data.

- **Doc2vec**(Le and Mikolov, 2014): A neural network-based approach that learns the distributed representation of documents by a dense vector trained to predict words in the document.
- **Bert**(Devlin et al., 2018): A bidirectional encoder representations from Transformers designed to pretrain deep bidirectional representations from unlabeled text.
- **Deberta**(He et al., 2020) A framework that improves Bert and RoBERTa using disentangled attention mechanism and an enhanced mask decoder.

A.4 Topic Similarities of Pre-train and Test Datasets

Table 7 presents the topic similarity based on the cosine similarity of keywords, illustrating that there is only a minor connection between the pre-training domain and the target domain. From the low similarity shown in the table, we can conclude that GPSocio can successfully deal with domain shift issue.

A.5 Metrics

For task 1: user prediction and task 2: assertion prediction, we compare the performance over two

Table 7: Topic similarity between the pre-train dataset and test datasets.

Tasks	Brics Superiority	Attack Zelensky	Ukraine Nazi Claims
Cosine Similarity	0.3086	0.2857	0.2182

metrics, HIT@K and NDCG@K, as is summarized in (Fayyaz et al., 2020). In this paper, we use H@K and N@K as a short representation of the metrics. For task 3: sentiment analysis, we compare the performance over micro-f1 and macro-f1. **Task 1 & Task 2**

- **HIT@K (H@K)**. Whether any of the top-K recommended items were in the test set for a given user.
- **NDCG@K (N@K)**. NDCG is a widely used metric in information retrieval. It is used to calculate a cumulative score of an ordered set of items.

Task 3

- **micro-f1**. Calculate metrics globally by counting the total true positives, false negatives and false positives.
- **macro-f1**. Takes the F1 scores of each class and averages them, treating all classes equally.

A.6 Pre-train Time and Inference Time of GPSocio

Table 8 displays the time required for computation (over 20 epochs until convergence) for pre-training with various sizes of pre-training data. It also includes the NDCG@10 results for the user prediction task on the "Brics Superiority" topic. The results indicate that GPSocio can still deliver satisfactory performance even when the scale of pre-training data is reduced.

In the upcoming user prediction task, we will compare the computation time for both the baseline models and GPSocio in the downstream task, which includes the embedding time and inference time. Referring to the results in Table 9, we can infer that GPSocio is efficient, especially in cases where model fine-tuning is involved during inference.

A.7 User Prediction Performance

In this part, we include full experimental results of the user prediction task, including different NDCG@10 scores with different representation propagation weights and different ratio of included common users, and the performance comparison with all baseline (or SOTA) models.

A.7.1 Influence of Propagation Weights

Figure 4 shows the NDCG@10 score with different representation propagation weight from 0 to 1 in user prediction task.

When the propagation weight is within a threshold, the performance of user prediction can improve. From the result, we can see the representation propagation will have a positive influence on the task performance in general. For different task datasets, the optimal weights are slightly different but the overall variation trends are the same. To test the optimal performance after domain adaptation, we pick the propagation weights as 0.5, 1, 0.5 for the "Attack Zelensky", "Brics Superiority" and "Ukraine Nazi Claims" datasets correspondingly.

A.7.2 Overall Performance

The evaluation result including all baseline models is shown in Table 10. As can be seen from the result, GPSocio achieves the best performance on all metrics. At the same time, no baseline model can consistently achieve the second-best performance over all metrics among all datasets. The result shows superiority of GPSocio. One key reason for GPSocio's better performance on user prediction over other models is the scale of target dataset is not very large, thus traditional representation techniques without pre-training will suffer from data sparsity and can hardly achieve good performance. This phenomenon shows the superiority of the Graph Foundation Models' frameworks.

A.7.3 Zero-shot Performance and Performance after User Filtering

We compare the NDCG@10 scores with different ratio of common users included in user prediction task. The result is shown in Figure 5.

