Structure Embedding for Knowledge Base Completion and Analytics

Zili Zhou*†, Guandong Xu*, Wenhao Zhu†, Jinyan Li*, Wu Zhang†
*Advanced Analytics Institute, University of Technology Sydney.
†School of Computer Engineering and Science, Shanghai University.
Email: Zili.Zhou@student.uts.edu.au, Guandong.Xu@uts.edu.au, whzhu@shu.edu.cn,
Jinyan.Li@uts.edu.au, wzhang@shu.edu.cn

***Abstract*—To explore the latent information of Human Knowledge, the analysis for Knowledge Bases (KBs) (e.g. WordNet, Freebase) is essential. Some previous KB element embedding frameworks are used for KBs structure analysis and completion. These embedding frameworks use low-dimensional vector space representation for large scale of entities and relations in KB. Based on that, the vector space representation of entities and relations which are not contained by KB can be measured. The embedding idea is reasonable, while the current embedding methods have some issues to get proper embeddings for KB elements. The embedding methods use entity-relation-entity triplet, contained by most of current KB, as training data to output the embedding representation of entities and relations. To measure the truth of one triplet (The knowledge represented by triplet is true or false), some current embedding methods such as Structured Embedding (SE) project entity vectors into subspace, the meaning of such subspace is not clear for knowledge reasoning. Some other embedding methods such as TransE use simple linear vector transform to represent relation (such as vector add or minus), which can't deal with the multiple relations match or multiple entities match problem. For example, there are multiple relations between two entities, or there are multiple entities have same relation with one entity.**

Insipred by previous KB element structured embedding methods, we propose a new method, Bipartite Graph Network Structured Embedding (BGNSE). BGNSE combines the current KB embedding methods with bipartite graph network model, which is widely used in many fields including image data compression, collaborative filtering. BGNSE embeds each entity-relation-entity KB triplet into a bipartite graph network structure model, represents each entity by one bipartite graph layer, represents relation by link weights matrix of bipartite graph network. Based on bipartite graph model, our proposed method has following advantages. BGNSE model uses one matrix for each relation, the relation transform between two entities can be done directly by forward and backward propagation of bipartite graph network, no need for subspace projection. Because of using bipartite graph network, the relation transforms between entities are nonlinear (network layer propagation), the multiple relations match or multiple entities match problems can be dealt. The learnt entity and relation embeddings can be used for problems such as knowledge base completions.

I. Introduction

Exploring the nature of knowledge is always an important task and an interesting challenge of AI area. Finding the useful knowledge representation is especially one of the most familiar concepts in AI [1]. Some recent developments build some large web-based Knowledge Bases (KBs), including some structured and multi-relational knowledge databases. Because of some Knowledge Base projects' long-term effort for knowledge data collection and management, a lot of useful KBs are available now, including Freebase, Wordnet, DBPedia, etc.

These KBs are valuable resources for learning latent information and computable representation of KBs. Most KBs consist of or contain great number of entity-relation-entity triplets, the elements of each triplet represent two entities and one relation between them. For example, in the triplet (”Abraham Lincoln”, ”Was a President of”, ”United States”), ”Abraham Lincoln” and ”United States” are entities, ”Was a President of” is the relation from ”Abraham Lincoln” to ”United States”. With all the entities as nodes and all the relations as edges, the triplets of KBs construct a huge knowledge graph. By mining the latent structure pattern in the knowledge graph, some useful representation of entities and relations can be discovered, such as knowledge base elements embedding.

Although some previous works offer good solutions for structured embedding, some issues exist. In these methods, Structured Embedding (SE) [2] and TranE [3] are both explainable and effective methods. Using entity-relation-entity triplets as input, these methods output the representations of entities and relations in different ways. The score function is designed to measure the existence of triplets based on such representations. In particular, TransE gives one vector for each entity and each relation, it uses simple vector transform, such as vector ”+/-” operation, to represent relation transforms. For example, it assumes that vector of ”Abraham Lincoln” add vector of ”Was a President of” is close to vector of ”United States”. The meaning of TransE model is clear, the structure of model is easy to understand, while the key limitation of TransE is that it can not handle the complex relation problems in KBs (e.g. the multiple relations match or multiple entities match). For Example, Abraham Lincoln was a president of United States, Abraham Lincoln was also a citizen of United States, this means the triplets (”Abraham Lincoln”, ”Was a President of”, ”United States”) and (”Abraham Lincoln”, ”Was a citizen of”, ”United States”) both exist. Another example is ”Robert Todd Lincoln” and ”Thomas 'Tad' Lincoln III” are both son of ”Abraham Lincoln”, the triplets (”Robert Todd Lincoln”, ”Is son of”, ”Abraham Lincoln”) and (”Thomas 'Tad' Lincoln III”, ”Is son of”, ”Abraham Lincoln”) both exist. In these examples, TransE model isn't able to give different representations for either different relations ("Was a President of", "Was a citizen of") or different entities ("Robert Todd Lincoln", "Thomas 'Tad' Lincoln III").

