Mining Homophilic Groups of Users using Edge Attributed Node Embedding from Enterprise Social Networks

Priyanka Sinha
Tata Consultancy Services Limited
IIT Kharagpur, India
priyanka.sinha.iitg@gmail.com

Ritu Patel
IIT Kharagpur
India
msvr4ritu123@gmail.com

Pabitra Mitra
IIT Kharagpur
India
pabitra@gmail.com

Dilys Thomas
Tata Consultancy Services Limited
India
dilys@cs.stanford.edu

Lipika Dey
Tata Consultancy Services Limited
India
lipikadey@gmail.com

ABSTRACT

We develop a method to identify groups of similarly behaving users with similar work contexts from their activity on enterprise social media. This would allow organizations to discover redundancies and increase efficiency. To better capture the network structure and communication characteristics, we model user communications with directed attributed edges in a graph. Communication parameters including engagement frequency, emotion words, and post lengths act as edge weights of the multiedge. Upon the resultant adjacency tensor, we develop a node embedding algorithm using higher order singular value tensor decomposition and convolutional autoencoder. We develop a peer group identification algorithm using the cluster labels obtained from the node embedding and show its results on Enron emails and StackExchange Workplace community. We observe that people of the same roles in enterprise social media are clustered together by our method. We provide a comparison with existing node embedding algorithms as a reference indicating that attributed social networks and our formulations are an efficient and scalable way to identify peer groups in an enterprise social network that aids in professional social matching.

CCS CONCEPTS

• **Computing methodologies** → *Factorization methods*; *Unsupervised learning*; • **Information systems** → *Social networks*; • **Applied computing** → *Law, social and behavioral sciences.*

KEYWORDS

clustering, tensor node embedding, enterprise social media

ACM Reference Format:
Priyanka Sinha, Ritu Patel, Pabitra Mitra, Dilys Thomas, and Lipika Dey. 2022. Mining Homophilic Groups of Users using Edge Attributed Node Embedding from Enterprise Social Networks. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3487553.3524726

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524726

1 INTRODUCTION

Enterprises consist of groups of people working towards a goal that aligns with its overall leadership directed plans. In commercial enterprises, there is often a hierarchy and people are assigned to groups. The groups may compose of persons expected to perform specific roles. In this paper, we develop a method to identify peer groups within an enterprise. This would be valuable in workplace analytics for human resource recommendation purposes thereby increasing engagement within an enterprise.

Any large and geographically distributed organization has people in different organizationally identified work contexts to be similar in behavior and, expertise. Oftentimes they are unaware of each other's existence. These people are potentially peers. Identifying such sets of employees is important for an organization to help reduce redundancies and increase productivity and efficiency. This is because when people with similar actual work contexts and behavior connect via such an automated recommendation they have a higher likelihood of exchanging information and collaborating towards better solutions faster. To facilitate such peer groups to emerge, recommendations need to be based on behavioral attributes without which it is unlikely they would make conversation.

We develop a method to identify groups of homophilic users with similar work contexts from their activity on enterprise social media. To better capture the network structure and communication characteristics, we model users with directed attributed edges in a graph with each attribute as an edge weight of a multiedge graph. Upon the resultant adjacency tensor, we develop a node embedding algorithm using higher order singular value tensor decomposition and convolutional autoencoders. We demonstrate that the node embedding better captures the network structure and communication characteristics as people of the same roles in enterprise social media cluster together.

We next propose a homophilic group identification algorithm obtaining the cluster labels using node embedding. It allows us to identify homomorphic subgraphs efficiently within a graph and thus can be applied to mine patterns in large graphs. We empirically present the results on Enterprise data sets such as an Enterprise Social Media, Enron emails, and StackExchange Workplace community. We provide a comparison with existing node embedding

algorithms as a reference indicating that attributed social networks in our formulations are an efficient and scalable way to identify homophilic groups in an enterprise social network that aids in professional social matching [23].

2 RELATED LITERATURE

In recent times, several research groups have shown that it is possible to profile social media users along behavioral attributes based on their social-network behavior [5, 26, 31, 32]. Mining group communications can similarly yield information related to the behavior attributes of members in a group and the role that these attributes play in group dynamics. Group recommendations even in popular social media such as Facebook and Linkedin are an important source of their new links. The authors in [38–40] mine academic teams using motif discovery methods and graphlets.

Non attributed networks node embedding has been approached via matrix factorization approaches as well as a sampling of node neighborhoods using random walks in methods such as node2vec [9] and struc2vec [27] and graph convolutional methods [18].

There has recently been extensive research in attributed networks, both clustering networks with node attributes as well as edge attributes. Authors use guided levy flights to learn node embedding in multigraphs termed Multigraph2Vec in [28] without weighted edges. [8] defines an algorithm to use a variational autoencoder to reduce the complexity of edge attributes in attributed social networks to provide node embedding for various tasks such as clustering. They incorporate social roles information into the embedding algorithm by using them as features. We already embed our features separately hence a simple deep autoencoder is able to reconstruct and compress the interdependencies.

Tensors help represent edge attributed graphs where two dimensions are the nodes and the third dimension are the attributes for the directed edges between the nodes. We have researched tensors previously [1]. [17] is a survey of tensor decomposition. There are several ways of decomposing tensors [16] meaningfully into lower rank matrices. Based on the Tucker decomposition, the higher order singular value decomposition (HOSVD) results in matrices that retain orthogonality amongst them and a lower rank core tensor. This allows a simple application of the heat kernel filter to generate spectral graph wavelets.

The authors in [4] use matrix factorization to identify homophilic users. They use click data to build the social network between users from which they statistically infer the trust relationship which is considered key to homophily between users. We embody such user relationships in our paper using the various complex features such as emotion and psycholinguistic attributes such as EMPATH [7] which would likely carry more information of homogeneity between users.