The result shows that there's no fixed performance change pattern with different ratios of included common users. This shows that GPSocio doesn't rely on common users to generate high quality representations. For ablation study, we include the scenarios when the representations are not adapted and common users are included, representations are not adapted and common users are not included, representations are adapted and common users are not included. The result is shown in

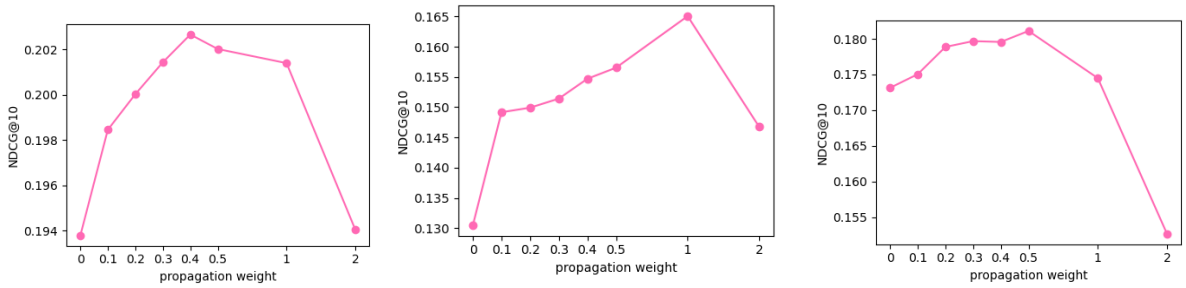
Table 8: The model pre-train time and performance of GPSocio with different pre-train data scales.

time period	#user	#asser	Avg. Seq.	time	NDCG@10
2023/05/01-2023/04/15	62901	37754	45.81	23h31min	0.1650
2023/05/01-2023/03/15	59336	34521	43.17	22h57min	0.1627
2023/05/01-2023/02/15	57939	32154	41.51	22h40min	0.1591
2023/05/01-2023/01/15	56398	30518	39.77	22h17min	0.1572
2023/05/01-2023/12/15	54193	29580	39.18	21h17min	0.1567

Table 9: Topic similarity between the pre-train dataset and test datasets.

Models	Brics	Attack Zelensky	Nazi Claims
VGAE	5min43s	9min43s	7min38s
InfVAE	7min25s	10min28s	8min06s
node2vec	5min27s	9min34s	7min55s
NeuralDiddusionModel	11min11s	23min12s	10min37s
MS-HGAT	12min34s	30min57s	19min41s
rotdiff (SOTA)	3min21s	7min11s	5min34s
GPSocio	1min42s	3min03s	2min39s

Figure 4: NDCG@10 scores with different propagation weights for user prediction task.



(a) Attack Zelensky. (b) Brics Superiority. (c) Ukraine Nazi Claims.

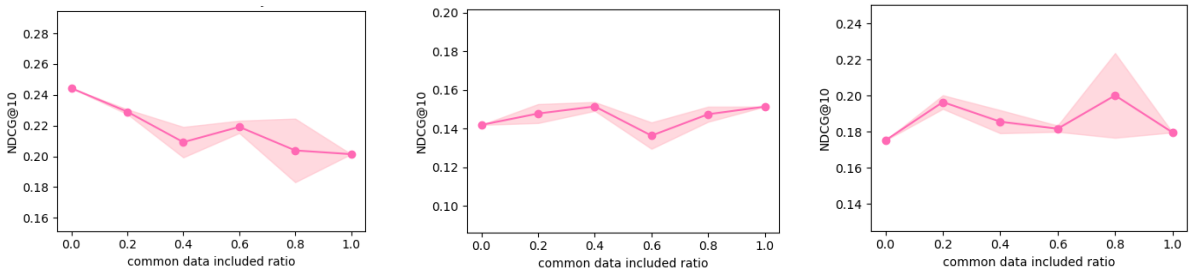
Table 10: Performance comparison of next user prediction.

