

Temporally Like-minded User Community Identification through Neural Embeddings

Hossein Fani*

University of New Brunswick
Fredericton, NB, Canada
hfani@unb.ca

Ebrahim Bagheri*

Ryerson University
Toronto, ON, Canada
bagheri@ryerson.ca

Weichang Du

University of New Brunswick
Fredericton, NB, Canada
wdu@unb.ca

ABSTRACT

We propose a neural embedding approach to identify temporally like-minded user communities, i.e., those communities of users who have similar temporal alignment in their topics of interest. Like-minded user communities in social networks are usually identified by either considering explicit structural connections between users (link analysis), users' topics of interest expressed in their posted contents (content analysis), or in tandem. In such communities, however, the users' rich temporal behavior towards topics of interest is overlooked. Only few recent research efforts consider the time dimension and define like-minded user communities as groups of users who share not only similar topical interests but also *similar temporal behavior*. Temporally like-minded user communities find application in areas such as recommender systems where relevant items are recommended to the users at the right time. In this paper, we tackle the problem of identifying temporally like-minded user communities by leveraging unsupervised feature learning (embeddings). Specifically, we learn a mapping from the user space to a low-dimensional vector space of features that incorporate both topics of interest and their temporal nature. We demonstrate the efficacy of our proposed approach on a Twitter dataset in the context of three applications: news recommendation, user prediction and community selection, where our work is able to outperform the state-of-the-art on important information retrieval metrics.

KEYWORDS

Community Detection, Neural Embedding, Social Network Analysis

1 INTRODUCTION

User community detection is the process of finding latent communities of users whose members share higher intra-cluster similarity compared to inter-cluster similarity. Community level methods have shown to be more effective than their user level counterparts in some application areas, e.g., in social recommender systems [16] and information diffusion modeling [14], just to name a few.

*Laboratory for Systems, Software and Semantics (LS³), Ryerson University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'17, November 6–10, 2017, Singapore, Singapore

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-4918-5/17/11...\$15.00

<https://doi.org/10.1145/3132847.3132955>

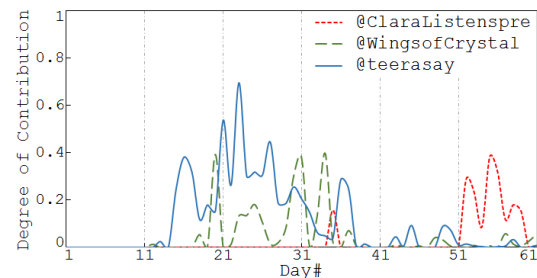


Figure 1: Different temporal inclination of three Twitter users with respect to the ‘War in Afghanistan’ topic.

Early methods for community detection were primarily based on the homophily principle [17] where densely connected groups of users imply the existence of a community. To this end, primitive graph structures such as components and cliques were considered to be the representation of user communities [11]. However, these methods fall short when the communities of interest need to take users' interest into account. This is mainly due to the fact that like-minded users are not necessarily always connected to each other because in some cases explicit social network connections do not mean that the two users share common interests and can merely denote some social relation such as kinship. For this reason, irrespective of the social network connections (structure), content (topic)-based methods utilize the topical similarity of user-generated content to detect like-minded communities of users [22, 27, 30]. There have also been work that consider the integration of both social connections and users' content for identifying user communities [8, 14, 37].

Several researchers have already explored the dynamic nature of user's interests on social networks [29, 35]. A user may become interested in a new topic, lose interest in a topic, or change degree of preference toward a topic over time [25]. For example, let us look at two Twitter users @teerasay and @WingsofCrystal and how their degree of interest towards the ‘War in Afghanistan’ topic changes from mid November to the end of December 2010, as shown in Figure 1. These two users seem to share a similar behavioral pattern towards this topic. However, another user @ClaraListenspre does not start posting about the same topic until much later in late December of the same year. While the three users share a similar interest, they do not exhibit this interest in similar time intervals. Contemporary community detection methods, such as the aforementioned methods, would cluster all these three users in the same community because they do not incorporate the temporal nature of users' topics of interest. This renders it difficult for applications such as news recommender systems to generate recommendations

that are temporally sensitive. If the three mentioned users were identified as members of the same community, they would be recommended the same news articles at the end of December on the given topic while @teerasay and @WingsofCrystal have already covered this topic in November and have now moved on and as a result are not interested in it any longer but @ClaraListenspre has just become interested in the topic.

More recently, two works have considered the issue of time-aware (temporal) like-minded user community detection [9, 13]. To include the temporal component, Fani et al. [9] have proposed a multivariate time series representation of users in topic and time spaces, while Hu et al. [13] have devised a unified probabilistic generative model of both topics and users. In this paper, while we follow the same underlying premise about temporality in like-minded user community detection, we introduce a time-aware topic-driven distributional representation (embedding) of users.

The concept of distributional representation has been explored well beyond the domain of computational linguistics in a number of different disciplines including graph analysis (node2vec [12] or DeepWalk [23]), Genetics [3] and video analysis (frame2vec) [26], just to name a few. In social network analysis, successful user embeddings into low-dimensional vector spaces have been attempted such as author2vec [15] in citation networks as well as in tweet recommendation [31]. Basically, such user embedding models propose that users with similar topics of interest should have similar embeddings. The main objective of our work is to identify like-minded user communities whose members exhibit temporally similar behaviour toward similar topics of interest. Specifically, we would like to embed those users who are interested in similar topics at similar points in time, e.g., @teerasay and @WingsofCrystal, distant from those who have similar interest towards the same topics but in different time intervals, e.g., @ClaraListenspre. Our proposed temporal topic-driven user embedding model in this paper represents a step forward with this respect. To this end, we build documents whose elements are users not words. We extend the concept of co-occurrence of words in documents to users and propose a new form of context for users such that two users co-occur if they show the same interest toward the same topics in similar time intervals. In order to illustrate the effectiveness of our proposed approach, we perform experiments on a Twitter dataset from the last two months of 2010. We evaluate our work on personalized news recommendation, user prediction and community selection. The experimental results show that our proposed approach outperforms the state of the art. Our main contributions in this paper are as follows:

- (1) We propose a novel temporal user embedding model which learns low-dimensional user representations such that users who exhibit similar temporal behaviour toward similar topics are closer in vector space.
- (2) We identify temporal like-minded user communities which are both topically and temporally cohesive based on our user embeddings.
- (3) We demonstrate the performance of our temporally like-minded user communities in the context of personalized news recommendation, user prediction and community selection compared to the state of the art.

The rest of the paper is organized as follows: we first present the related works in Section 2, then we continue with the problem definition. We propose our approach in Section 4. The experimental setup and evaluation is described in Section 5, followed by concluding remarks in Section 6.

2 RELATED WORK

The related works to this paper are largely centered around two areas of user community detection and distributional semantics.

2.1 User Community Detection

User community detection is one of the well-explored research topics in social network analysis; ranging from link (topology)-based community detection methods, which rely only on the network structure of the social network graph, to content (topic)-based approaches, which mainly focus on information content generated by the users. A particularly large number of more effective approaches have been proposed which integrate both the network structure (links) and content to improve community detection performance [8]. All these works assume that the user's topics of interest remain stable across time. However, very few consider the notion of temporality in users' topics of interest [9, 13], particularly in online social networks such as Twitter.

