

TransNet: Translation-Based Network Representation Learning for Social Relation Extraction

Cunchao Tu^{1,2}, Zhengyan Zhang¹, Zhiyuan Liu^{1,2*}, Maosong Sun^{1,2}

¹Department of Computer Science and Technology,

State Key Lab on Intelligent Technology and Systems,

National Lab for Information Science and Technology, Tsinghua University, China

²Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, China
{tcc13, zhangzhengyan14}@mails.tsinghua.edu.cn, liuzy@tsinghua.edu.cn, sms@mail.tsinghua.edu.cn

Abstract

Conventional network representation learning (NRL) models learn low-dimensional vertex representations by simply regarding each edge as a binary or continuous value. However, there exists rich semantic information on edges and the interactions between vertices usually preserve distinct meanings, which are largely neglected by most existing NRL models. In this work, we present a novel Translation-based NRL model, TransNet, by regarding the interactions between vertices as a translation operation. Moreover, we formalize the task of Social Relation Extraction (SRE) to evaluate the capability of NRL methods on modeling the relations between vertices. Experimental results on SRE demonstrate that TransNet significantly outperforms other baseline methods by 10% to 20% on *hits@1*. The source code and datasets can be obtained from <https://github.com/thunlp/TransNet>.

1 Introduction

How to represent networks is critical for network analysis, such as vertex classification [Lindamood *et al.*, 2009], clustering [Shepitsen *et al.*, 2008] and link prediction [Liben-Nowell and Kleinberg, 2007]. Typically, researchers employ one-hot representation and treat each vertex as a unique symbol. Such representation method is fairly simple. Nevertheless, it faces the sparsity issue, and deteriorates the performance of many machine learning algorithms when applied to network analysis tasks.

Inspired by recent trends of representation learning on image, speech and natural language, network representation learning (NRL) is proposed as an efficient technique to address the sparsity issue. NRL aims to encode the structure information of each vertex into a low-dimensional real-valued vector, which can be further utilized as features in various network analysis tasks.

There have been a large amount of NRL models proposed in recent years. These models tend to learn efficient vertex representations from local structure [Perozzi *et al.*, 2014;

Tang *et al.*, 2015] or global patterns [Cao *et al.*, 2015; Tu *et al.*, 2016a]. Besides, there are also some works proposed to incorporate heterogeneous information into NRL, such as text [Yang *et al.*, 2015] and labels [Tu *et al.*, 2016b; Li *et al.*, 2016].

What calls for special attention is that most existing NRL models neglect the semantic information of edges. As one of the essential network components, the edge is usually simplified as a binary or continuous value in conventional NRL models and most network analysis tasks. It is intuitive that such simplification can not model rich information of edges well. It is also well-acknowledged that the interactions between vertices in real-world networks exhibit rich and variant meanings. For example, the following behaviors to the same user in social media may be caused by different reasons; two authors have co-authored with another one in an academic network due to various common interests. Therefore, it is essential to integrate the detailed relation information of edges into NRL, which is expected to enable extracting latent relations between vertices in return.

In this work, we propose the task of Social Relation Extraction (SRE) to model and predict social relations for social networks. SRE is similar to the task of relation extraction in knowledge graphs (KG), for which the most widely-used methods are knowledge representation learning (KRL) such as TransE [Bordes *et al.*, 2013]. The difference is that there are usually no well pre-defined relation categories in SRE, and relations between vertices are typically hidden in their interactive text (such as co-authored papers between two researchers). It is intuitive that social relations can be represented by key phrases extracted from the interactive text, and there are usually multiple relational labels to indicate the complex relation between two vertices.

SRE cannot be well addressed by existing NRL and KRL methods. Conventional NRL models ignore the rich semantic information on edges when learning vertex representations, while typical KRL models such as TransE only perform well when the relation between two entities is specifically annotated with a single label. According to our statistics, only 18% entity pairs in *FB15k* (a typical KG) possess multiple relation labels, while the percentages of multi-label edges in SRE datasets are severalfold. To address this issue, we present a novel translation-based NRL model **TransNet** to incorporate multiple relational labels on edges into NRL.