Dataset: Attack Zelensky										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
VGAE	0.1384	0.1384	0.14021	0.1439	0.1519	0.1366	0.1366	0.1370	0.1377	0.1390
InfVAE	0.0372	0.0539	0.0665	0.0896	0.1362	0.0351	0.0412	0.0453	0.0514	0.0617
node2vec	0.1426	0.1616	0.1882	0.2110	0.2662	0.0980	0.1027	0.1083	0.1126	0.1217
NeuralDiffusionModel	0.1059	0.1462	0.1737	0.2246	0.3538	0.0759	0.0858	0.0917	0.1013	0.1221
MS-HGAT	0.0538	0.0744	0.0930	0.1207	0.1741	0.0507	0.0590	0.0649	0.0722	0.0849
rotdiff (SOTA)	0.0609	0.088	0.1111	0.1424	0.1973	0.0402	0.047	0.0519	0.0577	0.0666
GPSocio	**0.2500**	**0.2815**	**0.3053**	**0.3393**	**0.3946**	**0.2026**	**0.2106**	**0.2157**	**0.2221**	**0.2310**
Dataset: Brics Superiority										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
VGAE	0.0658	0.0767	0.0904	0.1233	0.1808	0.0581	0.0608	0.0637	0.0699	0.0793
InfVAE	0.0688	0.0854	0.1073	0.1583	0.2323	0.0473	0.0525	0.0590	0.0701	0.0877
node2vec	0.1028	0.1583	0.1806	0.2389	0.3528	0.0711	0.0850	0.0898	0.1006	0.1188
NeuralDiffusionModel	0.0909	0.1145	0.1347	0.1717	0.2155	0.0751	0.0808	0.0851	0.0920	0.0992
MS-HGAT	0.0794	0.0966	0.1168	0.1511	0.2083	0.0892	0.0954	0.1017	0.1109	0.1242
rotdiff (SOTA)	0.1069	0.1471	0.1834	0.2295	0.3364	0.075	0.0851	0.0928	0.1015	0.1187
GPSocio	**0.2466**	**0.3007**	**0.3480**	**0.4020**	**0.4966**	**0.1650**	**0.1795**	**0.1895**	**0.1996**	**0.2149**
Dataset: Ukraine Nazi Claims										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
VGAE	0.0737	0.0773	0.1277	0.1385	0.1655	0.0712	0.0721	0.0830	0.0850	0.0894
InfVAE	0.0703	0.0918	0.1094	0.1406	0.2122	0.0692	0.0767	0.0824	0.0894	0.1025
node2vec	0.1426	0.1635	0.1863	0.2053	0.2624	0.0983	0.1034	0.1082	0.1118	0.1211
NeuralDiffusionModel	0.1175	0.1384	0.1593	0.1854	0.2454	0.1090	0.1142	0.1187	0.1235	0.1331
MS-HGAT	0.1209	0.1644	0.1996	0.24748	0.3249	0.1111	0.1267	0.1376	0.1505	0.1685
rotdiff (SOTA)	0.1546	0.2103	0.2543	0.3280	0.4697	0.1029	0.1169	0.1262	0.14	0.1629
GPSocio	**0.2916**	**0.3532**	**0.3963**	**0.4353**	**0.5051**	**0.1811**	**0.1967**	**0.2061**	**0.2136**	**0.2250**

Table 11.

As can be seen from the result, except from the representation generated with full pipeline of GPSocio, the best performance shows in group when

Figure 5: NDCG@10 scores with different ration of included common users for next user prediction task.



(a) Attack Zelensky. (b) Brics Superiority. (c) Ukraine Nazi Claims.

Table 11: Ablation study of GPSocio for the user prediction task. One for analyzing the effect of the adaptation stage, the other for studying the influence of existence of common users with pre-training stage.