From among the work that consider temporality, Hu et al. [13] have proposed a probabilistic generative model, namely Group Specific Topics-over-Time (GrosToT), to simultaneously identify both user communities and topics. The members of the identified communities have temporal similarity with respect to the identified topics. The generative model has ex-ante knowledge about the number of topics and communities. Each community is associated with a distribution over topics according to a Dirichlet distribution (community-topic distribution), and for each specific topic within the community, the temporal variation is obtained by another Dirichlet distribution (community-topic-time distribution). Users belong to communities based on a Dirichlet distribution (user-community distribution). A user selects a community based on a multinomial distribution over her user-community distribution and generates documents (tweets) about each topic based on her selected community-topic distribution in each time interval according to the community-topic-time distribution. GrosToT is a mixture model in which a user is interested in all the topics in all time intervals and is member to all communities but with different probabilities.

Contrary to the unified generative model, Fani et al. [9] have proposed a framework to extract topics and user communities in tandem. To extract user communities, they first build a multivariate time series for each user in topic space within the time dimension and, then, employ cross-correlation similarity of users' time series to capture the respective users' temporal and topical similarities. Finally, a graph-based clustering method is applied on a weighted graph whose nodes are users and the weights are the users' similarity. While both of these works sketch different architectures, they have shown competitive performance in modelling like-minded user communities and to the best of our knowledge are the state of the art in this respect; hence, we use them as our baselines.

2.2 Distributional Semantics

The idea of distributional semantics states that words that occur in similar contexts are semantically similar. Its recent neural model implementation, named word2vec by Mikolov et al. [18], approximates the semantics of a word with a dense low-dimensional vector (embedding) so that the semantic similarity of words can be measured in terms of geometric distance between the respective vectors. The success of word2vec has extended beyond computational linguistics. For example, node2vec [12] and DeepWalk [23] are inspired by the skip gram model and employ a second order random walk to sample network neighborhoods for users (nodes) in the social network structure. They output user vector representations (embedding) that maximize the likelihood of preserving network neighborhoods of users. While node2vec is amodal which only relies on the social network graph structure, author2vec [15] is bimodal. It augments the graph with user-generated textual contents to learn better user embeddings. Author2vec includes content-info and link-info neural models. The content-info model predicts whether a given user has authored a given text and the link-info model predicts whether a given pair of users are connected. Also, Benton et al. [4] have proposed a variant of the generalized canonical correlation analysis (GCCA) to learn a single joint user embedding from each of the given sources of information, namely, content and network structure. None of the proposed user embedding models take the time dimension into consideration. Although Benton et al. offer the opportunity to integrate different information types, how temporality, which can be considered to be an aspect as opposed to a new information type, can be integrated is not clear. In this paper, we propose a novel way to incorporate user temporal content into a distributional representation.

There are also some application specific research that employ an existing technique, in one way or another, to produce user embeddings for downstream tasks such as gender prediction [7], sarcasm detection [2], or scholarly microblog recommendation [31]. However, to the best of our knowledge, no approach investigates the application of user embeddings for temporal like-minded user community detection purposes. The main objective of the user embedding in this paper is to accurately identify temporally like-minded user communities.

3 PROBLEM DEFINITION

Our goal is to identify like-minded user communities whose members exhibit similar temporal dispositions towards similar topics. Here, we provide a formal statement of the problem after which we propose our approach in detail in the next section. We view the problem of like-minded user community detection as an instance of the set partitioning task on a set of users \mathcal{U} . A partition \mathbb{P} of the set \mathcal{U} of all users is a set of nonempty subsets of \mathcal{U} as communities such that every user $u \in \mathcal{U}$ is in exactly one of these communities. Notationally, $\mathbb{P} = \{\mathcal{C} : \mathcal{C} \subseteq \mathcal{U}, |\mathcal{C}| \geq 1\}$ such that $\forall \mathcal{C}_i, \mathcal{C}_j \in \mathbb{P} : \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ and $\bigcup_{\mathcal{C} \in \mathbb{P}} \mathcal{C} = \mathcal{U}$. Since we do not consider a set with one user as a community, we relax the partition definition by assuming $|\mathcal{C}| \geq 2$ and drop the last union condition; i.e., $\mathbb{P}^* = \mathbb{P} \setminus \{\mathcal{C} : |\mathcal{C}| = 1\}$. The goal of like-minded user community detection is to infer \mathbb{P}^* such that highly similar users are in the same community \mathcal{C} , yet users of high dissimilarity are in different

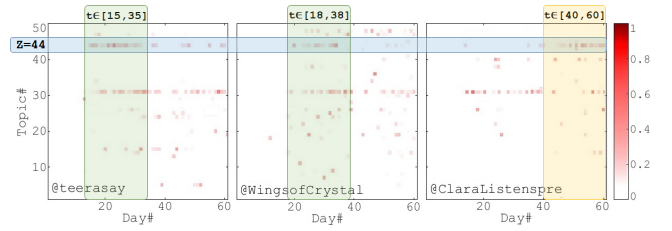


Figure 2: Points of temporal interest (PoTI) projected on the user dimension for three sample Twitter users.

communities \mathcal{C}_i and $\mathcal{C}_{j \neq i}$. In our work, we consider two users to be similar if they show similar temporal inclination towards a set \mathcal{Z} of possible topics.

4 PROPOSED APPROACH

Our proposed like-minded community detection method seeks to find \mathbb{P}^* with respect to the temporal topic-based sense of user similarity, defined in the previous section. The approach works through three pipelined phases: temporal topic-based user modeling, user vector representation (embedding), and user community detection. In the following, we describe the details of each step.

4.1 Temporal Topic-based User Modeling

Our work relies on users' behavior towards a set of topics within time period T . To incorporate both users' topics of interest and temporality, for each user $u \in \mathcal{U}$, we model her inclination towards each topic $z \in \mathcal{Z}$ at each time interval $1 \leq t \leq L$ through a matrix. The stacking of all user matrices will generate a cuboid denoted as *points of temporal interest* (PoTI). An entry in PoTI shows how much a user $u \in \mathcal{U}$ is interested in a topic $z \in \mathcal{Z}$ in time interval $1 \leq t \leq L$.

Definition 4.1. Points of Temporal Interest (PoTI). Let \mathcal{U} be a set of users, \mathcal{M} be the users' posts corpus, \mathcal{Z} be a set of topics, and T be a time period broken down into L intervals, *points of temporal interest* (PoTI) is a three dimensional matrix (cuboid) $\mathcal{U} \times \mathcal{Z} \times T = \{y_t^u[z]\}$ where $u \in \mathcal{U}$, $z \in \mathcal{Z}$ and $1 \leq t \leq L$ whose three dimensions correspond to users, topics and time intervals, respectively and the value $y_t^u[z]$ is the degree of u 's interest in topic z at time t .

Slices of PoTI made through the user dimension can be visualized in the form of heatmaps as shown for three sample Twitter users @teerasay, @WingsofCrystal and @ClaraListenspre in Figure 2. In this figure, the y-axis represents the topic indices, the x-axis denotes the time intervals, and the cell values show the degree of contribution of the user to that topic. For example, users @teerasay and @WingsofCrystal heavily post about 'War in Afghanistan' (z_{44}) in similar time intervals from mid to the end of November whereas user @ClaraListenspre did not react to this topic until the middle of the following month. From Figure 2, it is evident that non-temporal topic-based approaches would group all these three users in the same community and consider them like-minded, because they are interested in the shared topic z_{44} . However, user @ClaraListenspre can be considered to be dissimilar from the others because the period of time during which she reacts to z_{44} is not the same as the other users.