Differently, SE represent each entity with one vector. SE gives every relation two matrices, to project left and right entity vector into subspace vector respectively. SE assumes that the subspace projections of two entities are close to each other if the triplet is true, the projections are far from each other if the triplet is false. SE tries to project the entity into a common subspace by two direction of relation transform. This projection transform makes SE be able to handle some complex relation in KBs. The relation transform of SE is nonlinear, complex relations in KBs can be dealt, while the meaning of projection subspace is unclear.

Inspired by some graphical models [4], [5], especially Restricted Boltzmann Machine [6], we proposed using a graph-based model for translating embeddings for multi-relational knowledge, which is simple, universal, and efficient. In particualr, our model, namely Bipartite Graph Network Structure Embedding (BGNSE), uses one complete bipartite graph to represent each triplet in KB. In one triplet, each entity is mapped into one layer of bipartite graph, and links between layers represent relation. The forward and backward relation transforms are performed by forward and backward propagation of bipartite graph network, which share the same bipartite graph transition matrix. The Bipartite Graph Network can use only one matrix (link weight matrix of Bipartite Graph Network) for each relation representation instead of two matices used by SE. In BGNSE, the relation transform between entities are direct, the representation of an entity is transformed by relation matrix into another entity representation, no subspace projection is needed. On the other hand, TransE model is simpler and uses one vector for each relation transform, but it can not distinguish complex relation problems (e.g. the multiple relations match or multiple entities match). Because the BGNSE uses nonlinear relation transform, i.e., the network propagation, the complex relations in KBs can be accordingly handled and derived.

This paper is organized as follows. First, we introduce and summarize some related works. Then we introduce and explain the model framework of our method. The training steps and some training details of our method are also mentioned. The experiment results are listed and analyzed after that. Eventually, we conclude our method and discuss the extension and future.

II. Related Work

A. Structured Embedding (SE) and TransE

As mentioned above, SE and TransE model are both effective solution for KB embedding problem. Although there are some issues with these two models, the framework and score function are well designed.

SE [2] is a classic model for Multi-relational data embedding. Given a triplet (h,r,t) of KB, it use a vector to represent each entity, vector $\vec{h}$ for entity h, vector $\vec{t}$ for entity t. SE use two matrices for one relation r, R_{lhs} and R_{rhs}, to project entity h and t into a subspace. The score function used in the model is defined as

$$score(h,r,t) = \|R_{lhs}\vec{h} - R_{rhs}\vec{t}\|_1. \quad (1)$$

The model assumes that if the triplet (h,r,t) is true according to KB, the score function value of triplet, $score(h,r,t)$, should be low. The model offers a good framework for solving KBs Structure Embedding problem, while the meaning of projection subspace is unclear.

TransE [3] is another effective embedding model. With the similar framework, the author give a different definition of score function. Using vectors $\vec{h}$ and $\vec{t}$ for entities h and t, using another vector $\vec{r}$ for each relation r. Using the score function

$$score(h,r,t) = \|\vec{h} + \vec{r} - \vec{t}\|_2^2. \quad (2)$$

TransE uses simple and understandable relation transform, but has a new problem. TransE can't handle complex KB relation problems, such as multiple relations match or multiple entities match.

B. Other Models

Unstructured Model (UM) UM [7] is a model which is close to TransE, but with a simpler score function

$$score(h,r) = \|\vec{h} - \vec{r}\|_2^2. \quad (3)$$

The UM model can't identify the difference between relations, it can only output the relation of two entities is close or not.