Dataset: Attack Zelensky										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
w/o adaptation, w common users	0.2275	0.2487	0.2622	0.2847	0.3303	0.1938	0.1991	0.2020	0.2061	0.2136
w/o adaptation, w/o common users	0.2841	0.3118	0.3310	0.3610	0.4180	0.2327	0.2398	0.2439	0.2496	0.2589
w adaptation, w/o common users	**0.3018**	**0.3418**	**0.3703**	**0.40803**	**0.4650**	**0.2443**	**0.2546**	**0.2606**	**0.2677**	**0.2769**
GPSocio	0.2500	0.2815	0.3053	0.3393	0.3946	0.2026	0.2106	0.2157	0.2221	0.2310
Dataset: Brics Superiority										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
w/o adaptation, w common users	0.1824	0.1959	0.2196	0.2770	0.3615	0.1305	0.1335	0.1378	0.1481	0.1627
w/o adaptation, w/o common users	0.2290	0.2970	0.3515	0.4263	0.5601	0.1302	0.1502	0.1625	0.1769	0.1959
w adaptation, w/o common users	0.2290	**0.3469**	**0.4172**	**0.4966**	**0.6122**	0.1420	0.1720	**0.1875**	**0.2020**	**0.2202**
GPSocio	**0.2466**	0.3007	0.3480	0.4020	0.4966	**0.1650**	**0.1795**	**0.1895**	0.1996	0.2149
Dataset: Ukraine Nazi Claims										
Methods	H@10	H@20	H@30	H@50	H@100	N@10	N@20	N@30	N@50	N@100
w/o adaptation, w common users	0.2402	0.2834	0.3121	0.3614	0.4251	0.1731	0.1838	0.1894	0.1988	0.2088
w/o adaptation, w/o common users	0.2659	0.3239	0.3536	0.3861	0.4710	0.1723	0.1869	0.1927	0.1989	0.2123
w adaptation, w/o common users	0.2857	0.3479	0.3805	0.4300	**0.5318**	0.1753	0.1913	0.1979	0.2075	0.2237
GPSocio	**0.2916**	**0.3532**	**0.3963**	**0.4353**	0.5051	**0.1811**	**0.1967**	**0.2061**	**0.2136**	**0.2250**

representations are adapted and the common users are excluded. It shows the effectiveness of domain adaptation and reveals GPSocio doesn't rely on common users for high representation quality, as removing common users doesn't bring a consistent performance drop.

A.8 Assertion Prediction Performance

In this part, we include full experimental results of the assertion prediction task, including different NDCG@5 scores with different representation propagation weights and different ratio of included common users, and the performance comparison with all baseline (or SOTA) models.

A.8.1 Influence of Propagation Weights

Figure 6 shows the NDCG@5 score with different representation propagation weight from 0 to 10 in assertion prediction task.

Similar to the user prediction task, domain-specific adaptation will have a positive influence on assertion prediction performance when propagation weight is within a threshold. For different task datasets, the optimal weights are slightly different but the overall variation trends are the same. To test the optimal performance after domain adaptation, we pick the propagation weights as 1, 3, 3 for the "Attack Zelensky", "Brics Superiority" and "Ukraine Nazi Claims" datasets correspondingly.

A.8.2 Overall Performance

The evaluation result including all baseline models is shown in Table 12.

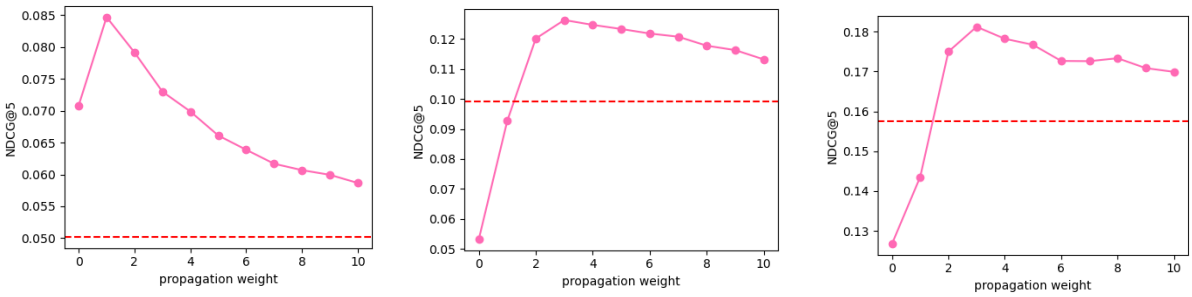
As can be seen from the result, GPSocio achieves the best performance on all metrics among all datasets. At the same time, no baseline model can consistently achieve the second-best performance over all metrics among all datasets. The result shows GPSocio's superiority in the assertion prediction task.