*Corresponding Author: Zhiyuan Liu

“Translation” here means the movement that changes the position of a vector in representation space. Inspired by the successful utilization of translation analogy in word representation [Mikolov *et al.*, 2013] and KGs [Bordes *et al.*, 2013], we embed vertices and edges into the same semantic space and employ a translation mechanism to deal with the interactions among them, i.e., the representation of tail vertex should be close to the representation of head vertex plus the representation of edge. To handle the multi-label scenario, in TransNet we design an auto-encoder to learn edge representations. Moreover, the decoder part can be utilized to predict labels of unlabeled edges.

We construct three network datasets for SRE, in which edges are annotated with a set of labels. Experimental results show that TransNet achieves significant and consistent improvements comparing with typical NRL models and TransE. It demonstrates that our proposed TransNet is efficient and powerful on modeling relationships between vertices and edges.

To summarize, we make the following contributions:

- (1) We are the first to formalize social relation extraction (SRE) for evaluating the effectiveness of NRL models.
- (2) We propose a novel translation-based NRL model, TransNet, to incorporate the semantic information of edges. As a simple and straightforward model, it models the relationships among vertices and edges with translation mechanism.
- (3) We collect three annotated datasets for SRE, and experimental results on these datasets demonstrate the outstanding capabilities of TransNet to model and predict the relations between vertices.

2 Related Work

Recently, network representation learning (NRL) becomes an active research field that learns vertex representations from network structure. Various NRL models have been presented, which can be broadly divided into three categories. DeepWalk [Perozzi *et al.*, 2014], LINE [Tang *et al.*, 2015], node2vec [Grover and Leskovec, 2016] and SDNE [Wang *et al.*, 2016] tend to learn representations from local network structure. Besides, there are also some works that intend to capture the global structure and community patterns, such as GraRep [Cao *et al.*, 2015], CNRL [Tu *et al.*, 2016a] and MNMF [Wang *et al.*, 2017b]. Moreover, how to incorporate heterogeneous information into NRL is also critical. TADW [Yang *et al.*, 2015] introduces text information into NRL with matrix factorization. CANE [Tu *et al.*, 2017] learns context-aware network representation with mutual-attention mechanism. MMDW [Tu *et al.*, 2016b] and DDRW [Li *et al.*, 2016] incorporate labelling information to learn discriminative network representations. Besides, SiNE [Wang *et al.*, 2017a] learns vertex representations in signed networks, where each edge is either positive or negative. Nevertheless, such consideration of edges is over-simplified and not suitable to other types of networks.

To the best of our knowledge, less work has been done to consider the rich semantic of edges and make detailed predictions of relations on edges. It’s worth noting that relation extraction has been an essential task in knowledge

graphs [Mintz *et al.*, 2009; Riedel *et al.*, 2010; Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012; Lin *et al.*, 2016], which aims to extract relational facts to enrich existing KGs. It usually performs as relation classification, as there exist various large-scale KGs such as Freebase [Bollacker *et al.*, 2008] and DBpedia [Auer *et al.*, 2007], with labelled relations between entities. Nevertheless, there are usually no annotated explicit relations on edges in social networks, and it is also time-consuming to annotate edges in large-scale networks with human efforts. To address this issue, we propose to obtain the relations from interactive text information through NLP techniques automatically.

How to model the relationships between vertices and edges is crucial for predicting the relations precisely. In word representation learning field, [Mikolov *et al.*, 2013] found translation patterns such as “King”-“Man”=“Queen”-“Woman”. In knowledge graphs, [Bordes *et al.*, 2013] interprets the relations as translating operations between head and tail entities in the representation space, i.e., “head”+“relation”=“tail”. Inspired by these analogies, we assume there also exists translation mechanism in social networks, and propose translation-based NRL model, TransNet.

3 Social Relation Extraction

SRE is similar to the task of relation extraction (RE) in KGs, which is an important technique that aims to extract relational facts to enrich existing KGs. Knowledge representation learning (KRL) methods such as TransE [Bordes *et al.*, 2013], have been widely used for RE in KGs.

