2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Enhancing Knowledge Graph Completion with Positive Unlabeled Learning

Jinghao Niu, Zhengya Sun, Wensheng Zhang
Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences, Beijing, China.
Email: niujinghao2015@ia.ac.cn, zhengya.sun@ia.ac.cn, zhangwenshengia@hotmail.com

***Abstract*—Knowledge graphs have proven to be incredibly useful for many artificial intelligence applications. Although typical knowledge graphs may contain a huge amount of facts, they are far from being complete, which motivates an increasing research interest in learning statistical models for knowledge graph completion. Learning such models relies on sampling appropriate number of negative examples, as only the positive examples are contained in the data set. However, this would introduce errors or heuristic biases which restrict the sampler to visit other potentially reliable negative examples for better prediction models. In this paper, we present a novel perspective on skillfully selecting the negative examples for knowledge graph completion. We develop a two-stage logistic regression filter under the positive-unlabeled learning (PU learning) framework, which enables an automatic and iterative refinement of the negative candidate pools. We then contrast positive examples with the resulting negative ones based on the improved embedding-based models. In particular, we work with a cost-sensitive loss function by weighting the semantic differences between negative examples and particular positive ones. This weighting scheme reflects the importance of predicting the preferences between them correctly. In experiments, we validate the effectiveness of negative examples in refining and weighting schemes, respectively. Besides this, our proposed prediction model also outperforms the state-of-the-art methods on two public datasets.**

I. Introduction

Knowledge graphs (KGs) have played an essential role in a variety of applications such as question answering and text analysis. They provide information about a variety of relational facts represented in the form of triples (h, r, t), where h denotes a head entity, and r is a relation that connects h to a tail entity t. Although typical KGs, such as Freebase [1], Yago [2], and Knowledge Vault [3] have reached an impressive size, they are far from being complete. This has motivated a tremendous research interest in knowledge graph completion, i.e., automatically discovering new facts between entities in the KG.

The approaches of knowledge graph completion, which have been recently developed and validated, can be divided into two categories: latent feature models and observed feature models. The methods in the first major category attempt at embedding entities and relations into a low-dimensional continuous vector space, with each triple represented as a score function in the vector space [4]–[6]. They typically formulate the loss functions that preserve certain preferences between positive and negative triples. The second category of methods, however, directly construct binary features for candidate triples based on a simplified variant of path ranking algorithm (PRA) [7]. These features together with their weights are used to define the score of a triple, which is helpful in capturing correlations among multiple relation types [8].

Learning the models discussed so far relies on sampling appropriate number of negative examples (unknown triples), as only the positive examples (known triples) are contained in the data set. For example, early methods obtain negative candidates by replacing the head entity or the tail entity randomly (e.g., TransE [6]), which can easily lead to false negatives. Subsequently, various specific heuristics are incorporated into the random sampling process with an improved quality [8]–[10]. However, this would introduce biases which restrict the sampler to visit other potentially reliable negative examples for better prediction models.

In this paper, we propose a **t**wo **s**tage **l**ogistic **r**egression **f**ilter (TSLRF), i.e. a novel negative example generation approach based on the positive-unlabeled learning framework. Specifically, it extracts a set of reliable negative examples from the initial unlabeled data, which together with the available positive examples, are then used to train a binary classifier. It performs in an iterative manner and outputs the set of the low scoring negative candidates for the downstream training. We further devise a novel embedding-based model which works with cost-sensitive losses, by weighting the semantic differences between negative examples and particular positive ones. This weighting scheme reflects the importance of predicting the preferences between them correctly. Experimental results manifest that reliable negative examples benefit both the latent and the observed feature models, and lead to further performance improvement when combined with the weighting scheme in the latent feature models.