The authors in [25] explain how existing node embedding techniques such as node2vec [9] can be expressed as matrix factorization methods. This inspires us to our own methods using tensor factorization for attributed edges. The paper [3] takes the approach where they use community detection using Louvain hierarchically to identify smaller subgraphs that they then embed. In the paper [29], the authors identify social homophily using graph neural networks in a similar vein that we identify peer groups. They identify all position and social aware temporally influenced users in a social media network on multiple data sets such as Stackoverflow, Weibo, and Digg.

2.1 Graphwave Node Embedding

The Graphwave algorithm [6] has been used earlier to compute structural subgraph similarity using a node embedding. Here, the adjacency matrix consisting of the weights of the directed edges in the graph are taken into consideration. A spectral graph wavelet (such as heat kernel) is applied on its Laplacian transforming it into one-hot vector representation of the node and its characteristics with respect to its neighbors. Thereafter, a characteristic function is calculated based on the influence of nearby nodes. This gives a characteristic representation of the subgraph surrounding the given node. From this characteristic function, considering it to be a distribution, points are sampled and they constitute the node embedding.

Graphwave takes as input the weighted directed adjacency matrix of the graph. It calculates the spectral graph wavelets of each node considering a heat kernel at the node. Thereafter, it calculates the effect on neighboring nodes' on itself. They then view that as a distribution and using a characteristic function defined over it, they sample evenly spaced points as being representative of the node and its neighborhood that make up the embedding. On the Enron data set, they demonstrate good separation between top executives and junior executives while close proximity amongst top executives using only the structural embedding of them as nodes.

Graphwave is not defined for attributed networks. We extend this algorithm for edge attributed networks and experimentally compare our algorithm to this and other existing node embedding algorithms such as [9], and on the Enron data set to demonstrate the increase in fine-grained quality of identifying structurally important nodes.

2.2 Social Media Attributes Relevant to Human Behavior

We give an overview of the various features relevant to the psychological state of the authors' mind indicating their behavior that are mined from social media. Apart from unstructured text in the form of posts and comments, social media also provides structured attributes such as likes, favorites, shares, mentions, follows, type of post, and other such quantifiable information that are relevant to behavior.

Various measures from enterprise social media that are relevant to behavior analysis are listed in the paper [10]. They summarize from previous literature the aspects of enterprise social media that are relevant to user behavior such as regularity of logging in, frequency of participation, initiation taken in participation, asking questions, answering, dispersion of focus on varied communities and topics, engagement, popularity, quality of content and network structure. Based on these they identify enterprise social media relevant metrics with factor analysis to demonstrate relevance to user behavior such as the number of replies exchanged with other users and diversity spread of these users, network measures such as centrality and betweenness, posting activity, time delay in receiving replies, receiving thanks as replies.

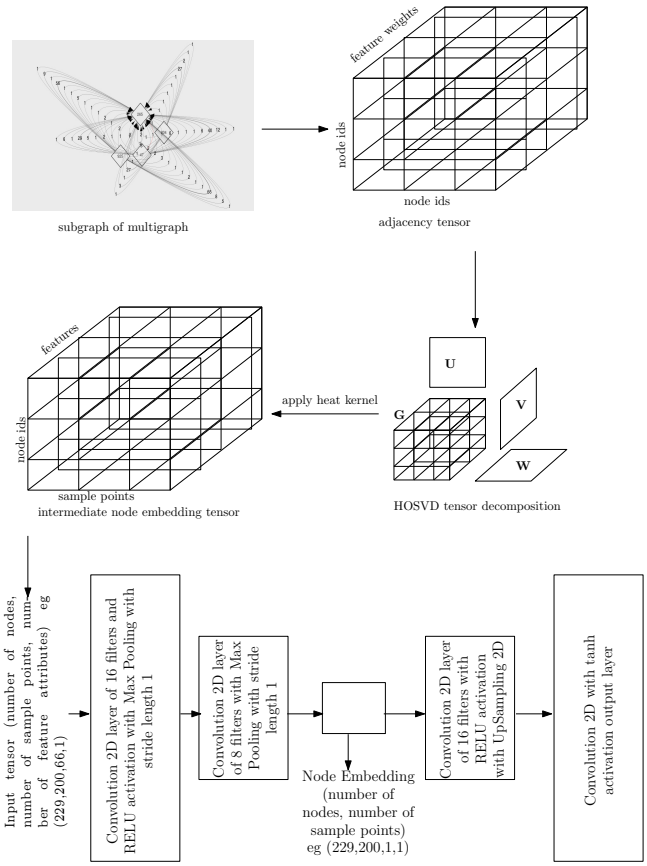

Figure 1: Block diagram of the proposed edge attributed node embedding algorithm

Furthermore, 16 metrics relevant to enterprise behavior from an enterprise social media of an Australian professional services firm using factor analysis are identified in paper [36] along four dimensions i.e. contribution and networking, information provision, contact dispersion, and invisible usage. The metrics are similar to the ones in [10] such as threads created, contributed to, replies received, diversity of users interacted with, thanks received, network degree, and centrality measures.

In [10, 30, 34] researchers give us a list of measures that are relevant for enterprise behavior understanding. Most of them use some counts and ratios of the number of messages exchanged and few linguistic features of the messages. These measures are mostly focused on mailing lists and email exchanges. We extend the same for our enterprise social media data sets. These complex features both from unstructured text and structured attributes have relevance to human behavior analysis.

3 MODELING ENTERPRISE SOCIAL MEDIA AS A MULTIGRAPH

Enterprise social networks can be considered as attributed graphs where not only users have attributes attached to themselves, communication with each other is also attributed based on the various features that can be inferred from the communication. This includes text content, activity statistics, and other derived attributes. It can be modeled using an attributed graph with multiedges and self-loops. Let there be a graph $G = (V, E, T)$ consisting of vertices (V) of the graph as users of the enterprise social network, and edges E as the communications between the nodes of the graph, i.e. the users. Each edge E can consist of a vector of feature values, an entry for each type of feature. We denote such vector attributed edges as *multiedges*. T is the tensor or multilinear array that represents these directed multiedges from one user to another where the tensor values are the edge weights.

