TransConv: Relationship Embedding in Social Networks

Yi-Yu Lai, Jennifer Neville, Dan Goldwasser
Computer Science
Purdue University
West Lafayette, IN 47907, USA
{lai49, neville, dgoldwas}@purdue.edu

Abstract

Representation learning (RL) for social networks facilitates real-world tasks such as visualization, link prediction and friend recommendation. Traditional knowledge graph embedding models learn continuous low-dimensional embedding of entities and relations. However, when applied to social networks, existing approaches do not consider the rich textual communications between users, which contains valuable information to describe social relationships. In this paper, we proposed *TransConv*, a novel approach that incorporates textual interactions between pair of users to improve representation learning of both users and relationships. Our experiments on real social network data show *TransConv* learns better user and relationship embeddings compared to other state-of-the-art knowledge graph embedding models. Moreover, the results illustrate that our model is more robust for sparse relationships where there are fewer examples.

Introduction

Representation learning has been applied widely in different areas to extract useful information from data when building classifiers for inferring node attributes or predicting links in graphs. Many previous studies proposed low-dimensional network embeddings to learn graph representations (Cao, Lu, and Xu 2015; Grover and Leskovec 2016; Perozzi, Al-Rfou, and Skiena 2014; Tang et al. 2015; Wang, Cui, and Zhu 2016). When applied to social networks, these models project users to a hyperspace to capture the relational and structural information conveyed by the graph. However in social networks, because a user often has different roles for different relationships, learning a single unique representation for all users/relations may not be effective. For example, a user could be close to a one set of friends because they were college classmates but close to another because they are colleagues at work. To capture this information, is important to consider the characteristics of relationships between users when learning representations of social networks.

Knowledge graphs are multi-relational graphs that are composed of entities as nodes and relations as different types of edges. An edge instance is a triplet of fact (head entity, relation, tail entity). There has been a surge of interest in learning graph representations of social networks by simultaneously learning user and relationship embeddings based on the concept of triplet (Bordes et al. 2013; Ji et al. 2015; Lin et al. 2015; Wang et al. 2014). These methods have considered both network structure and node relations to improve the quality of embedding. At the same time, semantic content of entities can also provide abundant information for representation learning. (Xie et al. 2016; Xiao et al. 2017) take the description of entities into account to incorporate text into embedding learning. However, typically users' descriptions do not provide much information about the relationships between pairs of users.

In this work, we make the observation that social network data often contain *textual communications* among users, and that this information is a valuable signal about the types and strength of relationships between users. However, to date this information has not been used effectively in network embedding methods. To address this, we propose a novel relationship embedding model, *TransConv*.

TransConv is a structural embedding approach using relation hyperplanes, where every relationship can be viewed as a translation of users in the embedding space. To incorporate textual communication into the learned embeddings, we develop two different types of *conversation factors* to include in the objective function when learning the embeddings. Our work was inspired by Word2Vec word embedding model (Mikolov et al. 2013) and knowledge graph completion models (Bordes et al. 2013; Wang et al. 2014). Word2Vec allows people to use vector arithmetic to work with word analogies, for instance, $King - Man + Woman \approx Queen$. This can be interpreted to mean that the relationship between *King* and *Queen* is similar to the one between *Man* and *Woman*. Instead of working with analogies, our model will directly learn vector representations of relationships between users in social networks. Therefore, we aim to leverage ideas from the knowledge graph completion problem to jointly learn representations of entities and relations. We extend previous approaches by incorporating conversation-based factors to improve the learning process. We evaluate *TransConv* on three different classification tasks: social network completion, triplets classification, and multilabel classification. The experimental results show that our approach outperforms other state-of-the-art models on two real-world social network datasets, and notably it improves prediction accuracy for both frequent and infrequency relations.

Problem Formulation

In social network data, we have a set of users ($U = \{u_i\}$), user attributes ($\mathbf{X} = \{X_i\}$) collected from user profiles and their group memberships, and messages exchanged among the users ($D = \{d_{ij}\}$). More specifically document $d_{ij} \in D$ represents the set of posts t_{ij} sent from u_i to u_j. Relationships between pairs of users are either defined by the attribute values that they share in common (e.g., $r_k(i, j) = 1$ if $x_i = x_j = k$ and 0 otherwise), or defined by certain directional attributes (e.g., u_j is u_i's top friend, u_j is senior to u_i). Given attribute values of interest in the data, we define a set of relations ($R = \{r_k\}$).

Given a pair of user and their relation as a triplet (u_i, r, u_j), the goal of this work is to learn a joint embedding for users and relationships, such that every relation can be viewed as a translation of users in the embedding space. Let $\hat{u}_i$ and $\hat{u}_j$ denote the user embeddings of u_i and u_j, and $\hat{r}$ denote the relationship embedding of r. The embedding $\hat{u}_i$ is close to $\hat{u}_j$ by adding the relationship embedding $\hat{r}$ (i.e., $\hat{u}_i + \hat{r} \approx \hat{u}_j$). The embeddings of users and relationships are in the same space $\in \mathbb{R}^k$. Let Δ denote the set of golden (positive; true) triplets, for which the relationship holds in the data and $\Delta'_{(u_i,r,u_j)}$ stand for the set of negative triplets constructed by corrupting a golden triplet (u_i, r, u_j).