A.8.3 Zero-shot Performance and Performance after User Filtering

We compare the NDCG@5 scores with different ratio of common users included in assertion prediction task. The result is shown in Figure 7.

The result shows that there's no fixed performance change pattern with different ratios of included common users. This shows that GPSocio doesn't rely on common users to generate high

Figure 6: NDCG@5 scores with different propagation weights for assertion prediction task. The red dotted line represents the score when users' representation are not considered.

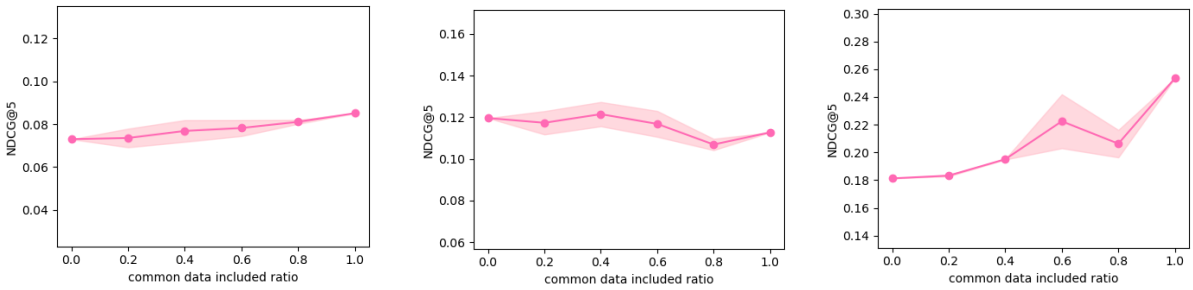


(a) Attack Zelensky. (b) Brics Superiority. (c) Ukraine Nazi Claims.

Table 12: Performance comparison of different propagation prediction models.

Dataset: Attack Zelensky						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
VGAE	0.0142	0.0245	0.0614	0.0081	0.0107	0.01779
InfVAE	0.0256	0.0477	0.1399	0.0101	0.0156	0.0336
node2vec	0.0350	0.0719	0.1472	0.0197	0.0291	0.0439
NeuralDiffusionModel	0.0446	0.0728	0.1277	0.0213	0.0284	0.0391
MS-HGAT	0.0642	0.0865	0.1306	0.0627	0.0708	0.0834
rotdiff (SOTA)	0.0685	0.0967	0.1529	0.0459	0.053	0.0641
GPSocio	**0.1305**	**0.2059**	**0.3208**	**0.0847**	**0.1088**	**0.1377**
Dataset: Brics Superiority						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
VGAE	0.0909	0.1364	0.2727	0.0586	0.0700	0.0974
InfVAE	0.1703	0.2102	0.2757	0.1073	0.1195	0.1365
node2vec	0.1094	0.1875	0.3594	0.0433	0.0627	0.0949
NeuralDiffusionModel	0.0687	0.1164	0.2358	0.0506	0.0662	0.0959
MS-HGAT	0.0874	0.1151	0.1762	0.0882	0.0981	0.1155
rotdiff (SOTA)	0.0905	0.1216	0.2675	0.0724	0.0824	0.1139
GPSocio	**0.2021**	**0.3184**	**0.5018**	**0.1263**	**0.1640**	**0.2101**
Dataset: Ukraine Nazi Claims						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
VGAE	0.0438	0.075	0.1969	0.0244	0.0322	0.0557
InfVAE	0.1044	0.1313	0.1925	0.0514	0.0582	0.0707
node2vec	0.0770	0.1304	0.2275	0.0519	0.0695	0.0937
NeuralDiffusionModel	0.0562	0.1066	0.2306	0.0325	0.0450	0.0693
MS-HGAT	0.1278	0.1594	0.2207	0.1237	0.1351	0.1525
rotdiff (SOTA)	0.1477	0.1998	0.2723	0.1148	0.1316	0.1499
GPSocio	**0.2925**	**0.4059**	**0.4989**	**0.1812**	**0.2178**	**0.2414**

Figure 7: NDCG@5 scores with different propagation weights for assertion prediction task. The red dotted line represents the score when users' representation are not considered.