An alternate formulation may be as a set of graphs (layers) $G_1, G_2, .., G_n$ for n features G_i is the layer of the social network that has that ith feature as their attribute, E_i are the corresponding directed edges amongst them and A_i is the adjacency matrix containing the values for the ith features as edge weights.

As we can see in Section 5.1 various features are relevant to interactions between people in an enterprise. Identifying a single feature amongst these that characterize employee behavior and group behavior is hard. Each of these features has varied ranges and are either independent of each other or are non-linearly dependent. Hence, since the relationship between two nodes has multiple facets, collapsing them into a single value loses important information, and it becomes imperative that we have attributed edges such as noted in [8] and [14].

By using attributed edges we are able to identify the structural context better. For example, without both topical edges and the sentiment towards a topic, we would only be able to identify relationships and structures based on aggregate sentiment scores which may never reflect the agreement dynamics between the participants.

4 PROPOSED EDGE ATTRIBUTED NODE EMBEDDING AND HOMOPHILY GROUPING

The proposed algorithm consists of three steps. First, we find an embedding for the edge attributed graphs in an enterprise social network using the proposed multiwave tensor embedding algorithm. Next, we cluster the embedding using the k-means algorithm. Finally, we use the cluster labels to determine homophily groups of users. The details of the algorithms are presented in the subsequent sections.

4.1 Multiwave Embedding of Edge Attributed Graphs

We develop an extension to the Graphwave embedding algorithm [6] to incorporate the multiple attributes that we have identified to represent an enterprise social network. We denote this as the *Multiwave* algorithm. We use tensor factorization, for dimensionality reduction and finally compute higher order singular value decomposition to obtain the matrices useful in applying spectral graph wavelets used in obtaining the node embedding upon non linearly weighting them through a convolutional autoencoder. The block diagram of our node embedding algorithm is shown in Fig. 1.

We incorporate attributes as are deemed important for behavior characterization. In a naive graph representation, each type of edge attribute is separated into a graph of its own and considered as a

layer. If this G_i is then naively used as input and the Graphwave algorithm is applied on each layer of the graph to generate structural embedding for each node for that layer χ_i [6] it becomes intractable as the number of features increase. Therefore, we concatenate the adjacency matrices of each graph layer into an adjacency tensor with each entry constituting the weight of the attributed directed edge. We apply the spectral wavelets by splitting the tensor into orthogonal basis matrices and a core tensor [17] using Higher Order Singular Value Decomposition (HOSVD) extending Laplace transforms and Eigen decomposition from simple weighted directed graphs.

$$\mathbb{T} = \mathbb{U}\mathbb{V}\mathbb{W}\mathbb{G}\mathbb{U}^{\mathbb{T}}\mathbb{V}^{\mathbb{T}}\mathbb{W}^{\mathbb{T}} \tag{1}$$

using Tucker decomposition. We used the SKTensor Python 3 library [21]. Here $\mathbb{G}$ is the reduced dimension core tensor, while the orthogonal basis matrices are $\mathbb{U}$, $\mathbb{V}$ and $\mathbb{W}$ with dimensions corresponding to each face of the core tensor $\mathbb{G}$ and original tensor dimensions.

We adjust the values of the parameter τ as per the data ranges and dimensionality of the core tensor. We apply the heat kernel of the spectral wavelets on the core tensor to propagate the effect of perturbation of a node on its neighbors by applying a filter such as the heat kernel $g_s\,(t) = e^{-ts}$ on each element on the core tensor and calculating its effect on each neighbor as

$$\Psi_{ma} = \sum_{l=1}^{N} g_s\left(g_{ijk}\right)\mathbb{U}_{ml}\mathbb{V}_{ml}\mathbb{W}_{ml}\mathbb{U}_{al}\mathbb{V}_{al}\mathbb{W}_{al} \tag{2}$$

which is the effect of node m on node a, where the transformed kernel is combined together with the orthogonal decomposed matrices using tensor times matrices.

where $\delta_a = \mathbb{1}(a)$ is the one hot vector for the node a that isolates the effect of neighbors on node a.

As in [6] algorithm, Ψ gives us the diffusion pattern for every node. It is an $N \times N$ matrix. And similar to Graphwave its ath column vector is the spectral graph wavelet for a heat kernel centered at node a and Ψ_{ma} represents the amount of energy that node a has received from node m. Calculating these values for pair of nodes in the graph is computationally expensive. Therefore, we too like in [6] treat the wavelet coefficients as a probability distribution and characterize the distribution via empirical characteristic functions and sample from it via evenly spaced points to result in the structural embedding as in the Algorithm 1. We take uniformly d points and calculate the characteristic function for each of those points. The column mean for it results in the aggregate effect of neighbors on that point. The real and imaginary values of each of the d points then form the $2d$ length structural node embedding with a depth equal to the number of features as a matrix.

The dimensions of the core tensor are kept at a fraction of the original tensor so as to obtain a good representation. This representative embedding as detailed in the Algorithm 1, results in a tensor created of the values of the directed edge attributes. It is the output of the heat kernel applied as the characteristic function χ whose dimensions consist of the number of nodes, dimension size, and the number of features. Normalization using attributes themselves is not feasible since the relationship between each attribute is nonlinear. Therefore, in order to reduce the dimensionality, we pass it through a deep convolutional autoencoder that is able to compress the representation to a $2d$ embedding vector of depth 1. The input to the autoencoder is the representative χ tensor. A combination of 2D convolution layers with RELU activation and max-pooling with stride length 1 is applied to compress the feature representation down to 1 which is the encoded 2D node embedding for each node in our graph of length twice of the number of sampling points usually 200. A 2D upsampling and 2D convolution layers with RELU activation and final tanh activation layer completes the autoencoder that is compiled and fit with an 80-20 split of train and test data. The number of filters ranges from 8 to 16. This final embedding of each node now contains the structural embedding of the entire graph that is made up of all the layers.