Previous work on structural embedding began with TransE (Bordes et al. 2013), which first adopted the concept to learn entity embedding of knowledge bases (KB). TransE (Bordes et al. 2013) assumes the error $||\hat{u}_1 + \hat{r} - \hat{u}_2||_{l1/l2}$ is low if (u_1, r, u_2) is a golden triplet. This works well for irreflexive and 1-to-1 relations but fails to deal well with reflexive, N-to-1, 1-to-N or N-to-N relations. TransH (Wang et al. 2014) addressed the issues of TransE by introducing relation-specific hyperplanes w_r. Several models, such as TransR (Lin et al. 2015) and TransD (Ji et al. 2015), then extended TransH and enhanced the embedding performance by learning mapping matrices to relation spaces.

However, the previous work only considers the network structure among entities and ignore the textual information content in messages. In this work, we aim to exploit the message information among users to improve the learned embedding, e.g., by automatically identifying content relevant to particular relations. Specifically, when modeling people's relationships in social networks, we consider a sophisticated model to utilize the "interaction" between two users rather than design a complicated hyperplane projection. For example, u_1, u_2, and u_3 are three users who described themselves as supporters for the same political party, but (u_1, u_2) discuss politics extensively and (u_1, u_3) rarely discuss it. Let's denote $r_{politics}$ as "the same political party". If we model the relation $r_{politics}$ by TransH or its extended models, they treat the triplets $(u_1, r_{politics}, u_2)$ and $(u_1, r_{politics}, u_3)$ in the same way (because they do not consider the content of discussion between users). In contrast, our approach will focus more on (u_1, u_2) than (u_1, u_3) when learning the embedding, under the assumption that the frequent discussion indicates a stronger relationship with respect to $r_{politics}$.

More specifically, to incorporate interaction information in the embedding, we define two new conversational factors to use during learning: *conversation similarity* and *conversation frequency* (defined below). Using these new factors, we then outline our novel relationship embedding model, *TransConv*.

Conversation Similarity Factor

To capture the textual similarity of the interaction between a user pair regarding a particular relation, we define a *conversation similarity* factor $\mu^r_{i,j}$. The factor represents the textual similarity of the interaction between a user pair (u_i, u_j) with respect to relation r, based on the documents d_{ij} and d_{ji} (the collection of messages between u_i and u_j). We compute $\mu^r_{i,j}$ as follows:

1. First, we identify the most representative set of words for each relation $r \in R$. To do this, we collect the set of pairs (u_i, u_j) with relation r and concatenate all their posts into a single (large) document D_r. We repeat this process for each of the relations in R. From the resulting documents, we compute the TF-IDF (Salton 1991) values for each word in each document D_r. TF-IDF scores are widely used as a numerical statistic to reflect how important a word is to a document in a collection. Then for each document D_r, we identify the top-K words with largest TF-IDF values and use those as the representative words as the dictionary W_r for the relation r.
2. Next, we compute a word existence vector (denoted as $wv_{r_{ij}}$ and $wv_{r_{ji}}$) based on the dictionary W_r to transform the textual interactions between u_i and u_j with regard to relation r. This tracks whether the pair has used the words that are representative to the relation r. For each word w in W_r, the value is set to 1 if w exists in the posts d_{ij} (or d_{ji}), otherwise it is set to 0.
3. Finally, we use the word existence vectors to compute the conversation similarity factor for each user pair using the similarity function SIM (e.g., cosine similarity): $\mu^r_{i,j}$ = $SIM(wv_{r_{ij}}, wv_{r_{ji}})$. This tracks whether the pair uses similar words from the relation r in their communication back and forth. We repeat this for every $r \in R$.

The similarity factor $\mu^r_{i,j}$ measures whether u_i and u_j's mutual discussion is relevant to r, and evaluates the degree of affinity between u_i and u_j.

Conversation Frequency Factor

We define a *conversation frequency* factor ϕ^r_{ij} to represent the strength of the interaction between a user pair (u_i, u_j) with respect to relation r. In this factor, we also use the relation dictionaries $\mathbf{W_r}$ from steps 1-2 above.

We first define $out^r(u_i, u_j)$ as the sum of fraction of words in dictionary W_r used in the messages from u_i to u_j:

$$out^r(u_i, u_j) = \sum_{p=1}^{m} \frac{|w^r_p|}{|w_p|} \tag{1}$$

Here m is the number of messages from u_i to u_j. w_p is the set of words used in message p. w^r_p is the intersection of w_p and W_r. Note that the more u_i communicates with u_j, using words relevant to relation r, the the value of $out^r(u_i, u_j)$ will be larger.

Next, we define the conversation frequency factor ϕ_{ij}^r, which reflects to the intensity of interaction between two users with respect to relation r, compared to other users:

$$\phi_{ij}^r = \frac{out^r(u_i, u_j)}{\sum_{k=1}^{n} out^r(u_i, u_k)}, \forall u_k \in \{u_1, u_2, ..., u_n\} \quad (2)$$

If u_i interacts more frequently with u_j compared to other users, the the frequency factor will be larger. The factor can also distinguish whether the interaction between u_i and u_j is one-way or two-way.

After computing the above factors $\{\mu_{i,j}^r\}$ $\{\phi_{i,j}^r\}$ for each relation, we will use them to weights the errors of triplets used in the embedding objective. We do not consider the documents D further.

TransConv: Translating on Conversation

In our *TransConv* model, we assume that people who have similar (stronger) textual interactions would share similar (stronger) relationships, which can be used to improve the learned embeddings. That is, their relationships can be translated better with the aid of their conversations. To achieve this goal, we jointly incorporate the conversation similarity factors $\{\mu_{i,j}^r\}$ and frequency factors $\{\phi_{i,j}^r\}$ introduced in last section when learning user and relationship representations.