In this work, we present the task of Social Relation Extraction (SRE), which is designed to extract relations between social network vertices. Comparing with conventional RE in KGs, there are two obvious distinctions of SRE:

- (1) In KGs, relation categories are well pre-defined, and relational facts are annotated precisely with human efforts. Conversely, SRE is proposed to deal with a new scenario, in which relations between social network vertices are latent and typically hidden in their interactive text information.
- (2) In social networks, relations between vertices are extremely dynamic and complex, and cannot be portrayed well with a single label, because such practice cannot provide sufficient and accurate descriptions of these social relations. It is intuitive to represent the social relations by extracting key phrases from the interactive text information as a relation label set. These key phrases are flexible and capable of capturing the complex semantic information within social relations, as well as making these relations interpretable.

Formally, we define the problem of SRE as follows. Suppose there is a social network $G = (V, E)$, where V is the set of vertices, and $E \subseteq (V \times V)$ are edges between vertices. Besides, the edges in E are partially labeled, denoted as E_L . Without loss of generality, we define the relations between vertices as a set of labels, instead of a single label. Specifically, for each labelled edge $e \in E_L$, the label set of edge e is denoted as $l = \{t_1, t_2, \dots\}$, where each label $t \in l$ comes from a fixed label vocabulary T .

Finally, given the overall network and the labeled edges in E_L , SRE aims to predict the labels over unlabeled edges in

E_U , where $E_U = E - E_L$ represents the unlabeled edge set.

4 TransNet

In this work, we focus on the problem of incorporating rich relation information on edges into NRL.

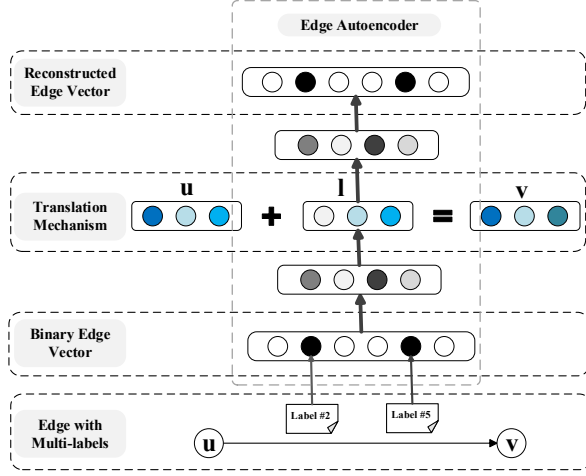


Figure 1: The framework of TransNet.

As shown in Figure 1, TransNet consists of two critical components, i.e., translation part and edge representation construction part. In the following parts, we first give the detailed introduction of translation mechanism in TransNet. Afterwards, we introduce how to construct the edge representations. At last, we give the overall objective function of TransNet.

4.1 Translation Mechanism

Motivated by translation mechanisms in word representations [Mikolov *et al.*, 2013] and knowledge representations [Bordes *et al.*, 2013], we assume that the interactions between vertices in social networks can also be portrayed as translations in the representation space.

Specifically, for each edge $e = (u, v)$ and its corresponding label set l , the representation of vertex v is expected to be close to the representation of vertex u plus the representation of edge e . As each vertex plays two roles in TransNet: head vertex and tail vertex, we introduce two vectors \mathbf{v} and \mathbf{v}' for each vertex v , corresponding to its head representation and tail representation. After that, the translation mechanism among u, v and e can be formalized as

$$\mathbf{u} + \mathbf{l} \approx \mathbf{v}'. \quad (1)$$

Note that, \mathbf{l} is the edge representation obtained from l , which will be introduced in details in section 4.2.

We employ a distance function $d(\mathbf{u} + \mathbf{l}, \mathbf{v}')$ to estimate the degree of (u, v, l) that matches the Eq. (1). In practice, we simply adopt L_1 -norm.

With the above definitions, for each (u, v, l) and its negative sample $(\hat{u}, \hat{v}, \hat{l})$, the translation part of TransNet aims to minimize the hinge-loss as follows:

$$\mathcal{L}_{trans} = \max(\gamma + d(\mathbf{u} + \mathbf{l}, \mathbf{v}') - d(\hat{\mathbf{u}} + \hat{\mathbf{l}}, \hat{\mathbf{v}}'), 0), \quad (2)$$

where $\gamma > 0$ is a margin hyper-parameter and $(\hat{u}, \hat{v}, \hat{l})$ is a negative sample from the negative sampling set N_e as

$$N_e = \{(\hat{u}, v, l) | (\hat{u}, v) \notin E\} \cup \{(u, \hat{v}, l) | (u, \hat{v}) \notin E\} \cup \{(u, v, \hat{l}) | \hat{l} \cap l = \emptyset\}. \quad (3)$$

In Eq. (3), the head vertex or tail vertex is randomly replaced by another disconnected vertex, and the label set is replaced by a non-overlapping label set.