To instantiate PoTI, we need to find i) a set of topics that have been observed in time period T , i.e., \mathbb{Z} , and ii) each user's degree of interest at time t towards each topic $z \in \mathbb{Z}$, i.e., $y_t^u[z]$. The set of possible topics can be derived by extracting the topics available in the collection of users' posts using various existing topic detection methods in the literature including topic modeling techniques such as latent Dirichlet allocation (LDA) [5] as suggested in [33, 34]. In order to identify the set of topics, we view all posts of a user $u \in \mathbb{U}$ for each time interval t , i.e., $m_t^u \in \mathbb{M}$, as a single document. A document m is a vector of N nonnegative integers, where the i^{th} number shows the occurrence frequency of the i^{th} term. N is the size of the unique terms in all posts. Topic z is a vector of N real numbers in range $[0, 1]$, summing to 1. The i^{th} number shows the participation score of the i^{th} term in forming that topic. Collectively, $\mathbb{Z} = \{z \in [0, 1]^N : \|z\|_1 = 1\}$ is the set of all topics. Topic distribution of a document is a function $f : \mathbb{M} \rightarrow [0, 1]^{\mathbb{Z}}; \forall m \in \mathbb{M}, \|f(m)\|_1 = 1$. Intuitively, f maps a document to a set of topics where $f(m)_z$ is the score of topic z for document m . Given the number of topics is known *a priori*, LDA produces a topic set \mathbb{Z} where each of its topics z is the multinomial distribution of terms specific to topic z . Also, the topic distribution function f is defined as a Dirichlet distribution with parameter α ; notationally, $f(m)_z \sim \text{Dir}(\alpha)_z$.

4.2 User Vector Representation (Embedding)

The key contribution of this paper is to learn user vector representations from users' topics of interest with the expectation that temporally like-minded users end up closer to each other in the vector space. We hypothesize that an appropriate embedding method would bring significant performance into our main downstream task of like-minded user community detection compared to the state of the art. To build user embeddings, we first formally formulate what we mean by a like-minded pair of users. Then, we propose an embedding method which preserves pairwise like-minded proximity of the users through maximizing the likelihood that two like-minded users stay close to each other in vector space.

4.2.1 User Like-minded Context Model. In our approach, users would be considered to be like-minded if they share similar temporal and topical interest. More formally, the more two user u_1 and u_2 share instances of $y_t^{u_1}[z] \approx y_t^{u_2}[z]$ in the PoTI for topics $z \in \mathbb{Z}$ across different time intervals $1 \leq t \leq L$, the more similar they would be. To illustrate an example, let us view Figure 3a where we show a region of PoTI for our three sample Twitter users with $\{u_1 = \text{@teerasay}, u_2 = \text{@WingsofCrystal}, u_3 = \text{@ClaraListenspre}\} \times \{z_{40}, \dots, z_{45}\} \times \{t_{20}, \dots, t_{30}\}$. As defined earlier in Section 4.1, the values of each cell of a PoTI are topic distribution scores normalized in the range of $[0, 1]$. As seen in the figure, there are very few, if any, corresponding cells that share the same value; therefore, it is very difficult to find users that have the exact same topic interest, in the exact same time, to the exact same extent. However, it is possible to relax the similarity condition to consider corresponding cells that are within a certain range to be similar. We refer to this relaxed condition as *the condition of homogeneity*. For the purposes of illustration, here we assume cells with values less than 0.1 represent insignificant levels

of interest, signified by gray, and hence, set our condition of homogeneity to encompass values within the range $[0.1, 1]$. As shown in Figure 3a, the first two users @teerasay and @WingsofCrystal share four similar regions in which instances of similar temporal-topical interest happen: $\{u_1, u_2\} \times \{z_{43}, z_{44}\} \times \{t_{20}, t_{22}\}$, $\{u_1, u_2\} \times \{z_{40}\} \times \{t_{22}, t_{27}\}$, $\{u_1, u_2\} \times \{z_{44}, z_{45}\} \times \{t_{26}\}$, $\{u_1, u_2\} \times \{z_{44}\} \times \{t_{23}, t_{24}, \dots, t_{30}\}$, $\{u_1, u_2\} \times \{z_{40}, z_{45}, z_{46}\} \times \{t_{22}\}$, which indicate that @teerasay and @WingsofCrystal are like-minded both in time and topic within these four regions. However, @ClaraListenspre only shares one region with @WingsofCrystal, i.e., $\{u_2, u_3\} \times \{z_{43}\} \times \{t_{24}, t_{25}\}$, and none with @teerasay. Indeed, @ClaraListenspre is interested in the same set of topics but in different time intervals $\{t_{40}, \dots, t_{60}\}$ as shown in Figure 3b. So, within the context of our work, we are interested in a model that would determine @teerasay and @WingsofCrystal to be similar users and distant from @ClaraListenspre.

Now, in order to be able to develop a notion of similarity between users, we learn user embeddings such that users that have a maximal number of shared regions are placed close to each other in the embedding space. The shared regions; therefore, present a sense of context for our users. For example, in Figure 3a, the four regions shared by the two users @teerasay and @WingsofCrystal provide a context for incorporating the two users in the embedding space. In order to be able to use the shared regions as context, we first introduce a deterministic algorithm to find all these regions as an input to our embedding method. We wish to learn an embedding for users such that the representation of a specific target user can be determined by other users in the same context. Let us now formalize this process:

Definition 4.2. Region of Like-mindedness (RoL). A three-dimensional subspace of PoTI, such as R , is defined to be a region of like-mindedness (RoL) iff (i) all the values in this subspace are *equal* with respect to a certain condition of homogeneity c ; notationally, $\forall y, y' \in R; c(y) = c(y')$ and (ii) it is *maximal* such that there exists no other region of like-mindedness such as R' which subsumes R .

To find all regions of like-mindedness, RoLs, in PoTI, we adopt a similar strategy to [36] where subspace submatrices are mined from three-dimensional gene-sample-time gene expression microarrays. First, we find the RoLs in user and topic dimensions at each time interval. The output is two-dimensional (2-d) RoLs indexed by the time interval $1 \leq t \leq L$, i.e., RoL_t . Then, we merge RoL_t of different time intervals to build the required RoLs. The details are as follows:

Finding 2-d RoLs for the time interval t . Given the PoTI and a condition c for homogeneity, let $z_i, z_j \in \mathbb{Z}$ be any two topic rows with $1 \leq i \leq j \leq |\mathbb{Z}|$ in the time interval t and let $y_t^u[z_i]$ and $y_t^u[z_j]$ be the degree of interest of user u with respect to topics z_i and z_j where $u \in \mathbb{U}$. We define $\mathbb{U}_{z_i, z_j}^t(c)$ to be the set of users whose interest towards z_i and z_j satisfies the condition of homogeneity c . $\mathbb{U}_{z_i, z_j}^t(c)$ is maximal with regard to c if we cannot add another user to it while respecting c . Given $\mathcal{U}^t = \{\mathbb{U}_{z_i, z_j}^t(c) : z_i, z_j \in \mathbb{Z}\}$, we construct a directed multigraph $\mathcal{G}^t = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \mathbb{Z}$ and $\mathbb{E} = \mathcal{U}^t$, i.e., for each $\mathbb{U}_{z_i, z_j}^t(c) \in \mathcal{U}^t$ there exists a directed edge ($z_i \rightarrow z_j$) labeled with the set of users in $\mathbb{U}_{z_i, z_j}^t(c)$. For example, Figure 4a shows the multigraph \mathcal{G}^{22} constructed from Figure 3a for time t_{22} where the condition of homogeneity c is satisfied if

(a)		t_{20}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}
u_1 @teerasay	z_{40}			0.2					0.1			
	z_{41}											
	z_{42}											
	z_{43}	0.2	0.2	0.3								
	z_{44}	0.2	0.5	0.3	0.7	0.4	0.3	0.4	0.5	0.2	0.2	0.3
z_{45}							0.2					
u_2 @WingsofCrystal	z_{40}	0.4		0.1	0.6				0.2	0.2	0.2	0.3
	z_{41}			0.1					0.2			
	z_{42}								0.2			
	z_{43}	0.3		0.1		0.9	0.5		0.2		0.4	
	z_{44}	0.4		0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.8	0.3
z_{45}				0.5	0.2	0.5	0.2	0.2	0.4	0.2		
u_3 @ClaraListenspre	z_{40}					0.4	0.2	0.8				
	z_{41}											
	z_{42}											
	z_{43}					0.1	0.3	0.8				
	z_{44}											
z_{45}												