For a triplet $(u_i,\ r,\ u_j)$, we learn the relationship-specific hyperplane w_r for relation r as well as the user embeddings $\hat{u}_i$ and $\hat{u}_j$ by projecting users on the relationship hyperplane. The projections are denoted as $\hat{u}_{i\perp}$ and $\hat{u}_{j\perp}$, respectively. If $(u_i,\ r,\ u_j)$ is a golden triplet, the aim is to ensure that $\hat{u}_{i\perp}$ and $\hat{u}_{j\perp}$ are connected by a translation vector $\hat{r}$ on the hyperplane with low error measured by $||\hat{u}_{i\perp} + \hat{r} - \hat{u}_{j\perp}||_{l_{1/2}}$.

We define a score function $f_r(u_i, u_j)$ to assess the quality of the embeddings for u_i and u_j wrt relation r, and weight the score using their conversation similarity and frequency factors:

$$f_r(u_i, u_j) = [1 + \alpha\mu_{ij}^r + (1-\alpha)\phi_{ij}^r] \cdot ||\hat{u}_{i\perp} + \hat{r} - \hat{u}_{j\perp}||_{l_{1/2}} \quad (3)$$

Here α is a tunable parameter for assigning different learning weights to the similarity factor μ_{ij}^r and frequency factor ϕ_{ij}^r. The two factors play important roles augmentating the score function f_r. By constraining $||w_r||_2 = 1$, we formulate $\hat{u}_{i\perp}$ and $\hat{u}_{j\perp}$ as:

$$\begin{aligned} \hat{u}_{i\perp} &= \hat{u}_i - w_r^T \hat{u}_i w_r \\ \hat{u}_{j\perp} &= \hat{u}_j - w_r^T \hat{u}_j w_r \end{aligned} \quad (4)$$

Then the score function $f_r(u_i, u_j)$ can then be rewritten as:

$$\begin{aligned} f_r(u_i, u_j) =& [1 + \alpha\mu_{ij}^r + (1-\alpha)\phi_{ij}^r] \cdot \\ & ||(\hat{u}_i - w_r^T \hat{u}_i w_r) + \hat{r} - (\hat{u}_j - w_r^T \hat{u}_j w_r)||_{l_{1/2}} \end{aligned} \quad (5)$$

The score is expected to be lower for golden triplets and higher for negative triplets. Since golden triplets with affinity (i.e., higher *similarity*) and stronger (i.e., higher *frequency*) interactions are weighted more heavily in the objective, the optimization will pay more attention to reducing the translation error for those triplets.

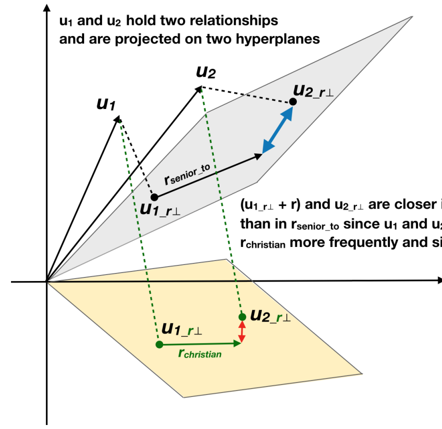

Figure 1: Simple illustration of *TransConv*.

The concept of *TransConv* is illustrated in Figure 1. We simultaneously learn the user embeddings for u_1 and u_2 as well as the relationship embeddings for r_{senior_to} and $r_{christian}$. When u_1 and u_2 have more conversations related to a certain relation, *TransConv* minimizes the score $f_r(u_1, u_2)$ further. In other words, if u_1 and u_2 have two relations r_{senior_to} and $r_{christian}$, but they use more words relevant to $r_{christian}$ compared to r_{senior_to}, *TransConv* will attempt to minimize $f_{r_{christian}}(u_i, u_j)$ more than $f_{r_{senior\_to}}(u_i, u_j)$. As illustrated in Figure 1, during the training phase, $f_{r_{christian}}(u_i, u_j)$ (i.e., the distance of the red double-headed arrow) is minimized compared to $f_{r_{senior\_to}}(u_i, u_j)$ (i.e., the distance of the blue double-headed arrow).

By considering projections on relational hyperplanes along with the augmentation of our proposed conversation factors, *TransConv* can encode different representations for each user, which depends on his/her relationships with others as well as the similarity and frequency of their textual discussions.

Optimization

In order to maximize the difference between golden triplets and negative triplets, we define our loss function as:

$$L = \sum_{\substack{(u_i, r, u_j) \in \Delta \\ (u'_i, r, u'_j) \in \Delta'_{(u_i, r, u_j)}}} [f_r(u_i, u_j) + \gamma - f_r(u'_i, u'_j)]_+ \quad (6)$$

Here $[x]_+ \triangleq max(x, 0)$ and $\gamma > 0$ is the discriminative margin separating golden and negative triplets. The loss function sums over a corrupted negative triplet for each golden triplet (described more below). We adopt stochastic gradient descent (SGD) to minimize the above loss function. When minimizing the loss function, we enforce constraints as $\forall u \in U$, $||u||_2 \leq 1$ and $\forall r \in R$, $||w_r||_2 = 1$.