Algorithm 1 Learning Edge Attributed Structural Node Embedding from Adjacency Tensor

Input: Graph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{T})$ where $\mathbb{T}$ is adjacency tensor with feature weights
Input: scale s
Input: evenly space sampling points $\{t_1,t_2,..,t_d\}$
Output: Structural node embedding $\chi_a \in \mathbb{R}^{2d}$ for every node $a \in \mathbb{V}$

Decompose via HOSVD($\mathbb{T}$) = $\mathbb{U}\mathbb{V}\mathbb{W}\mathbb{X}\mathbb{U}^{\mathbb{T}}\mathbb{V}^{\mathbb{T}}\mathbb{W}^{\mathbb{T}}$
Apply heat kernel on core tensor
$\Psi_a = \mathbb{U}\mathbb{V}\mathbb{W}g_s\,(\mathbb{X})\,\mathbb{U}^{\mathbb{T}}\mathbb{V}^{\mathbb{T}}\mathbb{W}^{\mathbb{T}}\delta_a$, where g_s=e^{-ts}
Calculate structural node embedding tensor $\mathcal{X}$
for t $\in \{t_1,t_2,..,t_d\}$ **do**
    Compute $\phi\,(t_j)$ = column-wise mean $\left(e^{it_j\psi}\right) \in \mathbb{R}^N$
    for a $\in \mathbb{V}$ **do**
        Append $\mathfrak{R}\,(\phi_a\,(t))$ and $\mathfrak{I}\,(\phi_a\,(t))$ to χ_a
    end for
end for
Apply 2D Convolutional Autoencoder with Max Pooling on $\mathcal{X}$ to obtain node embedding $\chi_a \in \mathbb{R}^{2d}$

4.2 Clustering using Node Embedding

The embedding captures the structural view of the graph. A simple L_2 (euclidean) distance between embedding tells us how similar the neighborhood around two nodes are in a graph. A low L_2 distance implies that the two nodes have structurally similar neighborhoods. We hypothesize that this would correspond to the similarity in roles as similar roles would require similar communication and interactions.

In order to assess our role similarity correspondence with structural similarity and to ascertain the performance of the node embedding algorithms, we use K-means clustering to cluster the nodes using their embedding with L_2 distance as the distance measure. We also cluster using the cosine distance in a separate experiment since our embedding is high dimensional. Before we run K-means we run principal component analysis (PCA) as a pre-processing step for dimensionality reduction to remove noise.

As we note in the section on experiments, clustering these embedding gives us much superior results than non-attributed graphs in terms of roles being identified and segregated in different clusters; thereby identifying structurally similar neighborhoods around

nodes across the graph. The Davies-Bouldin (DB) index reported in Table 1 indicates the internal consistency and performance of the clustering for different data sets.

4.3 Mining Homophilic Groups using User Cluster Labels

Algorithm 2 Identifying Homophilic Groups using Cluster Labels based on Edge Attributed Node Embedding

Input: Sets $p \in \mathbb{P}$ where $p = (v, c)$, node $v \in \mathbb{V}$, cluster label $c \in \mathbb{C}$
Input: threshold τ
Input: scale s
Output: Sets p of structurally and behaviorally similar subgraphs across the network

```
Initialize ℙ to contain singleton sets (v, c) of all nodes v ∈ 𝕍, where 𝔾 = (𝕍, 𝔼, 𝕋) and corresponding cluster label c ∈ ℂ
for k=1 to s do
    for Each pair p_i,p_k ∈ ℙ do
        Calculate Jaccard Similarity index J_ik = |{c_i}∩{c_k}| / |{c_i}∪{c_k}|
        if J_ik > τ then
            Merge p_i and p_k
            Add one hop neighbors of nodes in the merged set, (v, c) into the set
        end if
    end for
end for
```

Based on our node embedding and clustering, we can identify the role labels of users by their cluster labels. We are basing this group identification algorithm on the assumption that the node embedding that we construct provides us a representation of the structure of its k-hop neighborhood. Therefore, nodes whose representations are similar, have structurally similar neighborhoods of up to k hop away. Also, the clustering algorithm assigns every node a cluster membership where members of the same cluster have structurally similar neighborhoods. As we observe from Graphwave [6] and its application on the data set for clustering using k-means that indeed nodes who have similar role labels fall in the same cluster, which re-affirms the idea that most assigned roles are structurally similar in their communication pattern.

In our homophilic group identification methodology described in Algorithm 2, the initial list contains singleton sets of a node and its cluster label. In each iteration, a pair of sets is taken and the Jaccard similarity of their constituent nodes and k hop neighbors of those nodes, using their cluster labels are calculated as the closeness or similarity between the sets. If this measure is above a threshold, these sets are merged and the constituent nodes are considered in the next iteration. If the composition of two sets of nodes neighbors' cluster memberships are identical, it gives confidence that they are structurally similar and we add these nodes to candidate node sets. This way we increase our bucket until finally at some s hops we have sets of nodes that have near-identical composition of cluster membership ids and thus they can with confidence be considered to be structurally similar subgraphs.

In order to measure the homophily of the peer groups identified, we extend normalized mutual information for groups. The existing role labels for the purpose of this measure are considered to be good. The cluster labels are compared with them using Jaccard similarity. A group is the collection of nodes in the set and its k-hop neighbors. Every hop away from the base node contributes less weight towards how much it affects the homophily of the labeling of the base node. This is captured using a weight to the similarity score as seen in Equation (3).

Our baseline comparison peer groups are identified using Jaccard similarity on the role labels of employees where the groups are k hop neighbors of the node. In our algorithm we leverage the identification of role labels from their cluster id labels that approximate their roles based on their communication and structural attributes. The definition of Jaccard similarity (J) and the goodness of homophilic groups are as follows:

$$\begin{aligned} \text{Jaccard Similarity } J_i &= \frac{|c_i \cap r_i|}{|c_i \cup r_i|} \\ \text{Goodness of group } g &= \sum_{i=0}^{k} J_i * w^i, \\ &\text{where hop weight } w \text{ usually } 0.5 \end{aligned} \tag{3}$$

Cluster quality is measured using the Davies-Bouldin (DB) index and the normalized mutual information (NMI) index. The DB index is defined as follows.