Our main contributions include the followings:

- The proposed TSLRF adopts PU learning instead of purely random sampling or heuristic search; the output set is iteratively refined to find reliable negative examples in a data-driven manner.
- The introduced weighting scheme between positive-negative example pairs is generic for the embedding-based models, since the negative candidates do not have the equal semantic distance to any particular positive one.
- Empirical evidence on benchmark collections confirm the advantage of our approach for discovering new facts in comparison with several baseline methods.

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

II. Related Works

Early rule-based methods for the knowledge graph completion have been proved to be effective, such as first-order logic approaches [11] and probabilistic soft logic approaches [12]. However, these methods perform poorly at large KGs because of the convergence problem. Recently, the statistical methods based on the latent and observed features are proposed, which could effectively alleviate this issue.

Methods based on the latent features usually embed each item (an entity or a relation) into a flexible continuous vector space. The Unstructured Model (UM) [13] only uses the entity item embedding, not considering relation-related information. The Structured Embedding (SE) model [4] learns two matrices for head-specific relations and tail-specific relations, respectively. Semantic Matching Energy (SME) model [14] further captures the correlations between entities and relations by matrix operations. Other models, such as Latent Factor Model (LFM) [15], RESCAL [16], and the Neural Tensor Network (NTN) [5] employ tensor-like structures to detect such correlations. A recent, state-of-the-art model is TransE [6], which considers the relation as a translation from head entity to tail entity in the embedding space and achieves a remarkable performance with reduced parameters. However, TransE has limitations in modeling one-to-many, many-to-one, and many-to-many relations. To alleviate this issue, TransH [9], TransR/CTransR [17], TransD [18] were put forward successively, but lose the simplicity and efficiency of TransE. As a remedy, HOLE [19] introduces circular correlation to create compositional representations, which offers a better representation capacity with the equivalent amount of parameters of TransE. Note that most current embedding models work by minimizing a margin-based pairwise ranking loss function. Yet, the semantic differences between negative examples and particular positive ones are typically ignored.

Instead of using latent features, several recent studies proposed to train the prediction models based on the observed features (e.g., the graph structure similarity and direct links between entity pairs), with surprising results [8], [20], [21]. A common characteristic of the latent and observed feature models is that they all rely on sampling appropriate negative examples, as only the positive examples are contained in the data set. However, a major limitation is that the involved negative example generators would introduce the errors or heuristic biases which restrict the sampler to visit other potentially reliable negative examples.

III. Preliminaries

A. PU-learning

Learning from Positive and Unlabeled Examples (PU-learning) is an approach for two-class semi-supervised classification problem, which utilizes labeled positive examples and unlabeled examples [22]. The labeled positive examples are scarce for PU-learning. But unlabeled data (e.g., raw Internet data) are convenient and cheap to get, which can be used to reduce the workload of labeling data and improve the classification performance for many applications. For example, Xia et al. [23] use PU-learning framework for instance selection and weighting, improving the performance of cross-domain sentiment classification. Liu et al. [24] use PU-learning method to improve and facilitate the drug discovery process. PU-learning has two main processes:

- Generating a reliable negative example set according to labeled positive examples.
- Training a binary classifier with the labeled positive examples and reliable negative examples.

For the knowledge graph completion task, labeled examples (known triples) are far less than unlabeled examples (unknown triples), which particularly fits the setting of PU-learning.

B. Generating Reliable Negative Examples

Generally speaking, there are four types of negative example generating strategies:

- Sampling negative examples randomly (e.g., TransE [6]).
- Replacing the h and t entities with different probabilities, which depend on the types (mapping properties) of relations [9].
- Choosing the h and t entities according to their entity types. [8], [10].
- Extracting candidate negative examples that are relatively close to the positive examples in the embedding vector space [25].

Sampling approaches are nearly inevitable to choose some false negative examples. The above methods, except for the random sampling, are proposed to reduce the chance of generating false negative examples or find more significant negative examples. However, they highly depend on the heuristic knowledge (e.g., the predefined rule [8]–[10]) or the high-quality additional resource (e.g., pre-trained embeddings [25]). In this study, we propose a data-driven approach to improve the quality of the negative candidate pool iteratively. Our method only requires the observed information of the given KGs.