(a) Attack Zelensky. (b) Brics Superiority. (c) Ukraine Nazi Claims.

quality representations. For ablation study, we include the scenarios when the representations are not adapted and common users are included, representations are not adapted and common users are not included, representations are adapted and common users are not included, user representations are not added and common users are included, user representations are not added and common users are not included. The result is shown in Table 13.

As can be seen from the result, except from the representation generated with full pipeline of GPSocio, the best performance shows in group when representations are adapted and the common users are excluded. It shows the effectiveness of domain adaptation and reveals GPSocio doesn't rely on common users for high representation quality, as removing common users doesn't bring a consistent performance drop. We also discover that including assertion representation is very important for getting a high quality user history sequence representation. This is because the semantic identification of a user only includes the hashtags, while assertions include more detailed semantic information, thus is more important for defining the overall representation of a user history sequence representation.

A.9 Sentiment Label Criteria

We check the rationality of the GPT-4 labeled assertions of being positive / negative / neutral considering the criteria below. **Positive polarized**

- **Explicit language** Including explicit positive words, e.g. great, happy, etc.
- **Subjective implication** When the user conveys a his / her personal positive emotion, e.g., showing hope to an event or a political party.

Negative polarized

- **Explicit language** Including explicit positive words, e.g. terrible, awful, etc.
- **Subjective implication** When the user conveys a his / her personal negative emotion, e.g., showing disappoint to an event or some political decisions.
- **Sarcasm and irony** When the content conveys sarcasm or irony information, it will be considered as negative polarized.

Neutral

- **Objectivity** When the statement is objective and conveys no emotion polarization, e.g. objective news, it will be considered as neutral.

For the assertions we find hard to label, we will exclude them from the sentiment analysis dataset.

Table 13: Ablation study of GPSocio for the assertion prediction task. One for analyzing the effect of the adaptation stage, the other for studying the influence of existence of common users with pretraining stage.

Dataset: Attack Zelensky						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
w/o adaptation, w common users	0.1013	0.1525	0.2349	0.0708	0.0873	0.1079
w/o user rep., w common users	0.0851	0.1582	0.2695	0.0502	0.0736	0.1015
w/o adaptation, w/o common users	0.1076	0.1679	0.2641	0.0760	0.0951	0.1192
w/o user rep., w/o common users	0.0978	0.1801	0.3052	0.0565	0.0828	0.1142
w adaptation, w/o common users	**0.1471**	**0.2445**	**0.3871**	**0.0903**	**0.1217**	**0.1576**
GPSocio	0.1305	0.2059	0.3208	0.0847	0.1088	0.1377
Dataset: Brics Superiority						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
w/o adaptation, w common users	0.0764	0.1304	0.2773	0.0532	0.0707	0.1071
w/o user rep., w common users	0.1727	0.2832	0.4900	0.0991	0.1345	0.1862
w/o adaptation, w/o common users	0.0964	0.1566	0.2952	0.0685	0.0892	0.1229
w/o user rep., w/o common users	0.1807	0.2651	0.3916	0.0975	0.1261	0.1574
w adaptation, w/o common users	0.1928	0.2771	0.4277	0.1127	0.1425	0.1798
GPSocio	**0.2021**	**0.3184**	**0.5018**	**0.1263**	**0.1640**	**0.2101**
Dataset: Ukraine Nazi Claims						
Methods	H@5	H@10	H@20	N@5	N@10	N@20
w/o adaptation, w common users	0.1723	0.2744	0.3628	0.1267	0.1592	0.1812
w/o user rep., w common users	0.2608	0.3700	0.4683	0.1576	0.1932	0.2178
w/o adaptation, w/o common users	0.2143	0.2937	0.3492	0.1548	0.1783	0.1783
w/o user rep., w/o common users	0.4048	0.5317	0.5714	0.2495	0.2907	0.3000
w adaptation, w/o common users	**0.4048**	**0.5397**	**0.6032**	**0.2534**	**0.2953**	**0.3111**
GPSocio	0.2925	0.4059	0.4989	0.1812	0.2178	0.2414