The vertex representations in Eq. (2) are treated as parameters, and the edge representations are generated from the corresponding label set, which will be introduced in the following part.

4.2 Edge Representation Construction

As shown in Figure 1, we employ a deep autoencoder to construct the edge representations. The encoder part composes of several non-linear transformation layers to transform the label set into a low-dimensional representation space. Moreover, the reconstruction process of the decoder part makes the representation preserve all the label information. In the following sections, we introduce how to realize it in detail.

Input Mapping: We first map the label set to an input vector of autoencoder. Specifically, for a label set $l = \{t_1, t_2, \dots\}$ of edge e , we obtain a binary vector $\mathbf{s} = \{\mathbf{s}_i\}_{i=1}^{|T|}$, where $\mathbf{s}_i = 1$ if $t_i \in l$, and $\mathbf{s}_i = 0$ otherwise.

Non-linear Transformation: Taking the obtained binary vector \mathbf{s} as input, the encoder and decoder parts of the autoencoder consist of several non-linear transformation layers as follows:

$$\begin{aligned} \mathbf{h}^{(1)} &= f(\mathbf{W}^{(1)}\mathbf{s} + \mathbf{b}^{(1)}), \\ \mathbf{h}^{(i)} &= f(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}), i = 2, \dots, K. \end{aligned} \quad (4)$$

Here, K represents the number of layers and f denotes the activation function. $\mathbf{h}^{(i)}$, $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ represent the hidden vector, transformation matrix and bias vector in the i -th layer respectively. Specifically, we employ *tanh* activation function to get the edge representation $\mathbf{l} = \mathbf{h}^{(K/2)}$ as the vertex representations are real-valued, and *sigmoid* activation function to get the reconstructed output $\hat{\mathbf{s}}$ as the input vector \mathbf{s} is binary.

Reconstruction Loss: Autoencoder aims to minimize the distance between inputs and the reconstructed outputs. The reconstruction loss is shown as:

$$\mathcal{L}_{rec} = \|\mathbf{s} - \hat{\mathbf{s}}\|. \quad (5)$$

Here, we also adopt L_1 -norm to measure the reconstruction distance, the same as in Eq. (2).

However, due to the sparsity of the input vector, the number of zero elements in \mathbf{s} is much larger than that of non-zero elements. That means the autoencoder will tend to reconstruct the zero elements rather than non-zero ones, which is incompatible with our purpose. Therefore, we set different weights to different elements, and re-defined the loss function in Eq. (5) as follows:

$$\mathcal{L}_{ae} = \|(\mathbf{s} - \hat{\mathbf{s}}) \odot \mathbf{x}\|, \quad (6)$$

where \mathbf{x} is a weight vector and \odot means the Hadamard product. For $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{|T|}$, $\mathbf{x}_i = 1$ when $\mathbf{s}_i = 0$ and $\mathbf{x}_i = \beta > 1$ otherwise.

With the utilization of deep autoencoder, the edge representation not only remains the critical information of corresponding labels, but also has the ability of predicting the relation (labels) between two vertices.

4.3 Overall Architecture

To preserve the translation mechanism among vertex and edge representations, as well as the reconstruction ability of edge representations, we combine the objectives in Eqs. (2) and (6), and propose a unified NRL model TransNet. For each (u, v, l) and its negative sample $(\hat{u}, \hat{v}, \hat{l})$, TransNet jointly optimizes the objective as follows:

$$\mathcal{L} = \mathcal{L}_{trans} + \alpha[\mathcal{L}_{ae}(l) + \mathcal{L}_{ae}(\hat{l})] + \eta\mathcal{L}_{reg}. \quad (7)$$

Here, we introduce two hyper-parameters α and η to balance the weights of different parts. Besides, \mathcal{L}_{reg} is an $L2$ -norm regularizer to prevent overfitting, which is defined as

$$\mathcal{L}_{reg} = \sum_{i=1}^K (\|W^{(i)}\|_2^2 + \|b^{(i)}\|_2^2). \quad (8)$$

In order to prevent overfitting, we also employ dropout [Srivastava *et al.*, 2014] to generate the edge representations. At last, we adopt Adam algorithm [Kingma and Ba, 2015] to minimize the objective in Eq. (7).