Initially, we construct the sample data from only golden triplets in Δ. In order to reduce false negative instances, we follow TransH (Wang et al. 2014) and apply Bernoulli sampling method to sample negative triplets. For each golden triplet $(u_i,\ r,\ u_j)$ in Δ, our approach samples one negative triplet from $\{(u'_i, r, u_j) \mid u'_i \neq u_i, u'_i \in U\} \cup$

$\{(u_i, r, u'_j) \mid u'_j \neq u_j, u'_j \in U\}$ and adds it to Δ'_{u_i,r,u_j}. We assign different probabilities for replacing the head user (u_i) or the tail user (u_j) when corrupting the triplet, which depend on the mapping property (i.e., 1-to-N, N-to-1, and N-to-N) of the relation. Among all the triplets of a relation r, let tph denote the average number of tail users per head user and hpt denote the average number of head users per tail user. Then we define a Bernoulli distribution with parameter $\frac{tph}{tph+hpt}$ for sampling: given a golden triplet (u_i, r, u_j), we corrupt the triplet by replacing the head user with probability $\frac{tph}{tph+hpt}$, and we corrupt the triplet by replacing the tail user with probability $\frac{hpt}{tph+hpt}$.

Related Work

Network Embedding Models

There has been increasing attention on low-dimensional graph embedding recently. Many approaches have been proposed for data visualization, node classification, link prediction, and recommendation. DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) predicts the local neighborhood of nodes embeddings to learn graph embedding. LINE (Tang et al. 2015) learns feature representations in first-order proximity and second-order proximity respectively. GraRep (Cao, Lu, and Xu 2015) learns graph representation by optimizing k-step loss functions. Node2Vec (Grover and Leskovec 2016) extends DeepWalk with a more sophisticated random walk procedure and explores diverse neighborhoods. Although many studies have reported their performance on social network datasets, we argue that the actual social networks are more complicated. Users in social networks could have different neighbor structures based on different relationships. Jointly learning representations for users and relationships can help to describe users in social networks more precisely.

Structural Knowledge Graph Embedding Models

The main stream of structural embedding models follows the basic idea that every relation is regarded as translation in the embedding space. The embedding of one entity, say h, is close to another embedding, say t, by adding a relation vector r. A triplet (h, r, t) could be described as the equation $h + r \approx t$. TransE (Bordes et al. 2013) first adopted the concept to learn entity embeddings in knowledge bases (KBs). However, TransE does not perform well on relations with reflexive (i.e., r is a reflexive map for triplets (h, r, t) and (t, r, h)), 1-to-N, N-to-1, and N-to-N properties. TransH (Wang et al. 2014) addressed this issue by introducing relationship hyperplanes so entities can be represented differently with respect to different relations. TransR (Lin et al. 2015) considers that entities and relations should be projected into different embedding spaces and mapped together by mapping matrices of relations. TransD (Ji et al. 2015) extends TransR but reduces its complexity by constructing two dynamic mapping matrices for each triplet and replacing matrix-vector multiplication operations by vector operations. Structural embedding models perform well for entity embedding in KBs, however, they only consider the network structure of entities—they do not use any information about textual communication. Since our study focuses on relationship and user embeddings in social networks, we conjecture that the textual communication between users plays an especially crucial role.

Text-aware Knowledge Graph Embedding Models

Some studies have introduced text-aware embeddings, which attempt to represent the knowledge graph with textual information. DKRL (Xie et al. 2016) proposed an encoder architecture with continuous bag of words (CBOW) and convolutional neural network (CNN) to learn entity embeddings based on network structure and entity description. SSP (Xiao et al. 2017) introduced semantic hyperplanes to capture semantic relevance and correlate entity descriptions to certain topics. These models perform well on knowledge graph embeddings, however, they only consider the description of entities as their textual information. Since our goal is to leverage the impact of interaction and communication between users in social networks, the user description does not provide sufficient details to describe these relationships between users. TransRev (Garcia-Duran et al. 2018) learned a text representation for each pair of heterogeneous source and target nodes. Unlike knowledge graph models, it learned the ”relationship” (i.e., textual review representation) between every user-product pair rather than a global relationship representation. Thus, the relationship learned from TransRev is incapable of modeling multiple relationships between a node pair like our proposed model.

Specifically, existing representation learning models for knowledge graphs only consider the information of each entity itself and then build a translative bridge to interpret the relation of two entities. As such, applying the existing models directly to social networks will disregard meaningful information because textual interactions between users can be important signals of the relationship of users. For example, messages between users suggest the topics they have in common. Those interactions enable us to estimate the strength of relationships and to further identify specific types of relationships among users. It facilitates a more accurate learning of hidden representations in social networks.

Comparison of TransConv to Related Work

To highlight differences with prior work, we list the score functions of related models in Table 1. The embeddings of user u_i and u_j are represented by vectors $\hat{u}_i$ and $\hat{u}_j \in \mathbb{R}^k$. In contrast with these models, which do not include textual communication in their score functions, we use the proposed conversation factors to augment the *TransConv* objective.

Experiments

We evaluate our approach and related methods on three various tasks: social network completion, triplets classification and multilabel classification.