Let M_{ij} represent the average inter-cluster distance between points in cluster i and j. Similarly let S_i and S_j be the average intra-cluster distance between points in clusters i and j respectively. R_{ij} is defined as:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \tag{4}$$

Define -

$$D_i = \max_{i \neq j} R_{ij} \tag{5}$$

If N is the total number of clusters, then the Davies Boulding index is defined as follows:

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{6}$$

A lower value of the DB index implies a better clustering. The Normalized Mutual Information (NMI) is defined as follows:

$$NMI = \frac{2 \times I(Y, C)}{H(Y) + H(C)}, \tag{7}$$

where Y is the actual role label and C is the identified cluster label. $H()$ is the entropy and $I(Y, C)$ is the mutual information between Y and C. A higher value of NMI denotes a better clustering.

In the next section, we present details of the results of our experiments in applying the above homophilic group identification algorithm to the enterprise social network data sets.

5 EXPERIMENTAL RESULTS

In this section, we present details of the data set used in our experiments along with the important attributes available in each of the two data sets. We then present the results of the node embedding and clustering algorithms. Comparison with related works is shown next. Finally, we present a discussion of the results.

5.1 Data Sets

We consider two data sets for our experiments. We collect traces of communication including content and other structured interactions from two sources of enterprise social media, StackExchange Workplace community and Enron email data set. The computable size for StackExchange data set was 20000 posts that had 2181 users and for the Enron data set was 20000 emails that had 2546 employees. For the StackExchange data set, we sampled 20000 posts from the workplace StackExchange involving 2181 users. The StackExchange data set contains comments that are on average 50 words in length and are considered too short to individually contain multiple topics. The Enron data set on the other hand have long emails that may contain multiple topics.

5.1.1 StackExchange Q&A Data Set. We download the StackExchange archive [33] from March 2012 to August 2017 from archive.org using wget [22]. We transform it for ease of processing into PostgreSQL using the script at [35] It consists of 52074 users in a total of which 37086 users have badges or roles. The discussions in this data set are relevant to enterprises. The users have a total diversity of 139 badges or roles. There are no organizationally defined groups here but the role labels are earned based on user activity. The Workplace community itself can be considered as a high-level group with interested users participating in it. It is assumed however that over time, there would be affinity groups that may have emerged within users. It is known that many questions are physically marked as duplicates which indicates that indeed there are groups of users that are interested in the same areas but are not connected with each other directly.

We construct the graph from the StackExchange Workplace community data set where users are nodes with directed edges from the user commenting on a question to the user posting the question. The directed edges are multiedges with weights for each measure of communication relevant to behavior such as number of questions posted (average 1.17, max 23, min 1), number of comments (average 1.17, max 23, min 1), comment sentiment (average 0.14, max 5.28, min -1), popularity score (average 3.12, max 276, min 0) and EMPATH scores (average 0.002, max 0.63, min 0). We know from [37] that likes or score are an important indicator of behavior. For this experiment, we have not used complex linguistic measures other than the overall sentiment of the comments.

5.1.2 Enron Email Data Set. We collect the Enron email data set from [12] using wget [22]. Employee role labels are available for 149 users. It consists of emails of employees who were working at Enron. Just as any organization, we assume that they were redundancy in roles and work context and thus peer groups who are distanced in communication but working towards similar goals. As we represent the communication between employees as a graph, the nodes are employees identified by their email addresses with directed edges from the employee who has replied to an email to the sender employee.

Measures such as the number of replies and the average length of reply correspond to the volume of the communication, behaviorally determining possibly how involved the two people are in communication. Topics and corresponding polarity give us aspect-based sentiment between people. Ratios of each part of speech

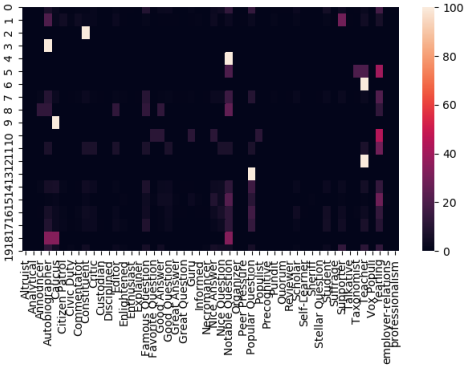

Figure 2: StackExchange user composition in each cluster using Multiwave

usage give us some stylistic hints as well. The average number of named entities and URLs used is also relevant. Excessive use of punctuation, capital letters, vocabulary size provides stylistic hints between people. Ratios from the EMPATH [7] categories provide psychologically tested behavioral measures of the communication. For the email Enron data set, we use measures such as number of emails sent (average 2, max 226, min 1), number of emails received (average 3, max 343, min 1), overall sentiment (average 0.3, max 22.8, min -4.3), number of entities (average 109, max 9972, min 0), number of capitalized words (average 222, max 21504, min 0), number of punctuation (average 137, max 13181, min 0), number of digits (average 19.5, max 2802, min 0), number of URLs (average 10, max 1002, min 0), verbs (average 141, max 13870, min 0), auxiliaries (average 198, max 19680, min 0), symbols (average 18, max 2189, min 0), numbers (average 38, max 5910, min 0), nouns (average 337, max 32349, min 0), adjectives (average 57, max 5403, min 0), adverbs (average 32, max 3420, min 0), pronouns (average 36, max 3848, min 0), number of words (average 1047, max 101549, min 1), size of vocabulary (average 949, max 92491, min 1), EMPATH categories (average 0.004, max 4.31, min 0).

5.2 Results and Comparison

The data sets were stored as a MySQL archive. We pre-process it in batch mode to calculate the attribute measures and store them in the PostgreSQL database for ease of experimentation. We use Spacy [13] to generate the linguistics features from the message text. We use Textblob [19] to calculate the overall sentiment of the message content. We use EMPATH [7] to identify the strength of various psychologically relevant categories from the messages. We create the graph representation using the features as weights of the multiedges using the Networkx library [11].