4.4 Prediction

With the learnt vertex representations and the edge autoencoder, TransNet is capable of predicting the labels on the edges in E_U .

To be specific, given an unlabeled edge $(u, v) \in E_U$, TransNet assumes that the representations of u and v conform to the Eq. (1) with the potential edge representation. Thus, we can get the approximate edge representation through $\mathbf{l} = \mathbf{v}' - \mathbf{u}$. Naturally, we decode the edge representation \mathbf{l} with the decoder part in Eq. (4) to obtain the predicted label vector $\hat{\mathbf{s}}$. A larger weight \hat{s}_i indicates t_i is more possible in l .

5 Experiments

In order to investigate the effectiveness of TransNet on modeling relations between vertices, we compare the performance of our model to several baseline methods on social relation extraction (SRE) with three automatically constructed social network datasets.

5.1 Datasets

ArnetMiner¹ [Tang *et al.*, 2008] is an online academic website that provides search and mining services for researcher social networks. It releases a large-scale co-author network², which consists of 1,712,433 authors, 2,092,356 papers and 4,258,615 collaboration relations.

In this network, authors collaborate with different people on different topics, and the co-authored papers can reflect the detailed relations between them. Therefore, we construct the co-authored network with labeled edges in the following

steps. Firstly, we collect all the research interest phrases from the author profiles and build the label vocabulary with these phrases. These phrases are mainly crawled from the authors' personal home pages and annotated by themselves. Hence, these phrases are rather credible, which is also confirmed by our manual check. Secondly, for each co-author relationship, we filter out the in-vocabulary labels in the abstracts of co-authored papers and regard them as the ground truth labels of this edge. Note that, as the edges in co-author networks are undirected, we replace each edge with two directed edges with opposite directions.

Specifically, to better investigate the characteristics of different models, we construct three datasets with different scales, denoted as **Arnet-S**(small), **Arnet-M**(medium) and **Arnet-L**(large). The details are shown in Table 1.

Table 1: Datasets. (ML indicates multi-label edges.)

| Datasets | Arnet-S | Arnet-M | Arnet-L |
|-------------------|-----------|-----------|-----------|
| Vertices | 187,939 | 268,037 | 945,589 |
| Edges | 1,619,278 | 2,747,386 | 5,056,050 |
| Train | 1,579,278 | 2,147,386 | 3,856,050 |
| Test | 20,000 | 300,000 | 600,000 |
| Valid | 20,000 | 300,000 | 600,000 |
| Labels | 100 | 500 | 500 |
| ML Proportion (%) | 42.46 | 63.74 | 61.68 |

5.2 Baselines

We employ the following NRL models as baselines:

DeepWalk [Perozzi *et al.*, 2014] performs random walks over networks to generate random walk sequences. With these sequences, it employs Skip-Gram [Mikolov *et al.*, 2013], an efficient word representation model, to learn vertex representations.

LINE [Tang *et al.*, 2015] defines the first-order and second-order proximities of networks and optimizes the joint and conditional probabilities of edges in large-scale networks.

node2vec [Grover and Leskovec, 2016] extends DeepWalk with a biased random walk strategy. It can explore the neighborhood architecture more efficiently.

For these NRL models, we treat SRE as a multi-label classification task. Therefore, we concatenate the head and tail vertex representations to form the feature vector and adopt one-vs-rest logistic regression implemented by [Pedregosa *et al.*, 2011] to train a multi-label classifier.

Besides, we also compare our model with the typical knowledge embedding model, **TransE** [Bordes *et al.*, 2013]. For each training instance (u, v, l) , where $l = \{t_1, t_2, \dots\}$, it is intuitive to obtain several triples, i.e., (u, v, t_i) for each $t_i \in l$, which can be directly utilized to train TransE model. We adopt the similarity based predicting method as introduced in [Bordes *et al.*, 2013].