C. Latent and observed features of triples

In latent feature models, each triple is represented as a score function or certain combination operator that depends only on learned embedding vectors of the entities and relations, and possibly additional global parameters. In this work, we use circular correlation as the compositional operator introduced by Nickel et al. [19]. It can capture rich interactions but simultaneously remain efficient to train. In contrast, observed feature models directly construct interpretable features for each triple, which together with their weights, are used to define the score of a triple. An early observed feature model [7] exploit Path Ranking Algorithm (PRA) to get appropriate weights for different paths, which are then used for the link prediction. In this work, we extract six types of observed features (Eq. (1)-Eq. (6)) for every candidate triple (e_i, r_k, e_j). To be specific, we employ four types of observed features (feature 1-4) introduced by Toutanova et al. [8] and further define another two types of new observed features (feature 5 and 6).

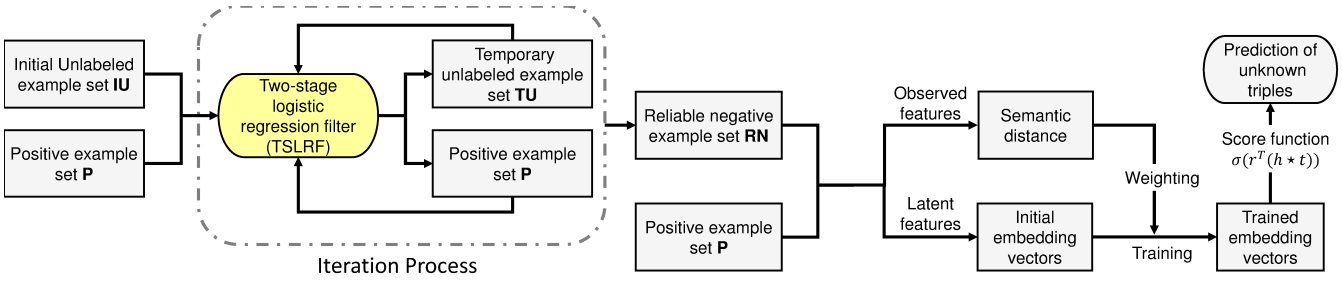

Fig. 1. The overall architecture of our proposed framework. First the Two-Stage Logistic Regression Filter (TSLRF) is used in an iterative manner to generate the reliable negative example set. Then a semantically weighted embedding-based model learns to generate the prediction of the unknown triples, using reliable negative examples and positive examples.

$$\mathbf{1}(r'\&r_k) = \begin{cases} 1 & (r' \neq r_k)(e_i, r', e_j) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (1)$$

$$\mathbf{1}(r'_{inv}\&r_k) = \begin{cases} 1 & (e_j, r'_{inv}, e_i) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (2)$$

$$\mathbf{1}(e_i = s\&r_k) = \begin{cases} 1 & (e_i, r_k, e) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (3)$$

$$\mathbf{1}(e_j = o\&r_k) = \begin{cases} 1 & (e, r_k, e_j) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (4)$$

$$\mathbf{1}(e_i = s) = \begin{cases} 1 & (e_i, r, e) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (5)$$

$$\mathbf{1}(e_j = o) = \begin{cases} 1 & (e, r, e_j) \in TrainingSet \\ 0 & otherwise \end{cases} \quad (6)$$

We use e, r to denote any possible entity and relation, and r' to denote the relation that is different from r_k. For feature 3-6, s and o indicate the subject and object positions, respectively. Generally speaking, feature 1, 2 are used to describe the correlation distribution of two entities in the candidate triple. Given a pair of head and tail entities, whether they appear together in a known relation r' (or the inverse relation r'_{inv}) triple provides useful information for predicting the candidate triple. For example, one person (e_i) living in (r') a certain city (e_j) might also works (r_k) in the same city (e_j). Feature 3, 4 are about the distribution of entities in the relationship domain. They capture a bias of the entity occurring in head or tail position for the specific candidate relation r_k. An entity might intuitively occur in several particular types of relations. For example, $UnitedState$ is a common nationality entity for the nationality relation in Freebase [8].