We apply our node embedding algorithm described in Algorithm 1 on these graphs to generate node embedding for each user of 200 dimensions each. We cluster the users using their embedding using SKlearn [24] Kmeans++ algorithm to take into account initial center choice. We set the number of clusters to 9 as we have 10 different roles in the data set of 149. We take the cluster labels of each user and apply our group discovery Algorithm 2 to identify peer groups amongst the users.

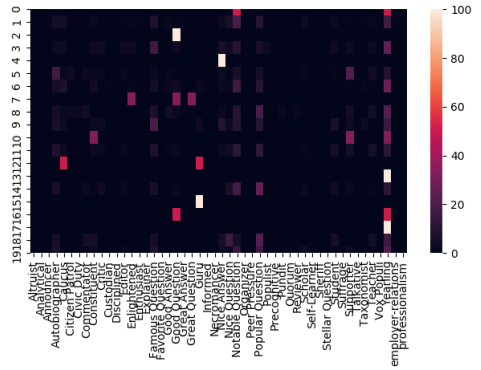

Figure 3: StackExchange user composition in each cluster using Graphwave

Figure 6: Heatmap of roles of Enron employees using Multiwave

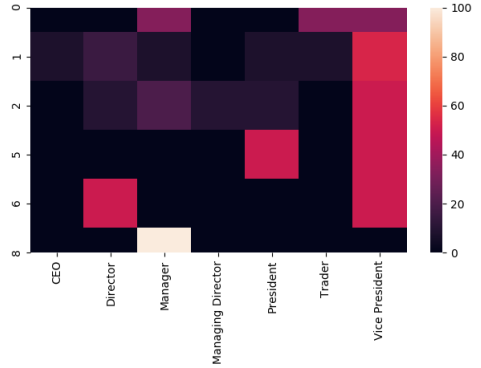

Figure 4: Enron employees composition in each cluster using Multiwave

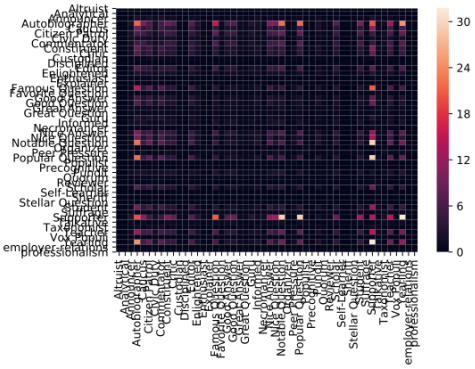

Figure 7: Heatmap of badges of StackExchange users using Multiwave

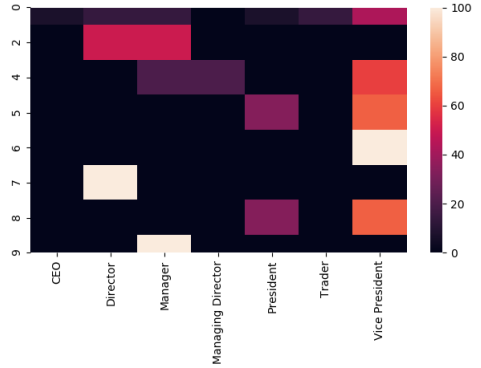

Figure 5: Enron employees composition in each cluster using Graphwave

We ran our experiments on a Dell PowerEdge 730 server with a 32 core Intel Xeon E5-2690 v3 processor and Xeon Phi 7120P co-processor. It has 128GB of RAM. Apart from being compute heavy; we found that the primary bottleneck was memory due to the size of the input tensor which is a Numpy array consisting of Numpy 64 bit floats.

To demonstrate that our node embedding results in a stable measure, upon computing the node embedding for each node in each data set; we use them as distance measures in clustering the users. We run principle component analysis (PCA) to remove the noisy data points. We then run K-means clustering with K-means++ [2] for initialization. We observe that the clustering is stable when we calculate the Davies-Bouldin (DB) Index score for the data sets with our modified algorithm. This provides us evidence that our node embedding captures the structural similarity of their neighborhood.

The cluster composition of the users obtained using the proposed algorithm and the Graphwave algorithm are shown in Fig. 2 and 3 for the StackExchange data set. Corresponding results for the Enron data set are shown in Fig. 4 and 5 respectively.

Using the clustering results and the homophily user group identification method we identify the groups of the user in each of the two data sets. The actual role of the users is available as a ground truth both for the Enron and the StackExchange data sets. In the StackExchange data set the badges represent the role of the users. We plot a heat map of the number of agreements between the ground truth roles and identified groups of the users. The heat

Table 1: Evaluation metrics for algorithms on Enron and StackExchange data sets with Grouping (G) as Homophily (H) and Louvain (L)

Embedding	Data set	DB Idx	Jaccard Score	NMI	G	Group NMI
Multiwave	Enron	0.43	0.973	0.245	H	0.238
					L	0.239
	Stack	2.37	0.201	0.108	H	0.496
					L	0.083
Graphwave	Enron	0.982	0.974	0.312	H	0.039
					L	0.026
	Stack	2.309	0.226	0.087	H	0.496
					L	0.083
Node2Vec	Enron	1.46	0.973	0.042	H	0.039
					L	0.026
	Stack	2.504	0.198	0.093	H	0.496
					L	0.083

map for the Enron data set is shown in Fig. 6. The heat map for Workplace community of the StackExchange data set is shown in Fig. 7.

We run our peer group identification Algorithm 2 upon the results of the clustering for both the data sets; we obtain peer groups where the group members have similar roles as well as their k hop neighbors. For our collected data sets; we compute the goodness measure Normalized Mutual Information (NMI) in Equation (3) between the role labels in our group members' neighbors to measure the overlap. Since the roles of the users are known for the data sets, we compute the Jaccard similarity measure to evaluate the cluster quality. The Davies-Bouldin cluster quality measure is also studied. The results are shown in Table 1.