(b)		t_{50}	t_{51}	t_{52}	t_{53}	t_{54}	t_{55}	t_{56}	t_{57}	t_{58}	t_{59}	t_{60}
u_3 @ClaraListenspre	z_{40}		0.1	0.2	0.1	0.2			0.1		0.2	
	z_{41}				0.1							
	z_{42}		0.1			0.1						
	z_{43}		0.2		0.1							
	z_{44}			0.3	0.2	0.1			0.1		0.1	
z_{45}		0.1	0.1	0.1							0.1	

Figure 3: The points of temporal interest (PoTI) in Figure 2 with: (a) $\{u_1 = \text{@teerasay}, u_2 = \text{@WingsofCrystal}, u_3 = \text{@ClaraListenspre}\} \times \{z_{40}, \dots, z_{45}\} \times \{t_{20}, \dots, t_{30}\}$, (b) $\{\text{@ClaraListenspre}\} \times \{z_{40}, \dots, z_{45}\} \times \{t_{50}, \dots, t_{60}\}$.

the cells have a value in the range $[0.1, 1.0]$. To illustrate that there may be parallel edges, we show how the same graph would look like if the condition c was set such that it would be satisfied if the difference of values fell in the range $[0, 0.1]$. Figure 4b shows \mathcal{G}^{22} for this condition. As seen, values in the RoL $\{u_1\} \times \{z_{43}, z_{44}\}$ and $\{u_2\} \times \{z_{43}, z_{44}\}$ satisfy the latter condition of homogeneity separately.

To find the final 2-d RoLs for time t , we apply depth-first-search (DFS) on the multigraph \mathcal{G}^t based on the pseudo code described in Algorithm 1. We start with a 2-d RoL $r = \mathbb{U} \times \emptyset$; all users \mathbb{U} , but no topics since no node (topic) has been processed yet and $C = [z_1, z_1, z_2, z_2, \dots, z_{|Z|}, z_{|Z|}]$ as the set of all initial nodes (topics) to be processed. Here, C includes duplicated initial topics to support for directed loops on each node. At each intermediate recursive call, we have a current candidate 2-d RoL $r = A \times B$ and a list of not yet processed topics C . We add r into an initially empty set \mathcal{R}^t if it satisfies c and is not already contained in some RoL $r' \in \mathcal{R}^t$. Then, we remove any 2-d RoL $r'' \in \mathcal{R}^t$, which has already been subsumed by r (lines 2-6). We expand the current candidate r from each of its old topics z_i to a new topic z_j if there is a directed edge $(z_i \rightarrow z_j) \in \mathcal{U}^t$. Then, the function is called on the new candidate $\{r.A \cap \mathbb{U}_{z_i, z_j}^t\} \times \{r.B \cup \{z_j\}\}$ (lines 7-15).

For example, let us consider how the 2-d RoLs are identified from the multigraph \mathcal{G}^{22} shown in Figure 4a. Initially the algorithm starts with the candidate 2-d RoL $r = \{u_1, u_2, u_3\} \times \emptyset, C = [z_{40}, z_{40}, z_{41}, z_{41}, \dots, z_{45}, z_{45}]$. We pop node z_{40} and recursively call

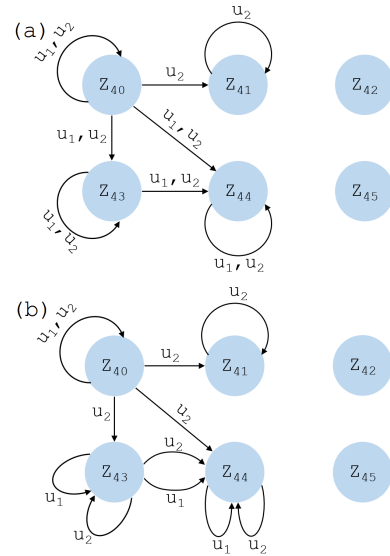


Figure 4: An example multigraph (\mathcal{G}^{22}) constructed from Figure 3a for time t_{22} if the condition c is satisfied when: (a) a value falls within the range $[0.1, 1.0]$, (b) the difference of values falls in the range $[0, 0.1]$.

Algorithm 1 Finding 2-d RoLs for time interval t

Inputs:
users \mathbb{U} , topics \mathbb{Z} , homogeneity condition c , multigraph \mathcal{G}^t

Initialization:
 $\mathcal{R}^t = \emptyset$
find_2d_RoLs($r = \mathbb{U} \times \emptyset, C = [z_1, z_1, z_2, z_2, \dots, z_{|Z|}, z_{|Z|}]$)

Output: 2-d RoLs in \mathcal{R}^t

- 1: **procedure** find_2d_RoLs($r = A \times B, C$)
- 2: **if** ($r \models c$) \wedge ($\nexists r' \in \mathcal{R}^t : r \subset r'$) **then**
- 3: **for all** $r'' \in \mathcal{R}^t$ **do**
- 4: **if** $r'' \subset r$ **then**
- 5: $\mathcal{R}^t \leftarrow \mathcal{R}^t \setminus r''$
- 6: $\mathcal{R}^t \leftarrow \mathcal{R}^t \cup r$
- 7: **for all** $z_j \in \mathbb{Z}$ **do**
- 8: $A \leftarrow r.A; B \leftarrow r.B \cup z_j; C \leftarrow C \setminus z_j$
- 9: **if** $r.B = \emptyset$ **then**
- 10: find_2d_RoLs($A \times B, C$)
- 11: **else**
- 12: **for all** $z_i \in r.B$ **do**
- 13: **for all** $(z_i \rightarrow z_j) \in \mathcal{G}^t.E$ **do**
- 14: $A \leftarrow r.A \cap \mathbb{U}_{z_i, z_j}^t$
- 15: find_2d_RoLs($A \times B, C$)

the function on $r = \{u_1, u_2, u_3\} \times \{z_{40}\}, C = [z_{40}, z_{41}, z_{41}, \dots, z_{45}, z_{45}]$ (line 10). Since $\{u_1, u_2, u_3\} \times \{z_{40}\}$ does not satisfy condition c , we continue by popping a new node (topic) which is again z_{40} . There is only one directed edge (loop) from $z_{40} \rightarrow z_{40}$, so we obtain a new candidate (line 14) and call the function on $r = \{u_1, u_2\} \times \{z_{40}\}, C = [z_{41}, z_{41}, \dots, z_{45}, z_{45}]$ (line 15). Now, the input r satisfies c and we add it to the thus far empty \mathcal{R}^{22} (line 6). Next, we pop z_{41} and there

is a directed edge from $z_{40} \rightarrow z_{41}$ with $\mathbb{U}_{z_{40}, z_{41}}^{22} = \{u_2\}$. So we call the function on $r = \{u_2\} \times \{z_{40}, z_{41}\}$, $C = [z_{41}, \dots, z_{45}, z_{45}]$ which leads to a new element in \mathcal{R}^{22} .