TransH TransH [8] is also a simple and effective model, it addresses some issues of TransE. TransH uses a hyperplane w_r to represent relation r and vector h and vector t for entities h and t. h and t are projected to hyperplane w_r, $\vec{h_\perp}$ and $\vec{t_\perp}$. Then the score funtion is defined as

$$score(h,r,t) = \|\vec{h_\perp} + \vec{r} - \vec{t_\perp}\|_2^2. \quad (4)$$

The projection operations are defined as $\vec{h_\perp} = \vec{h} - w_r^T\vec{h}w_r$ and $\vec{t_\perp} = \vec{t} - w_r^T\vec{t}w_r$.

Neural Tensor Network (NTN) NTN [9] defines the score function as follows,

$$score(h,r,t) = \vec{u_r}^T tanh(\vec{h}^T R\vec{t} + R_1\vec{h} + R_2\vec{t} + \vec{b_r}), \quad (5)$$

where u_r is a relation-specific linear layer, R is a 3-way tensor, R_1 and R_2 are weight matrices for entities h and r. The model is considered too complex for applying in knowledge base efficiently. [10]

TransR and CTransR TransR and CTransR [10] is another approach attempting to solve the issues of above models. It assumes that the entities and relations are not in the same semantic space. TransR uses matrix M_r for projecting entities from entity space to relation space by $\vec{h_r} = \vec{h}M_r$ and $\vec{t_r} = \vec{t}M_r$, where $\vec{h}$ and $\vec{t}$ are entity embeddings. The score function is

$$score(h,r,t) = \|\vec{h_r} + \vec{r} - \vec{t_r}\|_2^2. \quad (6)$$

III. Our Method

To get an reasonable and computable representation of entities and relations in knowledge bases, we offer a structured embedding framework to define the abstract score function and learning framework for our embedding solution. We also propose a new method, Bipartite Graph Network Structured Embedding (BGNSE). BGNSE method embeds entities and relation of KB triplet into bipartite graph and uses propagation between bipartite graph network layers to represent the relation transform between entities.

A. Structured Embedding Framework

Because of the issues of other methods, we intend to build a structured embedding framework which includes following features.

Directly transform entities to each other. The score function of each triplet is used for determining the triplet is true or not. We use a score function inspired by score functions of previous works, but we do some modification to transform entities to each other directly based on the relation between them. For each triplet (e_h, r, e_t) in the knowledge base, the score function of a give entity-relation triplet is defined as:

$$S(e_h, r, e_t) = \|R(h) - t\|_p + \|R_{rev}(t) - h\|_p, \tag{7}$$

where h and t mean the d dimensional representation vectors of entities e_h and e_t. R is forward relation transform function of relation r, and R_{rev} is backward relation transfrom function of relation r. R projects left entity vector h to a predict right entity vector $R(h)$ which should be approximate to real right entity vector t, R_{rev} projects right entity vector t to a predict left entity vector $R_{rev}(t)$ which should be approximate to real left entity vector h. We use p-norm to determine the approximation between predict entity vectors and real entity vectors.

Forward and backward relation transforms should be related. The relations of KB are directed. For example, if A "is_part_of" B, B can not have "is_part_of" relation to A. But the two direction of relation transform should be related, the relation from B to A is obviously not "is_part_to", may be "contain" or "include", but it should be strongly related to "is_part_to" relation. In that case, the R transform function and R_{rev} reverse transform function in equation (7) should be extremely relevant, and R and R_{rev} can be tranformed into each other by specific operation. We introduce our method in section "Bipartite Graph Network Architecture" that using one Bipartite Graph Network to represent both R transform and R_{rev} transfrom, R transform and R_{rev} transform share the link weight matrix of same bipartite graph.

Can be used to represent the complex knowledge relations. Knowledge bases contain some complex relations, such as multiple relations match or multiple entities match. For example, there are more than one relation between two entities, or there are more than one entity have same relation to same entity. Some previous methods, such as TransE, have issues with complex knowledge relations. This is because TransE use simple entity vector transform. Our method can distinguish the difference of two relation between the same entity pair by using Bipartite Graph Network. The detail method introduction is in section "Bipartite Graph Network Architecture".

The results contain latent information of knowledge. Some latent information hidden in human knowledge base can be used to discover the nature of human knowledge. By structure embedding operations, the latent information can be extracted by deep analyzing the relation structure of big knowledge bases. The results of our method will extract the latent features of original knowledge base into a compressed low dimensional representation.