On manually observing the emails of identified peer groups in the Enron email data set, we see a similar composition of employees in both the groups such as one or two traders; a legal person; a person involved in procurement and travel; a person who questions deals or upcoming trades; and a person who mostly emails about non-work topics such as birthday, new year, lunch. Behaviorally as well, they are both composed of employees sending high volume; short emails; mostly positive sentiment; negative sentiment; neutral; emailing thank yous; emailing sorries. On observing manually the badges and comments of the peer groups in the StackExchange data set, we again see a similar balanced composition of users in both the groups such as those who write nice answers versus nice questions; who are enlightened versus are pundits; who are long term StackExchange users versus are new to the platform; interested in salary, manager versus interviews, telecommute, employee rights versus work-life balance like a holiday, dress code.

We run a comparison with scalar node embedding approaches such as node2vec [9] and Graphwave [6]. We measure the efficacy of the node embedding itself as a distance measure in obtaining clusters of similarly behaving users in Table 1 as the Davies-Bouldin (DB) index of the clustering and the Normalized Mutual Information (NMI) of cluster member's role labels. We observe that our algorithm captures structural similarity in the multi-featured case and results in richer insights as is evident from the better value of the clustering indices.

We also see as opposed to the Graphwave [6] results on Enron email data set; where their node embedding was able to identify the high distance between the top management such as CEO; VP versus the traders. Here in the similar heat map in Fig. 6 ; we observe that we are able to distinguish between more types of roles and therefore our insights are more fine-grained.

We compare our proposed Homophily Group detection algorithm with the popular Louvain [3] community detection algorithm. Comparison is made with respect to the three embedding techniques used namely the proposed Multiwave embedding, the Graphwave algorithm, and the node2vec embedding. The results are presented in Table 1 in terms of the cluster quality measures. Using the Graphwave [6] node embedding and Louvain [3] community detection algorithm we find that the identified groups are not those that can be considered to have the same roles and same neighborhoods within the community from an enterprise perspective. As opposed to that our algorithm is able to identify such groups that are demonstrably semantically relevant from an enterprise perspective as we can see from the Jaccard similarity measures in Table 1 across data sets.

In comparison with Louvain [3], community detection algorithm; we observe significant improvement in identifying relevant groups with similar roles in the enterprise. The comparative measures of the role labels in groups identified in the various data sets with these algorithms are present in Table 1. The modularity measures [20] on the Enron data set with Louvain is 0.676 and on the Workplace data set with Louvain is 0.364. It demonstrates that the use of scalar edge weights is insufficient in capturing the required multi-dimensionality of communication between members in an organization in order to understand group behavior.

6 CONCLUSIONS

We present Multiwave, an embedding technique for edge attributed graphs. The method uses tensor factorization techniques. Considering various attributes of edges like the number of posts, sentiments, likes, etc leads to better representation of a user of an enterprise social media. The embedding is used to cluster the users using homophily groups. The clusters are mapped to role labels using a role labeling algorithm. Homogeneity of the homophily groups identified is evaluated using two cluster quality measures as well as their agreement to ground truth role labels.

Through our experiments on the Enron and StackExchange data set, we are able to demonstrate that our edge attributed edge node embedding algorithm is a stable measure relevant for enterprise roles. Using our homophily group identification algorithm that makes use of these node embedding, we are able to identify good groups within enterprises that have similar work contexts and behavior and are yet not discovered by each other. In future work we intend to consider a temporal evolving dynamic graph representation [15] of interactions as well.

7 ACKNOWLEDGMENT

We would like to thank Sirshendu Pan for help in explaining the Graphwave paper and Abir Naskar for similar discussions.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. 2005. Privacy Preserving OLAP. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data* (Baltimore, Maryland) *(SIGMOD '05)*. Association for Computing Machinery, New York, NY, USA, 251–262. https://doi.org/10.1145/1066157.1066187

[2] David Arthur and Sergei Vassilvitskii. 2007. K-Means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (New Orleans, Louisiana) *(SODA '07)*. Society for Industrial and Applied Mathematics, USA, 1027–1035.

[3] Ayan Kumar Bhowmick, Koushik Meneni, Maximilien Danisch, Jean-Loup Guillaume, and Bivas Mitra. 2020. LouvainNE: Hierarchical Louvain Method for High Quality and Scalable Network Embedding. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 43–51. https://doi.org/10.1145/3336191.3371800

[4] Rui Chen, Qingyi Hua, Bo Wang, Min Zheng, Weili Guan, Xiang Ji, Quanli Gao, and Xiangjie Kong. 2019. A Novel Social Recommendation Method Fusing User's Social Status and Homophily Based on Matrix Factorization Techniques. *IEEE Access* 7 (2019), 18783–18798. https://doi.org/10.1109/ACCESS.2019.2893024

[5] Lipika Dey and Bhakti Gaonkar. 2012. Discovering regular and consistent behavioral patterns in topical tweeting. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, Tsukuba, Japan, 3464–3467.

[6] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning Structural Node Embeddings via Diffusion Wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 1320–1329. https://doi.org/10.1145/3219819.3220025

[7] Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding Topic Signals in Large-Scale Text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4647–4657. https://doi.org/10.1145/2858036.2858535

[8] Palash Goyal, Homa Hosseinmardi, Emilio Ferrara, and Aram Galstyan. 2018. Embedding Networks with Edge Attributes. In *Proceedings of the 29th on Hypertext and Social Media* (Baltimore, MD, USA) *(HT '18)*. ACM, New York, NY, USA, 38–42. https://doi.org/10.1145/3209542.3209571

[9] 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* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 855–864. https://doi.org/10.1145/2939672.2939754

[10] Janine Hacker, Rebecca Bernsmann, and Kai Riemer. 2017. *Dimensions of User Behavior in Enterprise Social Networks*. Springer International Publishing, Cham, 125–146. https://doi.org/10.1007/978-3-319-45133-6_7

[11] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Gaël Varoquaux, Travis Vaught, and Jarrod Millman (Eds.). SciPy, Pasadena, CA USA, 11 – 15.