Finding regions of like-mindedness (RoL). Once the sets \mathcal{R}^t of 2-d RoLs for each time interval t have been extracted, we use them to mine the desired RoLs. We employ a similar process as in 2-d RoLs to find RoLs. We build a multigraph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ whose nodes are the time intervals: $\mathbb{V} = \{t_1, t_2, \dots, t_L\}$ and the edges include directed links between time intervals such as $(t_i \rightarrow t_j)$ only if there exists two 2-d RoLs $r \in \mathcal{R}^{t_i}$ and $r' \in \mathcal{R}^{t_j}$ such that $\{r.A \cap r'.A\} \times \{r.B \cap r'.B\} \times \{t_i, t_j\}$ satisfies the condition of homogeneity c . We start with the RoL $R = A \times B \times C$ initialized as $\mathbb{U} \times \mathbb{Z} \times \emptyset$ and a list of not yet processed time intervals $D = [t_1, t_1, t_2, t_2, \dots, t_L, t_L]$. Again each time interval is duplicated to support loop edge $t_i \rightarrow t_i$. We expand the intermediate candidate R from each of its old time intervals t_i to a new time interval t_j if there is a directed edge $(t_i \rightarrow t_j) \in \mathcal{G}.E$. Then, we recursively continue with the new candidate $R = A \times B \times C$ as $A \leftarrow R.A \cap \{r.A \cap r'.A\}$, $B \leftarrow R.B \cap \{r.B \cap r'.B\}$, and $C \leftarrow R.C \cup t_j$ where $t_i \in R.C$ and $r \in \mathcal{R}^{t_i}$ and $r' \in \mathcal{R}^{t_j}$ are 2-d RoLs for time intervals t_i and t_j , respectively.

Finally, we end up with the set \mathcal{R} of all RoLs whose elements $R = A \times B \times C$ show that users $u \in A \subseteq \mathbb{U}$ have a similar degree of interest towards the topics $z \in B \subseteq \mathbb{Z}$ within the time intervals $1 \leq t \leq L \in C$, which serve as the required context of our model. Users who are placed in similar RoLs are considered to be similar; therefore, the more two users are seen in the same RoLs with each other, the closer they should be to each other in the embedding space. On the basis of this context model, the next step is to learn vector representation for each user.

Time complexity analysis. In each time interval t , it takes $O(|\mathbb{U}| \times |\mathbb{Z}|^2)$ to calculate $\mathbb{U}_{z_i, z_j}^t(c)$ for all pairs of $z_i, z_j \in \mathbb{Z}$ and build the multigraph \mathcal{G}^t considering the fact that testing the condition of homogeneity can be done in $O(1)$. Furthermore, performing depth-first-search (DFS) on the graph to find 2-d RoLs takes $O(|\mathbb{U}|^{|\mathbb{Z}|})$ in the worst case, which happens when there exists an edge between each pair of z_i and z_j associated with $\mathbb{U}_{z_i, z_j}^t(c)$ containing only one user. The analysis of the time complexity for finding RoLs is similar but in the context of the number of time intervals and the number of identified 2-d RoLs in the former step. Here, for each pair of time intervals t_i and t_j , and a pair of 2-d RoLs, we test the condition of homogeneity which would take $O(|r| \times L^2)$ plus a final DFS in $O(|r|^L)$ where $|r|$ is the number of all 2-d RoLs. As seen, the most expensive parts are the DFS traversal on the multigraphs in the first and second steps which highly depend on the condition for homogeneity c . Nevertheless, our algorithm is empirically efficient since:

- (1) PoTI is inherently sparse. Users are interested in small sets of topics of interest. As a result, the average number of edges in the multigraphs drops significantly and the DFS time approaches linear complexity due to the sparsity of the graph.
- (2) The depth of the search in DFS is likely to be small in practice because the number of topics and time intervals are far fewer than the number of users, i.e., $|\mathbb{Z}| \ll L \ll |\mathbb{U}|$.
- (3) The proposed algorithm is inherently parallelizable over time intervals.

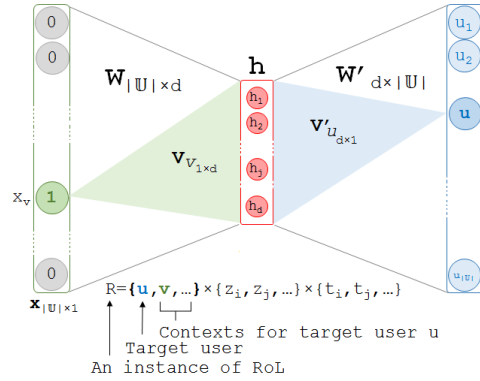


Figure 5: The neural network architecture.

4.2.2 User Embedding. Given a set of discovered regions of like-mindedness (RoLs), the context for a user u would be the set of users in each of the RoLs where u had been observed. We formulate user vector learning as a maximum likelihood (ML) optimization problem. In particular, for each user, we find her like-minded users by optimizing the conditional probability of observing users that have the same RoLs as her. To induce the user embeddings, we adopt an approach similar to [18] as follows:

Definition 4.3. (Embedding Objective) Given the set \mathcal{R} of all regions of like-mindedness (RoLs), the embedding function $g: \mathbb{U} \rightarrow [0, 1]^d$ maps each user $u \in \mathbb{U}$ onto a d -dimensional space, such that the following objective is optimized:

$$\arg \max_g \sum_{R \in \mathcal{R}, u \in R} \log \Pr(u|R \setminus u) \quad (1)$$

In order to make the optimization tractable, we assume conditional independence for observing users in a RoL such as R . So,

$$\Pr(u|R \setminus u) = \prod_{v \in R \setminus u} \Pr(u|v) \quad (2)$$

To learn the user embeddings, we use a single hidden layer, fully connected neural network. The architecture of our neural network is shown in Figure 5. The hidden layer \mathbf{h} is of size d , the dimensionality of the resulting user vectors, and the input and output layer is set to have as many neurons as $|\mathbb{U}|$. Thus, the input to hidden layer connections can be represented by matrix \mathbf{W} of size $|\mathbb{U}| \times d$ with each row representing a vector for user $u \in \mathbb{U}$. The input layer \mathbf{x} is a one-hot encoded vector and the hidden layer's neurons are all linear such that $\mathbf{h} = \mathbf{W}^T \mathbf{x}$. Given a user v in the input layer that is taken from the context of u , i.e., u and v have been observed in the same RoL, \mathbf{h} is the transpose of v 's corresponding row in \mathbf{W} named \mathbf{v}_v . In the same way, the connections from hidden layer to output layer can be described by matrix \mathbf{W}' of size $d \times |\mathbb{U}|$. The prediction task could be done via a softmax function to approximate the probability of observing the target user u given user v from the same RoL, i.e.,

$$\Pr(u|v) = \frac{\exp(\mathbf{v}_u^T \mathbf{h})}{\sum_{w \in \mathbb{U}} \exp(\mathbf{v}_w^T \mathbf{h})} = \frac{\exp(\mathbf{v}_u^T \mathbf{v}_v)}{\sum_{w \in \mathbb{U}} \exp(\mathbf{v}_w^T \mathbf{v}_v)} \quad (3)$$

where v'_u is u 's corresponding column of matrix \mathbf{W}' . With the assumption in Equation 2 and the above probability function, the objective function in Equation 1 simplifies to:

$$\arg \max_g \sum_{R \in \mathcal{R}, u \in R} \left[\sum_{v \in R \setminus u} [(v'_u)^T v_v] - \log \sum_{w \in \mathcal{U}} \exp(v'_w)^T v_v \right] \quad (4)$$

However, the formulation is computationally intractable as its time complexity is proportional to the size of \mathcal{U} . Morin and Bengio [20] have proposed hierarchical softmax to approximate the full softmax efficiently in practice. Accordingly, instead of a matrix, the hidden layer to output layer connection is a binary Huffman tree where leaves correspond to users. The probability of a user is estimated by the unique path from the root to her corresponding leaf. Therefore, the complexity of calculating softmax probabilities drops from $O(|\mathcal{U}|)$ to $O(\log(|\mathcal{U}|))$. We refer the reader to [18] for further details.