B. Bipartite Graph Network Architecture

Graphical Network has been widely used in many cases including some classic machine learning model. Bayesian Network [11] and Boltzmann Machine [12] are both widely used graphical models. Restricted Boltzmann Machine (RBM) [6] put units in two layers and only remain the links between layers, the units in same layer are unlinked. This method make the model effective and can be used in some application [13]. The success of RBM gives us hints to use bipartite graph network in our model.

Our new structured embedding method, Bipartite Graph Network Structured Embedding (BGNSE), uses entity-relation-entity triplet contained in knowledge bases as input, and it outputs the structured embeddings representation for entities and relation. The output represent a triplet by using one entity vectors for each entity and one matrix for relation in one triplet. BGNSE uses one complete Bipartite Graph Network to represent the transform operations of one triplet. Each entity vector is put into each layer of bipartite graph, use cells of vector for units of layer. The relation matrix is used as the weights of links between two layers of the bipartite graph. Then our structured embedding framework can use bipartite graph network architecture (shown in figure 1) to define the front and reverse relation transform functions used in score function, R and R_{rev} in equation (7).

Given triplet (e_h, r, e_t), using d-dimensional vector h to represent the entity vector of entity e_h, using d-dimensional vector t to represent the entity vector of entity e_t. The i_{th} cells of vector h is h_i, and the j_{th} cells of vector t is t_j. Put vector h into left layer of bipartite graph and put vector t into right layer of bipartite graph as figure 1. Using links between two layers to represent the relation between two entities, the link weight between h_i cells and t_j cells is $w_{i,j}$. Equation (8) and (9) can be gotten.

The values of right entity vector t, t_j $(j \in [1, d])$, are approximate to predict right entity vector values computed by weight sum of values in left entity vector h as figure 2, the weights are the bipartite graph link weights between cell t_j and all cells of vector h.

Reversely, the values of left entity vector h, h_i $(i \in [1, d])$, are approximate to predict left entity vector values computed by weight sum of values in right entity vector t as figure 3, the weights are the bipartite graph link weights between cell h_i and all cells of vector t.

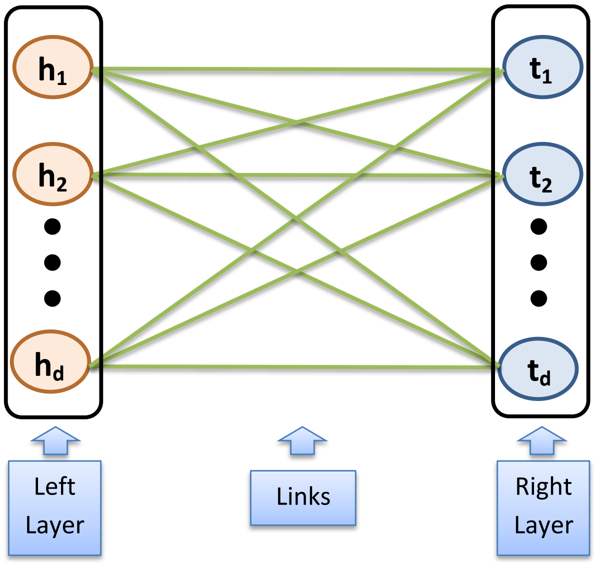

Fig. 1. Bipartite Graph Network Architecture

$$t_j \approx \sum_{i=1}^{d}(h_i \times w_{i,j}),\ j{\in}[1,d]. \quad (8)$$

$$h_i \approx \sum_{j=1}^{d}(t_j \times w_{i,j}),\ i{\in}[1,d]. \quad (9)$$

Obviously, relation r can be represented by link weight matrix of bipartite graph W (the link weight between h_i cell and t_j cell is $w_{i,j}$, the i_{th} row j_{th} column value of matrix W). The front transform function R and reverse transform function R_{rev} of relation r in equation (7) can be represented