Data

We analyze two social network datasets in our experiments:

- The public Purdue Facebook network data from March 2007 to March 2008, which includes 3 million post activities. There are 211,166 triplets with 19,409 users. For

Model	Score function $f_r(u_i, u_j)$
TransE	$\|\|\hat{u}_i + \hat{r} - \hat{u}_j\|\|_{l_{1/2}}; \hat{r} \in \mathbb{R}^k$
TransH	$\|\|(\hat{u}_i - w_r^T \hat{u}_i w_r) + \hat{r} - (\hat{u}_j - w_r^T \hat{u}_j w_r)\|\|_{l_{1/2}}$; $w_r, \hat{r} \in \mathbb{R}^k$
TransR	$\|\|M_r \hat{u}_i + \hat{r} - M_r \hat{u}_j\|\|_{l_{1/2}}$; $M_r \in \mathbb{R}^{n\times k}; \hat{r} \in \mathbb{R}^n$
TransD	$-\|\|M_{u_i r} \hat{u}_i + \hat{r} - M_{u_j r} \hat{u}_j\|\|_{l_{1/2}}$; $M_{u_i r}, M_{u_j r} \in \mathbb{R}^{n\times k}; \hat{r} \in \mathbb{R}^n$
DKRL	$\|\|\hat{u}_i + \hat{r} - \hat{u}_j\|\|_{l_{1/2}} + \|\|\hat{d}_i + \hat{r} - \hat{d}_j\|\|_{l_{1/2}} +$ $\|\|\hat{d}_i + \hat{r} - \hat{u}_j\|\|_{l_{1/2}} + \|\|\hat{u}_i + \hat{r} - \hat{d}_j\|\|_{l_{1/2}}; \hat{r} \in \mathbb{R}^k$
TransConv	$[1 + \alpha\mu_{ij}^r + (1-\alpha)\phi_{ij}^r] \cdot \|\|w_r^T \hat{u}_i w_r + \hat{r} - w_r^T \hat{u}_j w_r\|\|_{l_{1/2}}$ $w_r, \hat{r} \in \mathbb{R}^k$

Table 1: Score functions of embedding models.

every triplet (u_i, r, u_j), u_i posts at least one message (conversation) on u_j's timeline and vice versa. We construct 41 relationships from user attributes, groups and top friends information.

- Our Twitter dataset is sampled from the dataset collected by (Kwak et al. 2010). It contains 20 million post activities from June to July 2009. There are 300,985 triplets with 22,729 users. We use the posts with user mentions (e.g., "@david happy birthday!") as textual interactions. The 42 relationships types are constructed from user profiles and follower/following information.

We follow TransE (Bordes et al. 2013) to categorize relationships into four categories. In the Facebook (Twitter) dataset, there are 10.6% (23.6%) 1-to-1, 2.6% (6.6%) 1-to-N, 2.6% (6.6%) N-to-1 and 84.2% (63.2%) N-to-N relationships in generated triplets. Table 2 reports the statistics of two datasets. Compared to knowledge base datasets, our datasets is more challenging since it contains more N-to-N complex relationships. Table 3 lists the top-3 most frequent and bottom-3 least frequent relationships from the overall set of 41 (Facebook) and 42 (Twitter). Overall, relationships with more examples have more textual conversations associated with them.

Dataset	#User	#Rel	#Train	#Valid	#Test
Facebook	19,409	41	126,963	42,101	42,102
Twitter	22,729	42	180,606	60,189	60,190

Table 2: Statistics of datasets.

Experiment Settings

We evaluate *TransConv* compared to several knowledge graph embedding models: transE, transH, transR, transD, and DKRL. Both structural and text-aware embedding models are included. We follow the details in the papers to implement these models, and compare the performance of the above models by applying them on our social network datasets.

We perform stratified sampling to split the dataset into 80% training set and 20% validation set. The best configurations are selected based on the performance of validation set. Next, we perform *10-fold* cross validation on testing set

Relationship	#Sample	#Conversation
Facebook		
top-3		
gender-male	29,818	89,060
looking-for-friendship	24,522	94,231
interested-in-women	23,776	73,860
bottom-3		
religious-view-hindu	42	124
hometown-california	34	139
relationship-status-complicated	10	86
Twitter		
top-3		
unverified-account	38,332	133,604
is-followed-by	36,883	128,370
uploaded-profile-image	33,496	113,279
bottom-3		
language-italian	20	83
location-canada	8	24
language-indonesian	4	17

Table 3: Most- and least-frequent relationships in Facebook and Twitter datasets.

and report the average results. In training *TransConv*, we perform grid search over learning rate R for SGD among {0.001, 0.005, 0.01}, the batch size B among {100, 500}, the number of training epochs T among {200, 500}, the margin γ among {0.5, 1.0, 1.5}, the embedding dimension k among {100, 200, 300}, the norm used in score function among {*L1*-norm, *L2*-norm}, the top-K TF-IDF among {100, 500, 1000, 1500, 2000, 2500} and the learning weight α for conversation factors between 0 and 1. For the Facebook dataset, the optimal configurations of *TransConv* are: R = 0.001, B = 100, T = 500, γ = 1.0, k = 300, norm = *L1*-norm, K = 2000 and α = 0.5. The sensitivity of selecting α and top-K TF-IDF is reported in Figure 2. The best α we have is 0.5 and it suggests both conversation similarity and frequency factors take important roles in learning embeddings. The same configurations are applied to the Twitter dataset.

We follow the same process to select corresponding best configurations for other models. In training DKRL model, it is required to include textual information of each entity. In its original work (Xie et al. 2016), each entity's description is composed of a set of keywords selected from the entity's Wikipedia page. However, there is no direct textual description for Facebook and Twitter users. Therefore, we concatenate all the messages posted by an user as a document, and select keywords with top-K TF-IDF score to represent the user's textual description. We select $K = 1500$ for Facebook and $K = 2000$ for Twitter. Next, we apply Google's pre-trained Skip-Gram model (Mikolov et al. 2013), which is trained on part of Google News dataset (about 100 billion words), to generate each entity's description-based representation. Finally, we concatenate the learned description and structure-based representations for DKRL's prediction tasks.