[12] Jeff Heer and Andrew Fiore. 2015. UC Berkeley Enron Email Analysis. https://bailando.berkeley.edu/enron_email.html.

[13] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://spacy.io/.

[14] Vijay Ingalalli, Dino Ienco, and Pascal Poncelet. 2018. Mining Frequent Subgraphs in Multigraphs. *Information Sciences* 451-452 (Jul 2018), 50–66. https://doi.org/10.1016/j.ins.2018.04.001

[15] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. 2020. Representation Learning for Dynamic Graphs: A Survey. *Journal of Machine Learning Research* 21, 70 (2020), 1–73.

[16] Tamara G. Kolda. 2001. Orthogonal Tensor Decompositions. *SIAM J. Matrix Anal. Appl.* 23, 1 (July 2001), 243–255. https://doi.org/10.1137/S0895479800368354

[17] Tamara G. Kolda and Brett W. Bader. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (September 2009), 455–500. https://doi.org/10.1137/07070111X

[18] Yanbei Liu, Qi Wang, Xiao Wang, Fang Zhang, Lei Geng, Jun Wu, and Zhitao Xiao. 2020. Community enhanced graph convolutional networks. *Pattern Recognition Letters* 138 (2020), 462–468. https://doi.org/10.1016/j.patrec.2020.08.015

[19] Steven Loria. 2020. TextBlob: Simplified Text Processing. https://textblob.readthedocs.io/en/dev/.

[20] Mark E. J. Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69 (Feb 2004), 026113. Issue 2. https://doi.org/10.1103/PhysRevE.69.026113

[21] Maximilian Nickel and Evert Rol. 2019. SKTensor Python3 Library. https://pypi.org/project/scikit-tensor-py3/.

[22] Hrvoje Niksic. 2017. GNU Wget Software. https://www.gnu.org/software/wget/

[23] Thomas Olsson, Jukka Huhtamäki, and Hannu Kärkkäinen. 2020. Directions for Professional Social Matching Systems. *Communications of the ACM (CACM)* 63, 2 (January 2020), 60–69. https://doi.org/10.1145/3363825

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (November 2011), 2825–2830.

[25] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and Node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 459–467. https://doi.org/10.1145/3159652.3159706

[26] Kunal Ranjan and Lipika Dey. 2014. Email Analytics for Support Center Performance Analysis. In *2014 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE Computer Society, Los Alamitos, CA, USA, 810–817. https://doi.org/10.1109/ICDMW.2014.74

[27] Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. 2017. Struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 385–394. https://doi.org/10.1145/3097983.3098061

[28] Aman Roy, Vinayak Kumar, Debdoot Mukherjee, and Tanmoy Chakraborty. 2020. Learning Multigraph Node Embeddings Using Guided Lévy Flights. In *Advances in Knowledge Discovery and Data Mining*, Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan (Eds.). Springer International Publishing, Cham, 524–537.

[29] 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* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 510–518. https://doi.org/10.1145/3336191.3371811

[30] Daniel Schneider, Scott Spurlock, and Megan Squire. 2016. Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List. In *Proceedings of the 12th International Symposium on Open Collaboration* (Berlin, Germany) *(OpenSym '16)*. ACM, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/2957792.2957801

[31] Priyanka Sinha, Lipika Dey, Pabitra Mitra, and Anupam Basu. 2015. Mining HEXACO personality traits from Enterprise Social Media. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Lisboa, Portugal, 140–147. http://aclweb.org/anthology/W15-2920

[32] Priyanka Sinha, Lipika Dey, Pabitra Mitra, and Dilys Thomas. 2020. *A Hierarchical Clustering Algorithm for Characterizing Social Media Users*. Association for Computing Machinery, New York, NY, USA, 353–362. https://doi.org/10.1145/3366424.3383296

[33] StackExchange. 2017. StackExchange Dataset Archive. https://archive.org/details/stackexchange

[34] Sergio L. Toral, Rocío M. Torres, and Federico Barrero. 2009. Modelling Mailing List Behaviour in Open Source Projects: the Case of ARM Embedded Linux. *J.UCS: Journal of Universal Computer Science* 15, 3 (feb 2009), 648–664.

[35] Utkarsh Upadhyay. 2015. StackOverflow data to postgres. https://github.com/Networks-Learning/stackexchange-dump-to-postgres.

[36] Janine Viol, Rebecca Bernsmann, and Kai Riemer. 2015. "Behavioural Dimensions for Discovering Knowledge Actor Roles Utilising Enterprise Social Network Metrics". In *Proceedings of Australasian Conference on Information Systems (ACIS) 2015*. AIS, Adelaide,Australia, 13 pages. https://aisel.aisnet.org/acis2015/17

[37] Wu Youyou, Michal Kosinski, and David Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences* 112, 4 (2015), 1036–1040. https://doi.org/10.1073/pnas.1418680112 arXiv:http://www.pnas.org/content/112/4/1036.full.pdf

[38] Shuo Yu, Feng Xia, Kaiyuan Zhang, Zhaolong Ning, Jiaofei Zhong, and Chengfei Liu. 2017. Team Recognition in Big Scholarly Data: Exploring Collaboration Intensity. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*. IEEE, USA, 925–932. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.155

[39] Shuo Yu, Jin Xu, Chen Zhang, Feng Xia, Zafer Almakhadmeh, and Amr Tolba. 2019. Motifs in Big Networks: Methods and Applications. *IEEE Access* 7 (2019), 183322–183338. https://doi.org/10.1109/ACCESS.2019.2960044

[40] Kaiyuan Zhang, Shuo Yu, Liangtian Wan, Jianxin Li, and Feng Xia. 2019. Predictive Representation Learning in Motif-Based Graph Networks. In *AI 2019: Advances in Artificial Intelligence*, Jixue Liu and James Bailey (Eds.). Springer International Publishing, Cham, 177–188.