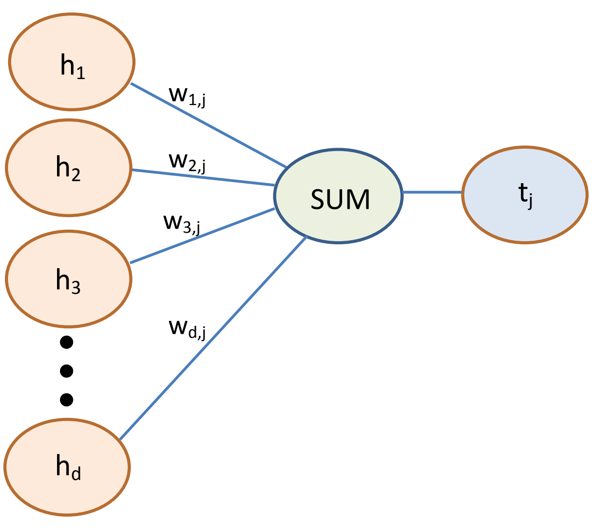

Fig. 2. The process of getting one unit value of predict right entity vector by using weight sum of all units in left entity vector, all the units in predict right entity vector are computed as the process described in figure.

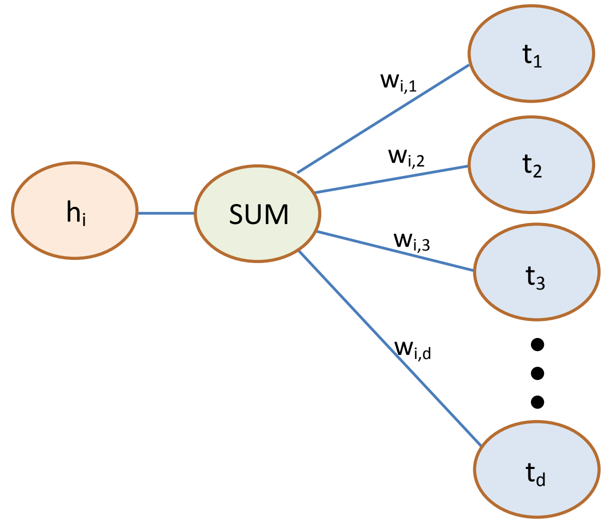

Fig. 3. The process of getting one unit value of predict left entity vector by using weight sum of all units in right entity vector, all the units in predict left entity vector are computed as the process described in figure.

by $R(h) = hW$ and $R_{rev}(t) = tW^T$. The equation (7) turns into following definition.

$$S(e_h, r, e_t) = \|hW - t\|_p + \|tW^T - h\|_p, \quad (10)$$

The front transform function R and reverse transform function R_{rev} are strongly related. R uses dot result of left entity vector h and bipartite graph link weight matrix W to get the predict right entity vector. R_{rev} uses dot result of right entity vector t and transpose of matrix W to get the predict left entity vector. Our method BGNSE uses only one bipartite graph link weight matrix to represent one relation. Because BGNSE use dot operation between entity vector and link weight matrix to get the predict result, it is obviously that BGNSE can solve the complex relation problems in KBs.

C. Training

In our model, the following parameters should be trained, entity vectors (h and t) and bipartite graph network link weight matrix (W). We choose $p = 1$ for our score function to simplify the training process. The final score function for training is defined as follows.

$$S(e_h, r, e_t) = \|hW - t\|_1 + \|tW^T - h\|_1, \quad (11)$$

Given the knowledge base triplet dataset X, true triplet $(e_h, r, e_t){\in}X$. We pick up false triplets as negative training sample, $(e_h, r, e){\notin}X$ and $(e, r, e_t){\notin}X$. The purpose of our training is to get the result that

$$S(e_h, r, e_t) < S(e_h, r, e), \forall : (e_h, r, e){\notin}X \quad (12)$$

and

$$S(e_h, r, e_t) < S(e, r, e_t), \forall : (e, r, e_t){\notin}X. \quad (13)$$

We want the score function result of true triplet is smaller than false triplet, as shown in equation (12) and (13).

We use stochastic gradient descent (SGD) for the parameters training of our method. We build an objective function

$$O = \begin{cases} m + S - S_{neg} + P & \text{if } m + S - S_{neg} \geq 0; \\ 0 & \text{if } m + S - S_{neg} < 0, \end{cases} \tag{14}$$