Model	Mean Rank		Mean Hits@N (%)							
	Raw	Filter	N=10		N=5		N=3		N=1	
			Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	305	304	50.6	52.3	37.3	39.9	27.3	30.3	11.4	13.5
TransH	168	168	73.8	76.3	57.5	62.2	43.1	49.0	18.7	23.7
TransR	195	194	75.5	78.7	56.3	61.9	41.6	48.0	18.0	22.7
TransD	295	294	50.6	52.2	37.3	40.0	27.5	30.5	11.4	13.8
DKRL(CBOW)+TransE	5,579	5,577	5.5	6.7	3.4	3.9	2.3	2.3	0.9	1.1
TransConv	**36**	**35**	**83.5**	**86.9**	**63.0**	**68.8**	**46.5**	**53.0**	**20.0**	**24.8**

Table 4: Evaluation results of link prediction on Facebook dataset.

Model	Mean Rank		Mean Hits@N (%)							
	Raw	Filter	N=10		N=5		N=3		N=1	
			Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	203	201	47.6	49.1	39.2	42.0	32.7	36.3	18.8	24.2
TransH	33	32	90.6	92.4	84.3	89.8	76.4	86.9	49.6	74.0
TransR	23	21	93.8	96.2	86.5	92.3	77.9	87.7	51.3	72.1
TransD	199	197	48.2	49.8	39.9	42.6	33.5	37.2	19.5	25.2
DKRL(CBOW)+TransE	5,706	5,704	1.1	1.1	0.7	0.7	0.5	0.5	0.2	0.2
TransConv	**9**	**5**	**95.6**	**98.3**	**88.6**	**95.7**	**80.2**	**91.6**	**52.2**	**74.2**

Table 5: Evaluation results of link prediction on Twitter dataset.

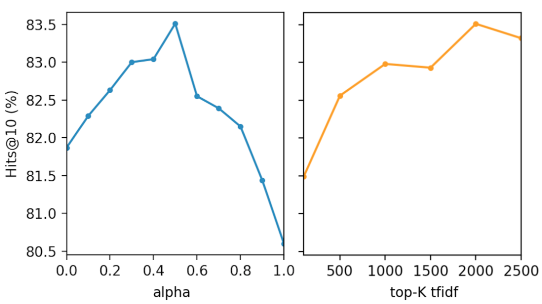

Figure 2: Sensitivity wrt α and top-K TF-IDF on Facebook.

Social Network Completion

In this experiment, we evaluate whether the learned user and relationship embeddings are useful in predicting the existence of user pairs that actually have certain relationships.

The task is to complete a golden triplet (u_i, r, u_j) by minimizing the score function $f_r(u_i, u_j)$, as defined in Table 1, when u_i or u_j is missing. For example, we predict u_j given (u_i, r) or predict u_i given (r, u_j). We follow the same protocol used by TransE (Bordes et al. 2013). First, we compute the "raw" scores for those corrupted triplets and rank them in ascending order, then get the rank of the original golden triplet. Additionally, it is possible that a corrupted triplet exists in the graph and is ranked before the original triplet. This case should not be considered as wrong, so we also compute the "filter" scores to eliminate the factor. The mean rank of correct users and *Hits@N*, the proportion of correct users in top-N ranked users, are reported in Table 4 and Table 5. A lower mean rank is better while a higher *Hits@N* is better. The results show that *TransConv* consistently outperformed other models and achieved 86.9% on Facebook dataset and 98.3% on Twitter dataset with filter setting in *Hits@10*. The results in bold statistically significant outperform other models at 0.01 level in paired t-tests.

First, it is interesting that *TransConv*, TransH, and TransR

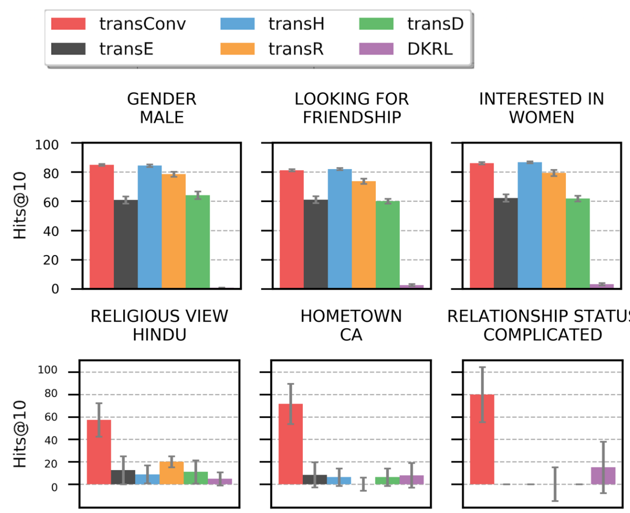

Figure 3: Evaluation results for link prediction on most frequent (top row) and least frequent (bottom row) relationships n Facebook dataset.

have the top-3 highest mean rank among models in both datasets, which shows projecting matrices to relation hyperplanes and spaces is effective when learning embeddings for social network data. Secondly, the performance difference between *TransConv* and TransH suggests that considering the text similarity and communication intensity between users improves the embedding learning significantly. Thirdly, as the reported results of different α values in Figure 2, it further indicates text similarity and communication intensity are complementary factors since neither $\alpha = 0$ nor $\alpha = 1$ achieve the best result. Furthermore, it is noticeable that DKRL model did not perform well with both datasets and that might be caused by the way how we generate the textual description for users. Unlike texts in Wikipedia page are used to define and describe an entity, the collected mes-