Our neural network is trained using stochastic gradient descent and updates \mathbf{W} and \mathbf{W}' gradually via backpropagation. After the training converges, each row of \mathbf{W} represents the d -dimensional user embeddings.

4.3 User Community Detection

Given the user embeddings, we identify communities of users through graph-based partitioning heuristics. We represent users and their pairwise similarities through a weighted undirected graph. Precisely, let $G = (\mathcal{V}, \mathbb{E}, s)$ be a weighted user graph in time period T such that $\mathcal{V} = \mathcal{U}, \mathbb{E} = \{e_{u,v} : \forall u, v \in \mathcal{U}\}$ and the weight function $s : \mathbb{E} \rightarrow [0, 1]$ is the cosine similarity of embeddings for the incident users of an edge defined as $s(e_{u,v}) = \frac{\mathbf{v}_u \cdot \mathbf{v}_v}{\|\mathbf{v}_u\|_2 \|\mathbf{v}_v\|_2}$. After constructing the user graph G for a given time period T , it is possible to employ a graph partitioning heuristic to extract clusters of users that form latent communities. We leverage the Louvain method (LM) [6] as it introduces linear heuristics to the problem of graph partitioning. The output is a set of induced subgraphs such as $G[C]$ whose vertex set $C \subset \mathcal{V}$ and edge set consists of all of the edges in \mathbb{E} that have both endpoints in C . Subgraph $G[C]$ with $|C| \geq 2$ form an instance of *temporal like-minded user community* assuming $C \in \mathbb{P}^*$. The application of graph partitioning algorithms on G will produce temporal user communities \mathbb{P}^* that consist of like-minded users who have contributed to the same topics with similar temporal behavior and contribution degrees.

5 EXPERIMENTAL SETUP AND EVALUATION

5.1 Dataset

In our experiments, we use a publicly available Twitter dataset collected and published¹ by Abel et al. [1]. It consists of approximately 3M tweets posted by 135,731 unique users between November 1 and December 31, 2010. In addition to its text, each tweet includes user id and timestamp. The whole two months time period is sampled on a daily basis, i.e., $L = 61$ days.

5.2 Setup

Our proposed approach consists of three phases to identify temporally like-minded user communities; finding topics, building user

¹www.wis.ewi.tudelft.nl/umap2011/

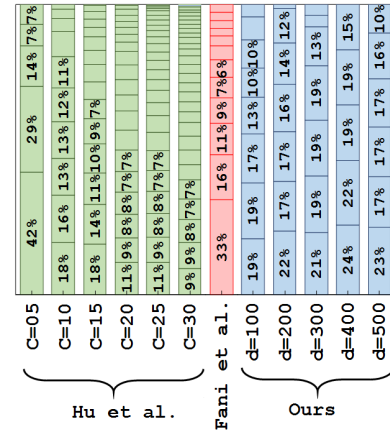


Figure 6: User distribution for the output communities of our approach and the baselines.

vector representations, and detecting user communities. Here, we provide the implementation details and the setup of our approach in each of these phases.

5.2.1 Finding topics. Extracting topics from tweets suffers from the sparsity problem when topic modeling methods such as LDA are used [32]. As suggested in [33, 34], we annotate each tweet with entities defined in Wikipedia to obtain better topics from Twitter with no change in the underlying topic detection methods. For instance, for a tweet such as ‘NATO Leaders Seek Time on Afghan Exit Strategy - <http://nyti.ms/cMMDuR>’, a semantic annotator such as TagMe [10] is able to identify and extract several Wikipedia entities, namely ‘NATO’², ‘Afghan’, and ‘Exit_Strategy’. Using entities instead of words can lead to the reduction of noisy content within the topic detection process, because each concept implicitly represents a collection of typical terms which are collectively more meaningful than a single word or a group of less coherent words [24]. We annotated the text of each tweet with Wikipedia entities using the TAGME RESTful API³, which resulted in 350,731 unique entities.

In order to find topics of interest in our dataset, we have applied MALLET⁴ for LDA. LDA-based approaches to topic detection need *a priori* knowledge for the number of topics. The number of topics has been already investigated and set to 50 for the same tweet dataset by other researchers in [9]. We populate the points of temporal interest (PoTI) for our topic set \mathcal{Z} on a daily basis, i.e., $L = 61$ days, and screen out values less than 0.1. The condition for homogeneity c is set such that the difference of values falls in the range $[0, 0.1)$.

5.2.2 Building user vector representation. We extended CBOW architecture in Gensim⁵ to learn user embeddings as already introduced in this paper. The training phase uses a learning rate of 0.025 and in each epoch we decrease it by 0.002 for 200 epochs. We perform the experiments on different vector sizes of $d = 100, 200, \dots, 500$ in an increasing order till we see no further performance gain.

²en.wikipedia.org/wiki/NATO

³services.d4science.org/web/tagme/documentation

⁴mallet.cs.umass.edu/topics.php

⁵radimrehurek.com/gensim/models/word2vec.html

5.2.3 Detecting user communities. We build temporal topic-based communities according to our proposed approach in Section 4.3. We build the weighted graph G and apply the Louvain method with resolution parameter 0.1 using Pajek⁶. This leads to our temporal topic-based communities \mathbb{P}^* .

5.3 Baselines

We compare our work against the following baselines whose details has been already given in the related work section:

Fani et al. [9]. This approach models user's contributions toward the topics of interest through a multivariate time series. We use LDA in its topic detection step with 50 topics and build the time series for daily time intervals $L=61$ days in its user modeling step. The approach uses two dimensional cross correlation to measure the similarity of a pair of users' time series. We use the implementation in MATLAB⁷ for calculating time series cross-correlation. Finally, we use the Louvain method in Pajek for its community detection step as proposed by the authors.

Hu et al. [13]. This is a parametric unified probabilistic generative model for topics and communities. The number of topics is set to 50 and we perform experiments on increasing number of communities for $C = 5, 10, \dots, 30$ till we see no performance gain. The number of iterations is set to 1,000. This method is a mixture model in which all users are members of all communities with a probability distribution. In our comparison, we only consider the community with the highest probability as each user's community.

Figure 6 provides an overview of the distribution of users across different communities. For Hu et al.'s work, the number of communities needs to be specified as shown ranging from 5 to 30. For our proposed approach, the number of communities is automatically determined by the graph partitioning method; however, the size of the embeddings needs to be provided, which has been set from 100 to 500. As seen in the figure, our proposed method leads to a more fair distribution of users across communities while the two baseline methods have a higher skewness in the distribution of users in their identified communities. While this by itself is not a measure of community quality, as we will show later, disproportionate distribution of users in communities could lead to poor application level performance.

5.4 Evaluation Protocol and Gold Standard

On the one hand, contrary to typically small scale networks or synthetic ones, gold standard communities for real social networks are not available. So, well-defined quality measures such as Rand index, Jaccard index, or normalized mutual information (NMI) that require comparison to a gold standard are not applicable. On the other hand and in the absence of a golden standard, quality functions such as *modularity* are not helpful either since they are based on the explicit links between users, which are not applicable to our work. For instance, in the context of our work and the baselines, a perfect community detection algorithm might have a low modularity value as those users that are deemed most similar might not have explicit social connection with each other. Therefore in

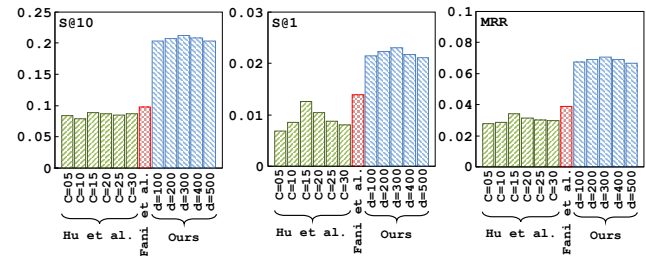


Figure 7: Comparative performance on the news recommendation application.

our work, the communities that achieve high modularity are not necessarily optimal from temporal and topical points of view [19].