where S means the score function of triplet in KB (true triplet), S_{neg} means the score function of negative triplet not in KB (false triplet). We use $P = \lambda \|W\|_1$ in objective function as sparse penalty function to keep the relation transform matrix W sparse and avoid the model from overfit. We use L1 regularization penalty function here, the same penalty function is also used in Sparse Coding method [14]. The detail of using L1 regularization can be found in [15]. $max(a, b)$ function gets the max value from a and b. We add a limitation to objective function that no optimization is applied if $m + S - S_{neg} < 0$. $m + S - S_{neg} < 0$ means that the margin between the score function results of true triplet and false triplet is large than m. If the optimization is still applied when $m + S - S_{neg} < 0$, the objective function value of some pairs of true and false triplet will be overfit, but the others won't fit well. In this case, we add limitation for training to minimize the objective function for most number of training set. Parameter m is the margin between score function results of true and false triplet. m can be set depends on the detail need of result. The larger m is set, the difference between score function results of true and false triplet is more evident. While the value of m should be tuned carefully, too large value for m can lead to local optimization problem (only results for several specific triplets are correct, others are incorrect), the global effectiveness of training results can be comprimised.

By minimize the objective function, the structured embeddings can be gotten. The detail train algorithm of our model is defined as in algorithm 1.

Algorithm 1 SGD for BGNSE

```
for Fix number of iterations do
  Select a triplet (e_h, r, e_t) from X at random.
  Select an entity e from entities at random.
  if (e_h, r, e) ∉ KB triplets then
    if m + S(e_h, r, e_t) − S(e_h, r, e) ≥ 0 then
      minimize{m + S(e_h, r, e_t) − S(e_h, r, e) + P}.
    end if
  end if
  if (e, r, e_t) ∉ KB triplets then
    if m + S(e_h, r, e_t) − S(e, r, e_t) ≥ 0 then
      minimize{0, m + S(e_h, r, e_t) − S(e, r, e_t) + P}.
    end if
  end if
end for
```

We also make a constriant after each iteration, the L2-Regularization of each entity vector should be regularized to 1, to avoid vectors of entities become to close to zero, the zero-close value entity vectors can also lead to a small result value for objective function.

D. Latent Information

The meaning of using Bipartite Graph Network for structured embedding is not only get a representation embedding for each relation. The process of training Bipartite Graph Network also allows our model to learning some abstract latent information from the structure of whole knowledge base. The biapartite graph network link weight matrix representing relation transform have to be trained for fitting all the triplets containing this relation, abstract features of relation are extracted into relation biapartite graph network link weight matrix during the training process.

If we use simple sparse representation for knowledge entities and relations embedding, such as one-hot vectors for entities and sparse matrices for relations, the result can represent the relations already contained in KB, but the latent information of knowledge can not be represented. The hidden or missed relation of knowledge base can not be found by sparse representation, because the pattern of relations are not learnt.

The original KB stores the entities and the relations between them with sparse representation, high dimensional sparse vectors and matrices will be used if we represent KB entities and relaitons directly. The structure embedding methods embedding the KB data into an low dimensional compressed representation containing latent information for knowledge completion and analytics such as missed knowledge relation found and relation similarity prediction. BGNSE takes the advantage of Bipartite Graph Network, the result of BGNSE is a highly compressed and abstract representation of knowledge, the latent information contained by BGNSE result can be used for knowledge nature discovery.

The relation matrices used in BGNSE model can fit the dataset during the training. The relation matrices are low dimensional compressed matrices. The result relation matrices have to fit most of data in KB, the result relation matrices has to contain the abstract pattern of knowledge relation. The accuracy of knowledge abstract representation will increase by using larger size of KB datasets.

IV. Experiments and Analysis

A. Data Sets

We choose a famous knowledge base dataset for our experiments, Freebase. This dataset is widely used in previous works. We test our methods on few subsets of FB15k dataset [16]. FB15k is a large scale dataset. FB15k contains 14,951 entities, 1,345 relations and 483,142 triplets in training dataset.

We get the top 200 and top 500 entities from FB15k dataset by entity frequency. We extracted triplets, which contains these top frequency ranked entities, from FB15k to build datasets for our experiment, FB200 and FB500. FB200 consists of triplets containing the FB15k top 200 entities, and FB500 consists of the triplets containing the FB15k top 500 entities. The details of FB200 and FB500 are contained in Table 1.