Model	Mean Rank				Mean Hits@10 (%)							
					Predicting *head user*				Predicting *tail user*			
Relationship Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE	1054	199	165	216	21.6	54.1	56.5	55.8	23.5	54.1	60.9	55.9
TransH	1062	65	147	58	22.2	74.2	59.4	83.7	21.7	73.7	63.6	83.7
TransR	230	361	331	180	73.2	**81.5**	77.7	79.6	71.7	76.8	80.3	79.5
TransD	1023	188	202	208	23.3	56.1	48.6	56.7	23.3	51.6	54.3	54.9
DKRL(CBOW)+TransE	5,612	5,758	5,323	5,575	4.6	4.5	3.3	5.8	4.6	6.7	2.2	5.7
TransConv	**47**	**31**	**30**	**35**	**88.2**	79.7	**80.6**	**86.8**	**88.6**	**84.3**	**83.7**	**87.3**

Table 6: Detailed results by relationship categories with *Filter* setting on Facebook dataset.

Model	Mean Rank				Mean Hits@10 (%)							
					Predicting *head user*				Predicting *tail user*			
Relationship Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
TransE	136	41	57	256	57.6	73.5	67.4	42.2	57.9	69.7	71.3	40.5
TransH	91	16	5	8	76.8	90.5	93.2	**98.4**	76.8	89.1	95.6	98.3
TransR	64	**3**	**3**	9	94.4	95.4	94.6	95.0	94.2	95.0	95.3	94.9
TransD	111	76	76	254	64.6	64.4	61.9	40.9	63.1	65.2	68.5	41.9
DKRL(CBOW)+TransE	5,588	5,796	5,511	5,758	1.1	0.6	1.0	1.0	1.4	1.1	1.1	1.1
TransConv	**6**	6	6	**4**	**98.1**	**98.3**	**98.0**	**98.4**	**98.2**	**98.1**	**97.6**	**98.5**

Table 7: Detailed results by relationship categories with *Filter* setting on Twitter dataset.

sages could be very casual, noisy and short of meaningful words to depict a user.

We further investigate the performance on each relationship category and report the results of mean rank and *Hits@10* in Table 6 and Table 7. In order to ascertain the consistency, we also evaluate the results of replacing head or tail users. In Figure 3, we examine more closely on the top-3 most frequent and bottom-3 least frequent relationships of the Facebook dataset. Generally speaking, all models have higher *Hits@10* scores and lower score variances on top-3 relationships. The results show all models perform stable on relationships that contain more samples of golden triplets. In top-3 relationships, *TransConv* and TransH outperform others and there is no significant performance difference between *TransConv* and TransH. It reconfirms that it is helpful to include relationship hyperplane projection for social network datasets. However, when examining the bottom-3 relationships, *TransConv* still achieves nearly over 60% in *Hits@10* and outperforms others; While the performances of other models, including TransH, have dropped significantly to lower than 20%. In addition, only *TransConv* and DKRL achieve over 10% in the bottom-1 relationship. It suggests that incorporating textual information is beneficial in learning social relationship representation. We do not include the figure for Twitter dataset here due to space limitation, but the result is also consistent with the result on Facebook dataset. In overall, *TransConv* consistently performs better on both top-3 and bottom-3 relationships and shows more robustness for lack of training samples.

Triplets Classification

In this task, we evaluate if the score function of *TransConv* is effective in discriminating golden and negative triplets by binary classification. For a triplet (u_i, r, u_j), it is predicted as positive if its score $f_r(u_i, u_j)$ is lower than the threshold σ_r; Otherwise, it is predicted as negative. The relation-specific threshold σ_r is determined by maximizing the classification accuracy on the validation set. It requires negative labels to perform the evaluation. We follow the same setting in TransE (Bordes et al. 2013) to construct negative examples for Facebook dataset resulting with equal number of positive and negative examples, and we further discuss three negative sampling strategies by replacing head users, replacing tail users or randomly selecting head (tail) users to replace. When constructing a negative triplet, we constrain the replaced users by only allowing users in a position if they appeared in that position and was ever in that relationship with others in the dataset. For example, with the strategy of replacing tail users, given a correct triplet ($user_7$, $is\_top\_friend\_of$, $user_{30}$), a potential negative example is ($user_7$, $is\_top\_friend\_of$, $user_{15}$). The $user_{15}$ adds other users as his (her) top friends on Facebook (in the position of tail user) but not including $user_7$.

We compare the performance of knowledge graph and networking embedding models and report classification accuracy on eight selected relationships in Table 8. We first select top-5 most frequent relationships, which all happen to be N-to-N category in both Facebook and Twitter dataset. We also include the relationship with largest number of triplets in 1-to-1, 1-to-N, and N-to-1 category respectively to consider all relationship categories in our experiments. For knowledge graph models, different score functions as described in Table 1 are evaluated. For network embedding models, since they do not learn relation embedding, we concatenate the learned user embeddings of each pair (u_i, u_j) as a feature vector $e_{u_i} \oplus e_{u_j}$ and train a binomial logistic regression model for each relation r. That is, if (u_i, r, u_j) is a golden triplet, the label of input $e_{u_i} \oplus e_{u_j}$ is true, otherwise false. The results show the score function of *TransConv* significantly outperforms other models in triplets binary classification. With incorporating conversation factors, it enables our proposed score function to identify golden/negative triplets in more precise way. In addition, we evaluate the results on different relationship categories. In Figure 4, we show the four relationships of Facebook dataset that each one contains the largest number of triplets

Dataset	Facebook			Twitter		
Negative Sampling	replace *head user*	replace *tail user*	replace *random*	replace *head user*	replace *tail user*	replace *random*
TransE	79.3	81.7	75.0	64.4	64.1	62.8
TransH	71.3	71.2	67.9	65.3	65.4	63.4
TransR	91.7	91.6	82.7	80.6	80.5	78.5
TransD	67.0	67.1	63.6	63.9	63.8	62.8
DKRL(CBOW)+TransE	50.3	50.3	52.1	50.0	50.0	44.0
Node2Vec	74.7	74.2	75.1	66.6	68.3	65.8
LINE(1st+2nd)	75.1	73.5	76.1	65.8	64.8	65.8
TransConv	**94.9**	**94.9**	**83.5**	**99.9**	**99.9**	**88.2**

Table 8: Mean accuracy (%) for triplet binary classification on selected relationships with different negative sampling strategies.