Fortunately, the performance of community detection methods can be measured through observations made at the application level, as suggested in [14, 19]. In these evaluation strategies, a temporal like-minded user community detection method is considered better iff its output communities improve an underlying application. We deploy three applications: news recommendation, user prediction, and community selection. Note should be taken that we do not attempt to improve the state of the art in any of these three applications but rather to show that the application of the proposed community detection method is able to provide a stronger performance compared to the other two state of the art community detection baselines.

To this end, we first build a gold standard dataset for the said applications by collecting news articles to which a user has explicitly linked in her tweets (or retweets). We postulate that users post news article urls since they are interested in the topics of those news articles. Similar to tweets, we annotate news articles with Wikipedia entities. We build the gold standard from a set of news articles whose urls have been posted by user u at time t . We see each entry as a triple (u, a, t) consisting of the news article a , user u , and the time t . As a result, $\mathbb{G} = \{(u, a, t) : u \in \mathbb{U}, a \in \mathbb{A}, 1 \leq t \leq L = 61\}$ forms our gold standard where \mathbb{U} and \mathbb{A} are sets of users and news articles. The gold standard \mathbb{G} consists of 25,756 triples extracted from 3,468 distinct news articles posted by 1,922 users.

5.5 News Recommendation

Given the gold standard \mathbb{G} and like-minded user communities \mathbb{P}^* , the objective in the news recommender application is to recommend the right news articles to the users of communities in the correct time. A right news article a to be recommended to a user u at time t would be one that is included in the gold standard, that is, $(u, a, t) \in \mathbb{G}$. The news recommender application works according to the following two steps:

- (1) For each $\mathbb{C} \in \mathbb{P}^*$ and time interval $1 \leq t \leq L = 61$, we recommend all news articles $a \in \mathbb{A}$ in a ranked list based on the similarity of the news article a and the community's overall topics of interest at the time t . The overall topic of interest for each topic $z \in \mathbb{Z}$ in a community is the sum over all entries in PoTI whose users belong to the community at time t , i.e., $\sum_{u \in \mathbb{C}} y_u^t[z]$.

⁶vlado.fmf.uni-lj.si/pub/networks/pajek/

⁷www.mathworks.com/help/signal/ref/xcorr2.html

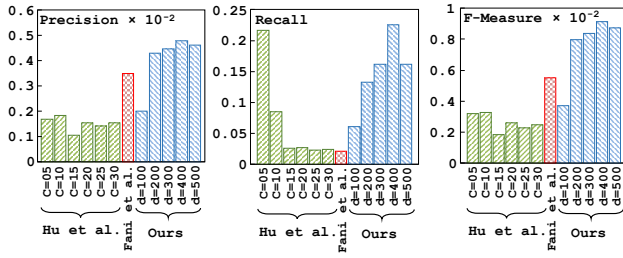


Figure 8: Comparative performance on the user prediction.

- (2) We recommend the news articles $a \in \mathbb{A}$ to a user $u \in \mathbb{U}$ based on the same ranked list as her community’s list.

In our work, a desirable community is one whose members are interested in the same topics of interest in the same time intervals. As a result, ideally, at time t , a news article is about the same topics of interest as the community’s overall interest iff all the members post about the same or highly similar news articles.

We evaluate the ranked list of news articles for recommendation by standard information retrieval metrics: mean reciprocal rank (MRR) and success at rank k ($S@k$). The former is the inverse of the first position that a correct item occurs within the ranked list and the latter shows the probability that at least one correct item occurs within the top- k items of the ranked list. In the following, the three methods, namely our approach, Fani et al. [9], and Hu et al. [13], are compared to each other in terms of MRR, $S@1$ and $S@10$. Figure 7 summarizes the results.

As shown, our approach with different dimensions achieves better performance compared with the approach proposed by Hu et al. [13] and Fani et al. [9] in terms of $S@1$, $S@10$, and MRR. Also, the results show that with $d = 300$ our approach reaches its best performance. We attribute our better performance to the fact that our embedding function preserves both topical and temporal proximity of users more effectively and, consequently, the extracted like-minded user communities capture temporal topic-based similarity of users more coherently than the other two baselines.

5.6 User Prediction

Another application with which we evaluate our approach and the baselines is the user prediction application. Given the gold standard \mathbb{G} and the like-minded user communities \mathbb{D}^* , this time the goal is to predict which users posted the news article a at time t . To do so, we find the closest community to the news article in terms of topics of interest at time t . Then, the members of such community would constitute our prediction list. We employ precision, recall, and f-measure to report user prediction performance. We summarize the results for these metrics in Figure 8. As shown, in terms of precision, our approach with all different dimensions except $d = 100$ outperforms all other baselines. In terms of recall, however, Hu et al. [13] with $C = 5$ competes with our proposed approach when $d = 400$. The reason for such high recall in Hu et al. with $C=5$ can be attributed to the lower number of communities in this method. The fewer the number of communities are, the higher the recall of the method would be. In other words, if we only have one community that includes all users, recall would be 1.0. As the

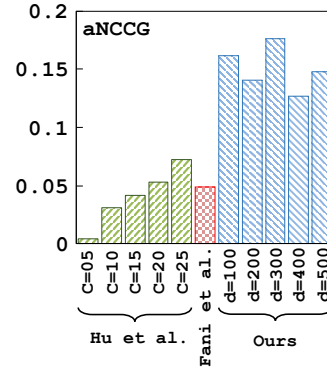


Figure 9: Performance on community selection application.

number of communities increases from $C = 10$ to $C = 30$ in Hu et al, recall decreases which supports our explanation. Overall, F-measure shows the superiority of approach in user prediction in all its variants except for $d = 100$, which is weaker than Fani et al.

5.7 Community Selection

In the realm of cluster-based information retrieval systems, the entire collection of documents are split into clusters such that only the documents in highly related clusters to a given query are accessed. As a result, fewer documents are searched from within a large collection of documents which results in improved response time. Better clustering solutions in this context are those that can group relevant documents for previously unseen queries. This approach is referred to as *collection selection* and normalised cumulative cluster gain (NCCG) [21] is a metric used for evaluating collection selection. According to NCCG, the best clustering would be the one where all the documents related to a given query are all located in the same cluster. The worst clustering is the one where the relevant documents to an input query are scattered across many clusters. NCCG is the difference between the current clustering gain and the worst possible, formulated as follows:

$$NCCG = \frac{s - s_{min}}{1 - s_{min}} \tag{5}$$

where $s = \frac{\sum^{|g|} cumsum(g)}{n^2}$ and g is a sorted gain vector whose elements represent each cluster’s gain, i.e., the number of relevant documents in a cluster, n is the total number of relevant documents and $cumsum$ represents the cumulative sum of a vector. The worst possible gain s_{min} happens when the relevant documents to the query are uniformly distributed over all clusters.

However, NCCG has been criticized for being sensitive to the number of clusters and population distribution; therefore, De Vries et al. [28] have proposed an *adjusted* version of NCCG (aNCCG), i.e., NCCG’s divergence from a random null base model, to alleviate such problem. We employ aNCCG to evaluate the temporal and topical coherence of the identified output communities of the different approaches in the application of community selection as follows: given a news article a at time t (the input query), we want to find the communities of those users (similar to documents related to an input query) who have mentioned the news article at that time. The

output user communities are more effective iff users who mention a news article a (topical) at time t (temporal) are all located in one community instead of being distributed across several communities. We report aNCCG for our approach and the baselines in Figure 9.