5.3 Evaluation Metrics and Parameter Settings

For a fair comparison, we evaluate the performance for each triple (u, v, t_i) as in TransE, where $t_i \in l$. Besides, we also employ *hits@k* and *MeanRank* [Bordes *et al.*, 2013] as evaluation metrics. Here, *MeanRank* is the *mean* of predicted ranks of all annotated labels, while *hits@k* means the proportion of

¹<https://cn.aminer.org/>

²<https://cn.aminer.org/aminetwork>

Table 2: SRE results on Arnet-S. ($\times 100$ for $hits@k$, $\alpha = 0.5$ and $\beta = 20$)

| Metric | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank |
|----------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|-------------|
| DeepWalk | 13.88 | 36.80 | 50.57 | 19.69 | 18.78 | 39.62 | 52.55 | 18.76 |
| LINE | 11.30 | 31.70 | 44.51 | 23.49 | 15.33 | 33.96 | 46.04 | 22.54 |
| node2vec | 13.63 | 36.60 | 50.27 | 19.87 | 18.38 | 39.41 | 52.22 | 18.92 |
| TransE | 39.16 | 78.48 | 88.54 | 5.39 | 57.48 | 84.06 | 90.60 | 4.44 |
| TransNet | 47.67 | 86.54 | 92.27 | 5.04 | 77.22 | 90.46 | 93.41 | 4.09 |

 Table 3: SRE results on Arnet-M. ($\times 100$ for $hits@k$, $\alpha = 0.5$ and $\beta = 50$)

| Metric | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| DeepWalk | 7.27 | 21.05 | 29.49 | 81.33 | 11.27 | 23.27 | 31.21 | 78.96 |
| LINE | 5.67 | 17.10 | 24.72 | 94.80 | 8.75 | 18.98 | 26.14 | 92.43 |
| node2vec | 7.29 | 21.12 | 29.63 | 80.80 | 11.34 | 23.44 | 31.29 | 78.43 |
| TransE | 19.14 | 49.16 | 62.45 | 25.52 | 31.55 | 55.87 | 66.83 | 23.15 |
| TransNet | 27.90 | 66.30 | 76.37 | 25.18 | 58.99 | 74.64 | 79.84 | 22.81 |

 Table 4: SRE results on Arnet-L. ($\times 100$ for $hits@k$, $\alpha = 0.5$ and $\beta = 50$)

| Metric | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| DeepWalk | 5.41 | 16.17 | 23.33 | 102.83 | 7.59 | 17.71 | 24.58 | 100.82 |
| LINE | 4.28 | 13.44 | 19.85 | 114.95 | 6.00 | 14.60 | 20.86 | 112.93 |
| node2vec | 5.39 | 16.23 | 23.47 | 102.01 | 7.53 | 17.76 | 24.71 | 100.00 |
| TransE | 15.38 | 41.87 | 55.54 | 32.65 | 23.24 | 47.07 | 59.33 | 30.64 |
| TransNet | 28.85 | 66.15 | 75.55 | 29.60 | 56.82 | 73.42 | 78.60 | 27.40 |

correct labels ranked in the top k . Note that, the above metrics will under-estimate the models that rank other correct labels in the same label set high. Hence, we can filter out these labels before ranking. We denote the primal evaluation setting as “Raw” and the latter one as “Filtered”.

We set the representation dimension to 100 for all models. In TransNet, we set the regularizer weight η to 0.001, the learning rate to 0.001 and the margin γ to 1. Besides, we employ a 2-layer autoencoder for all datasets and select best-performed hyper-parameters α and β on validation sets.

5.4 Results and Analysis

Table 2, Table 3 and Table 4 show the SRE evaluation results with different evaluation metrics on different datasets. In these tables, the left four metrics are raw results, and the right are filtered ones. From these tables, we have the following observations:

(1) Our proposed TransNet achieves consistent and significant improvements than all the baselines on all different datasets. More specifically, TransNet outperforms the best baseline, i.e., TransE, by around 10% to 20% absolutely. It demonstrates the effectiveness and robustness of TransNet on modeling and predicting relations between vertices.