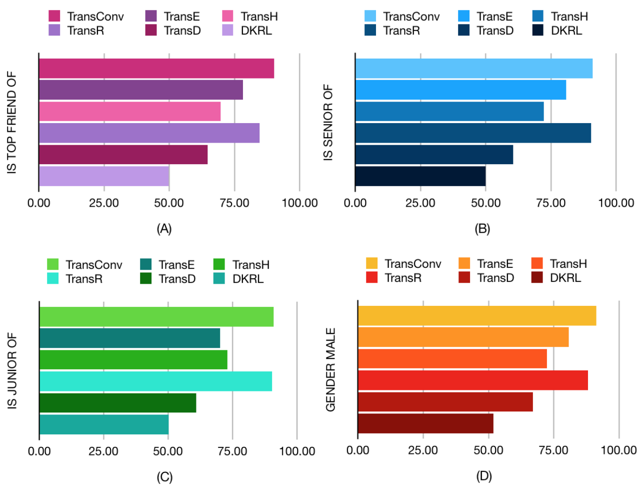

Figure 4: Results of triplet classification on different relationship categories of Facebook dataset. The relationship in (A) belongs 1-to-1, (B) belongs to 1-to-N, (C) belongs to N-to-1, and (D) belongs to N-to-N category.

in its own category. *TransConv* also outperforms other models in all four categories.

Multilabel Classification

In this section, we evaluate if the representations learned from *TransConv* is effective for multilabel classification on the relationship labels of user pairs. We use the same relationships selected in the experiments of triplet classification.

However, since the user representations learned from knowledge graph embedding models vary from different relationships, the common one-vs-all approach (Boutell et al. 2004) for multilabel classification is not applicable. We design a multilabel classification experiment based on global score threshold σ learned from the validation set. The experiment is constructed as below:

1. For each user pair (u_i, u_j) in the validation set, we retrieve the scores $f_r(u_i, u_j)$ on every relation r and further normalize the scores by z-score.
2. For each embedding model, we search a global score threshold σ among all relations and use it in prediction task. That is, if the normalized score of a user pair (u_i, u_j) in any relation r is smaller than σ, we predict (u_i, r, u_j) as a true triplet; otherwise, predict it as a negative one. Predicting a triplet as true means predicting the user pair hold the relationship label. It accordingly predicts each user pair a set of relationship labels.
3. We perform an exhaustive search for the global threshold σ that achieves the highest *hamming score* (Godbole and Sarawagi 2004) on validation set. Let T be the true set of labels and S be the predicted set of labels. Accuracy is measured by hamming score which symmetrically measures how close T is to S, defined as $Accuracy = |T \cap S|/|T \cup S|$.
4. We follow the same steps to get the normalized scores for every triplet in the testing set and predict the relationship labels by the global threshold σ. Then we report the hamming score as the testing accuracy of multilabel classification.

Model	Hamming Score - Accuracy (%)	
Dataset	Facebook	Twitter
TransE	12.4	37.7
TransH	13.8	37.8
TransR	14.0	38.1
TransD	10.8	37.4
DKRL(CBOW)+TransE	7.1	39.3
TransConv	**15.2**	**39.6**

Table 9: Results of multilabel 8-relationship classification.

We apply the same experiment procedure for all models and the results are shown in Table 9. *TransConv* performs best in 8-relationship classification task. It shows that taking our proposed conversation factors into consideration is effective to capture the strength of relationship.

Conclusion

In this paper, we proposed a novel relationship embedding model, *TransConv*, which is built upon structural translation on relationship hyperplane and further optimized through conversation factors established from textual communications. To the best of our knowledge, *TransConv* is the first model that considers both intensity and similarity of textual communications between users. Our experiments show that *TransConv* outperforms the state-of-the-art relationship embedding models in the tasks of social network completion, triplets classification and multilabel classification.

Acknowledgments

This research is supported by NSF and AFRL under contract numbers IIS-1546488, IIS-1618690, and FA8650-18-2-

7879. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern recognition* 37(9):1757–1771.

Cao, S.; Lu, W.; and Xu, Q. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 891–900. ACM.

Garcia-Duran, A.; Gonzalez, R.; Onoro-Rubio, D.; Niepert, M.; and Li, H. 2018. Transrev: Modeling reviews as translations from users to items. *arXiv preprint arXiv:1801.10095*.

Godbole, S., and Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, 22–30. Springer.

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

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, 687–696.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World Wide Web*, 591–600. ACM.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2181–2187.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710. ACM.

Salton, G. 1991. Developments in automatic text retrieval. *science* 253(5023):974–980.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077. ACM.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1225–1234. ACM.

Xiao, H.; Huang, M.; Meng, L.; and Zhu, X. 2017. Ssp: Semantic space projection for knowledge graph embedding with text descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3104–3110.

Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2659–2665.