As seen in the figure, our approach, for different number of dimensions, outperforms the other two baselines in terms of aNCCG. This means that, in our approach, the users who mention the same news articles in specific time intervals are placed within similar user communities, i.e., such users are distributed across fewer communities. A lower aNCCG value as exhibited by Hu et al. and Fani et al. means that these methods distribute users that have posted similar news articles at specific time intervals across a larger number of user communities, which is not desirable.

6 CONCLUDING REMARKS

In this work, we have proposed a neural embedding approach to identify temporally like-minded user communities, i.e., those communities of users who have similar temporal alignment in their topics of interest. We model the users' temporal contribution towards topics of interest by introducing the notion of regions of like-mindedness (RoLs) between users. These regions cover users who share not only similar topical interests but also similar temporal behavior. By considering the identified set of RoLs as *context*, we train a neural network such that the probability of a user in a RoL is maximized given other users in the same RoL. The final weights of the neural networks form our low-dimensional vector representation of each user that incorporates both users' topics of interest and their temporal nature. Finally, we apply a clustering technique to identify like-minded user communities on a weighted user graph in which the similarity of two users is based on the cosine similarity of their respective vectors. We demonstrate the effectiveness of our approach on a Twitter dataset in the context of news recommendation, user prediction, and community selection applications compared to two state of the art baselines. Possible future direction of our work includes the interpolation of our learnt user embeddings with neural embeddings learnt from a graph structure as in node2vec [12] or DeepWalk [23] such that both temporal topics as well as structural connections are considered when building user embeddings.

REFERENCES

- [1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. [n. d.]. Analyzing User Modeling on Twitter for Personalized News Recommendations. In *UMAP 2011*.
- [2] Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. [n. d.]. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In *CoNLL 2016*.
- [3] Ehsaneddin Asgari and Mohammad R. K. Mofrad. 2015. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE* 10, 11 (11 2015), 1–15.
- [4] Adrian Benton, Raman Arora, and Mark Dredze. [n. d.]. Learning Multiview Embeddings of Twitter Users. In *ACL 2016*.
- [5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.
- [7] Li Chen, Tiejun Qian, Peisong Zhu, and Zhenni You. [n. d.]. Learning User Embedding Representation for Gender Prediction. In *ICTAI 2016*.
- [8] Ying Ding. 2011. Community detection: Topological vs. topical. *J. Informetrics* 5, 4 (2011), 498–514. <https://doi.org/10.1016/j.joi.2011.02.006>
- [9] Hossein Fani, Ebrahim Bagheri, Fattane Zarrinkalam, Xin Zhao, and Weichang Du. 2017. Finding Diachronic Like-Minded Users. *Computational Intelligence: An International Journal* (2017).
- [10] Paolo Ferragina and Ugo Scaiella. [n. d.]. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM 2010*.
- [11] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3/5 (2010), 75 – 174.
- [12] Aditya Grover and Jure Leskovec. [n. d.]. node2vec: Scalable Feature Learning for Networks. In *KDD 2016*.
- [13] Zhiting Hu, Junjie Yao, and Bin Cui. [n. d.]. User Group Oriented Temporal Dynamics Exploration. In *The 28th AAAI Conference on Artificial Intelligence, 2014*.
- [14] Zhiting Hu, Junjie Yao, Bin Cui, and Eric P. Xing. 2015. Community Level Diffusion Extraction. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*. 1555–1569.
- [15] Ganesh J, Soumyajit Ganguly, Manish Gupta, Vasudeva Varma, and Vikram Pudi. [n. d.]. Author2Vec: Learning Author Representations by Combining Content and Link Information. In *WWW 2016*.
- [16] Deepika Lalwani, Durvasula V. L. N. Somayajulu, and P. Radha Krishna. [n. d.]. A community driven social recommendation system. In *International Conference on Big Data 2015*.
- [17] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27 (2001), pp. 415–444.
- [18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. [n. d.]. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS 2013*.
- [19] Farnaz Moradi, Tomas Olovsson, and Philippos Tsigas. 2012. An Evaluation of Community Detection Algorithms on Large-Scale Email Traffic. In *SEA 2012*.
- [20] Frederic Morin and Yoshua Bengio. [n. d.]. Hierarchical Probabilistic Neural Network Language Model. In *AISTATS 2005*.
- [21] Richi Nayak, Christopher M. De Vries, Sangeetha Kutty, Shlomo Geva, Ludovic Denoyer, and Patrick Gallinari. [n. d.]. Overview of the INEX 2009 XML Mining Track: Clustering and Classification of XML Documents. In *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009*.
- [22] Dunlu Peng, Xie Lei, and Ting Huang. 2015. DICH: A framework for discovering implicit communities hidden in tweets. *World Wide Web* 18, 4 (2015), 795–818.
- [23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. [n. d.]. DeepWalk: online learning of social representations. In *KDD 2014*.
- [24] Georgios Petkos, Symeon Papadopoulos, Luca Maria Aiello, Ryan Skraba, and Yiannis Kompatsiaris. [n. d.]. A soft frequent pattern mining approach for textual topic detection. In *WIMS 2014*.
- [25] Dimitrios Rafailidis, Pavlos Kefalas, and Yannis Manolopoulos. 2017. Preference dynamics with multimodal user-item interactions in social media recommendation. *Expert Syst. Appl.* 74 (2017), 11–18. <https://doi.org/10.1016/j.eswa.2017.01.005>
- [26] Vignesh Ramanathan, Kevin D. Tang, Greg Mori, and Fei-Fei Li. [n. d.]. Learning Temporal Embeddings for Complex Video Analysis. In *ICCV 2015*.
- [27] Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruque, and L. Venkata Subramaniam. [n. d.]. Using content and interactions for discovering communities in social networks. In *The 21st World Wide Web Conference 2012, WWW 2012*.
- [28] Christopher M. De Vries, Shlomo Geva, and Andrew Trotman. 2012. Document Clustering Evaluation: Divergence from a Random Baseline. *CoRR abs/1208.5654* (2012).
- [29] Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Xiaofang Zhou. 2015. Dynamic User Modeling in Social Media Systems. *ACM Trans. Inf. Syst.* 33, 3 (2015), 10:1–10:44.
- [30] Zhijun Yin, Liangliang Cao, Quanquan Gu, and Jiawei Han. 2012. Latent Community Topic Analysis: Integration of Community Discovery with Topic Modeling. *ACM TIST* 3, 4 (2012), 63:1–63:21. <https://doi.org/10.1145/2337542.2337548>
- [31] Yang Yu, Xiaojun Wan, and Xinjie Zhou. [n. d.]. User Embedding for Scholarly Microblog Recommendation. In *ACL 2016*.
- [32] Fattane Zarrinkalam and Ebrahim Bagheri. 2017. Event identification in social networks. In *Encyclopedia with Semantic Computing and Robotic Intelligence*. Vol. 01. 1630002 [7].
- [33] Fattane Zarrinkalam, Hossein Fani, Ebrahim Bagheri, and Mohsen Kahani. [n. d.]. Inferring Implicit Topical Interests on Twitter. In *ECIR 2016*.
- [34] Fattane Zarrinkalam, Hossein Fani, Ebrahim Bagheri, and Mohsen Kahani. [n. d.]. Predicting Users' Future Interests on Twitter. In *ECIR 2017*.
- [35] Qian Zhang, Dianshuang Wu, Guangquan Zhang, and Jie Lu. [n. d.]. Fuzzy user-interest drift detection based recommender systems. In *FUZZ-IEEE 2016*.
- [36] Lizhuang Zhao and Mohammed Javeed Zaki. [n. d.]. TriCluster: An Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data. In *SIGMOD 2005*.
- [37] Shenghuo Zhu, Kai Yu, Yun Chi, and Yihong Gong. 2007. Combining content and link for classification using matrix factorization. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*. 487–494.