(2) All NRL models perform poorly on SRE task under various situations, due to the neglect of rich semantic information over edges when learning vertex representations. On the contrary, both TransE and TransNet incorporate this information into the learned representations, thus achieve promising results on SRE. It indicates the importance of considering the detailed edge information, as well as the rationality of translation mechanism on modeling relations between vertices.

(3) Comparing with TransNet, TransE also performs poorly as it can only consider a single label on an edge each time, which turns the representation of labels on the same edge to be identical. Such practice may accord with the scenario in KG, where only 18% entity pairs possess multiple relation labels, according to our statistics on *FB15k* [Bordes *et al.*, 2013]. Conversely, the percentages of multi-label edges on SRE datasets are larger (42%, 64% and 62% on Arnet-S, Arnet-M, and Arnet-L respectively). Therefore, TransNet models all the labels over an edge simultaneously and can handle these issues well according to the results.

(4) TransNet has stable performance under different scales of networks. Moreover, when the number of labels turns larger, the performance of TransNet only has a small drop (from 90% to 80% on filtered $hits@10$), while NRL models and TransNet decrease more than 20%. This indicates the flexibility and stability of TransNet.

5.5 Comparison on Labels

To investigate the strengths of TransNet on modeling relations between vertices, we compare TransNet with TransE under high-frequency labels and low-frequency ones. In Table 5, we show the filtered $hits@k$ and *MeanRank* results on Arnet-S over top-5 labels and bottom-5 labels respectively.

From this table, we find that TransE performs much better on high-frequency labels than low-frequency labels since there is a liberal quantity of training instances for high-frequency labels. Compared with TransE, TransNet has stable and worthy performance on both types of labels. The reason is that TransNet employs an autoencoder to construct edge representations, which can take the correlations between

Table 5: Label comparisons on Arnet-S. ($\times 100$ for $hits@k$)

| Tags | Top 5 labels | | | | | Bottom 5 labels | | | |
|----------|--------------|--------------|--------------|-------------|--|-----------------|--------------|--------------|-------------|
| Metric | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank | | $hits@1$ | $hits@5$ | $hits@10$ | MeanRank |
| TransE | 58.82 | 85.68 | 91.61 | 3.70 | | 52.21 | 82.03 | 87.75 | 5.65 |
| TransNet | 77.26 | 90.35 | 93.53 | 3.89 | | 78.27 | 90.44 | 93.30 | 4.18 |

labels into consideration. These correlations can provide additional information for low-frequency labels, which will benefit the modeling and predicting of them.

5.6 Parameter Sensitivity

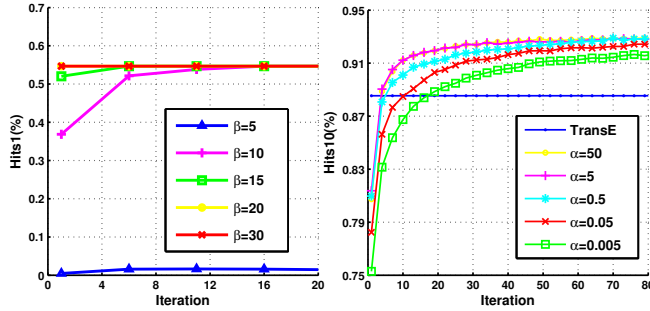


Figure 2: Parameter sensitivity.

There are two crucial hyper-parameters in TransNet, i.e., α and β . We investigate the parameter sensitivities on Arnet-S.

For β , it balances the weight of non-zero elements in auto-encoder. We determine the value of β according to the reconstruction performance when we initialize the auto-encoder using Eq. (6). In the left part of Figure 2, we show the raw $hits@1$ trends on validation set under different values of β . From this figure, we find that the reconstruction performance becomes stable after several iterations. More specifically, when β is too small (e.g., 5), the autoencoder will tend to reconstruct zero element rather than non-zero ones, which will lead to the poor reconstruction performance.

The hyperparameter α controls the weight of auto-encoder loss comparing with the translation loss. When the optimal β is determined, we show the raw $hits@10$ results in the right part of Figure 2 when α takes values under different orders of magnitude. From this figure, we observe that the performance of TransNet rises rapidly at the beginning, and then becomes stable. When α varies broadly, all the implements of TransNet can exceed TransE within 20 iterations and achieves stable performance at last. Therefore, TransNet can be easily implemented and well trained in practice due to its insensitivity of hyper-parameters.