TABLE I
SIZES OF EXTRACTED DATASETS

	entities	relations	train data triplets
FB200	146	121	2,238
FB500	485	314	11,649

TABLE II
RANKING RESULTS ON FB200 (146 ENTITIES)

	train data (left / right)	test data (left / right)
TransE	16.00 / 24.45	21.07 / 30.80
SE	15.55 / 18.26	21.69 / 23.85
BGNSE	**10.60 / 8.18**	**13.32 / 16.31**

B. Link Prediction and Ranking Evaluation

The link prediction is an evaluation method for multi-relational data structured embedding results. This evaluation method has been used in some previous works [2], [3], [10]. Link prediction aims to find the missing left or right entity of a given triplet. Link prediction can solve the problems as "What is the job title of Alan Turing". The triplet (Alan Turing, Job title, ?) misses its right entity, which might be found out by Structured Embedding methods.

We utilize link prediction to evaluate our model in translating embeddings in multi-relational KBs, which aims to find the missing left or right entity of a given triplet.

Given test triplet (h,r,t), we replace the left/right entity by all entities in entity list. We get the rank of entites by descending order of scores given by equation (7). We use the average rank of true entities, which appears in test triplets, to measure the effectiveness of our method. The smaller the rank, the better performance is.

C. Experiment Result

TransE and SE are two classic multi-relational embedding methods. The model of TransE method is understandable, the efficiency of TransE is high enough for most of embedding problems. SE methods can handle complex relation in relational triplet sets, while TransE can not handle complex relation problem. The datasets we used in our experiments contain high proportion of complex relations. We use TransE, SE and our method to compare the differences between our method and two type of classic embedding methods.

For these reasons, we conduct experiments using TransE and SE as baselines on two datasets extracted from Freebase dataset. The results of experiment are tubalated in Table 1 and 2, respectively. Because the complicated nature of relations contained in our datasets, the TransE perfoms worse due to its intrinsic weakness as described above. SE and BGNSE perform closely, the average rank result of BGNSE is slight better than SE.

V. CONCLUSION AND FUTURE WORK

We proposed BGNSE, an approach for multi-relational data structured embedding. In our method, due to the advantage of Bipartite Graph Network, our method only needs to learn one matrix to model structured embeddings and handle the complex KB relations without the need of subspace projection. The forward transform and backward transforms of one relation are strongly related. Forward and backward transform matrix can turns into each other by transpose of itself, because the relation transform is represented by bipartite graph network link weight matrix. The experiments have justfied the effectiveness of our approach.

TABLE III
RANKING RESULTS ON FB500 (485 ENTITIES)

	train data (left / right)	test data (left / right)
TransE	86.20 / 104.46	94.23 / 111.13
SE	29.73 / 26.76	35.68 / 34.99
BGNSE	**29.61 / 25.07**	**32.68 / 33.31**

Furthermore, we believe this is a knowledge relation representation method which is more natural and reasonable, because relation transform matrix can learn abstract latent informations of relation during the learning steps of bipartite graph network model.

The future works include how to extend better score function to improve the embedding effectiveness and how to extract and use the latent information derived from relation matrix obtained by BGNSE. Firstly, using the Bipartite Graph Network is not the only choice for building backward relation transform, R_{rev} function used in equation (7). In that case, we give the abstract score function of our method in equation (7), the R_{rev} function could be replaced by other reverse relation transform which is related to R function based on other theories. Furthermore, the learning process of BGNSE let the result entity vectors and relation matrices contain latent information of whole KB. We believe some latent information can be extracted from the embedding result of BGNSE when we use bigger KB datasets.

VI. ACKNOWLEDGEMENT

The authors thank the reviewers for their helpful comments. This work was partially supported by the Major Research Plan of National Science Fundation of China [No. 91630206].

REFERENCES

[1] R. Davis, H. Shrobe, and P. Szolovits, "What is a knowledge representation?" *AI magazine*, vol. 14, no. 1, p. 17, 1993.

[2] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Conference on Artificial Intelligence*, no. EPFL-CONF-192344, 2011.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2787–2795.

[4] G. E. Hinton and T. J. Sejnowski, "Learning and releaming in boltzmann machines," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, pp. 282–317, 1986.

[5] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. American Mathematical Society,, 1980.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing." in *AISTATS*, vol. 351, 2012, pp. 423–424.

[8] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes." in *AAAI*. Citeseer, 2014, pp. 1112–1119.

[9] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in Neural Information Processing Systems*, 2013, pp. 926–934.

[10] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion." in *AAAI*, 2015, pp. 2181–2187.

[11] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[12] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[13] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.

[14] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[15] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[16] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.