5.7 Case Study

To demonstrate the effectiveness of TransNet, we provide a case in the test set of Arnet-S. The selected researcher is ‘‘A. Swami’’, and the recommended labels for its co-authors are shown in Table 6. In this table, labels in bold are correct ones. We observe that, both TransE and TransNet can recommend reasonable labels to different neighbors, which can

Table 6: Recommended top-3 labels for each neighbor.

| Neighbors | TransE | TransNet |
|----------------|--|--|
| Matthew Duggan | ad hoc network ; wireless sensor network; wireless sensor networks | management system ; ad hoc network ; wireless sensor |
| K. Pelechrinis | wireless network ; wireless networks; ad hoc network | wireless network ; wireless sensor network; routing protocol |
| Oleg Korobkin | wireless network ; wireless networks; wireless communication | resource management ; system design ; wireless network |

reflect their distinct co-author topics. However, for a specific neighbor, TransE only recommends similar labels due to its similarity-based recommendation method. Conversely, TransNet can recommend discriminative labels with the usage of decoder.

6 Conclusion

In this paper, we formalize the problem of social relation extraction (SRE) in social networks, which aims to extract detailed relations between vertices and can be used to evaluate the ability of NRL models on relation modeling. Moreover, we propose a novel translation-based NRL model, TransNet, to model interactions between vertices with translation mechanism. Experimental results on SRE show that TransNet can successfully integrate the rich semantic information preserved on edges into vertex representations and achieves significant improvements comparing with all baseline methods.

In future, we will explore the following directions:

(1) The effectiveness of TransNet has been verified on extracting explicit relations in homogeneous networks. In future, we want to explore how to model the relations between various types of vertices in heterogeneous networks.

(2) In SRE, the co-occurring vertex pairs and labels on edges can be regarded as a hyperedge in hypergraphs, where each edge contains three or more various vertices. Thus, we will explore network representation learning in hypergraphs, and the learned representations of vertices and labels are expected to benefit the SRE task.

Acknowledgements

This work is supported by the Major Project of the National Social Science Foundation of China (No.13&ZD190), the National Natural Science Foundation of China (NSFC No. 61572273, 61532010), and Tsinghua University Initiative Scientific Research Program (20151080406).

References

- [Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735, 2007.
- [Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, pages 1247–1250, 2008.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795, 2013.
- [Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of CIKM*, pages 891–900, 2015.
- [Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of KDD*, 2016.
- [Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL-HLT*, pages 541–550, 2011.
- [Kingma and Ba, 2015] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.
- [Li *et al.*, 2016] Juzheng Li, Jun Zhu, and Bo Zhang. Discriminative deep random walk for network classification. In *Proceedings of ACL*, 2016.
- [Liben-Nowell and Kleinberg, 2007] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.
- [Lin *et al.*, 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, pages 2124–2133, 2016.
- [Lindamood *et al.*, 2009] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of WWW*, pages 1145–1146, 2009.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICIR*, 2013.
- [Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of IJCNLP*, pages 1003–1011, 2009.
- [Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 12:2825–2830, 2011.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of KDD*, pages 701–710, 2014.
- [Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163, 2010.
- [Shepitsen *et al.*, 2008] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of RecSys*, pages 259–266, 2008.
- [Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.
- [Surdeanu *et al.*, 2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, pages 455–465, 2012.
- [Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of KDD*, pages 990–998, 2008.
- [Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of WWW*, pages 1067–1077, 2015.
- [Tu *et al.*, 2016a] Cunchao Tu, Hao Wang, Xiangkai Zeng, Zhiyuan Liu, and Maosong Sun. Community-enhanced network representation learning for network analysis. *arXiv preprint arXiv:1611.06645*, 2016.
- [Tu *et al.*, 2016b] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In *Proceedings of IJCAI*, 2016.
- [Tu *et al.*, 2017] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of ACL*, 2017.
- [Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of KDD*, 2016.
- [Wang *et al.*, 2017a] Suhan Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of SDM*, 2017.
- [Wang *et al.*, 2017b] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of AAAI*, 2017.
- [Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Proceedings of IJCAI*, 2015.