As a natural extension, feature 5 and 6 that we introduce in this paper characterize the context of entities in the tuple domain. The two features contain the information of entities occurring at specific head or tail positions in a triple (given the relation and another entity of a particular triple). An entity might be closely related to several particular tuples of relations and entities. For example, broccoli, soybean, and milk are common food entities for the tuple (*contains, calcium*) in the KGs extracted from the Web.

IV. The Proposed Method

In this study, we proposed a triple prediction framework that can effectively exploit both the observed features and the latent features. We propose TSLRF to obtain sound negative examples iteratively and automatically. Then an embedding-based prediction model is trained by minimizing a properly weighted margin-based loss function.

A. Generating Reliable Negative Examples with TSLRF

As fig. 1 shows, our proposed prediction framework first generates the initial unlabeled example set IU and positive example set P depending on known triples in the KGs. For every positive example (h, r, t), we generate two types of unlabeled examples (h', r, t) and (h, r, t') (unknown triples in the training set) by random sampling under *local closed world assumption*. Every triple will correspondingly generate 10 examples for both types, which make up the initial unlabeled example set. The initial example set IU and P are then fed into TSLRF.

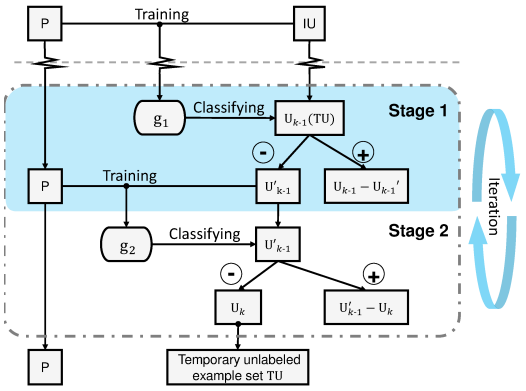

Fig. 2. Architecture of Two-Stage Logistic Regression Filter (TSLRF).

To adjust the imbalance between positive and negative examples, we design a cost-sensitive logistic regression loss function for TSLRF. Parameter α>1 is used to control the importance ratio of positive and negative examples. The loss function is:

$$L_s = -\sum_{i=1}^{(N^+ + N^-)} \{\alpha \cdot y_i \cdot ln(\sigma(\boldsymbol{w}\mathbf{x}^T + w_0)) + (1 - y_i) \cdot ln(1 - \sigma(\boldsymbol{w}\mathbf{x}^T + w_0))\} + \beta \cdot \|\boldsymbol{w}\|_1, \quad (7)$$

where β is the regularization parameter; σ denotes the logistic function; N^+ *and* N^- denote the number of positive and negative examples respectively; M is the number of observed features.

Algorithm 1 Two-Stage Logistic Regression Filter

Input: Positive example set P, initial unlabeled example set IU, the maximum number of iterations n_m, the convergence ratio r_c.

Initialize: $k \leftarrow 0$, $U_0 \leftarrow IU$, $TU \leftarrow U_0$;
repeat
  $k \leftarrow k + 1$;
  // The first stage
  Learn a logistic regression classifier g_1 from P and U_{k-1}, using Eq. (7);
  Classify U_{k-1} using g_1. Let the set of triples in U_{k-1} that are classified as negative be U'_{k-1};
  // The second stage
  Learn a logistic regression classifier g_2 from P and U'_{k-1}, using Eq. (7);
  Classify U'_{k-1} using g_2. Let the set of triples in U'_{k-1} that are classified as negative be U_k;
  $r \leftarrow \frac{|U_k|}{|TU|}$;
  $TU \leftarrow U_k$;
until $r > r_c$ or $k > n_m$
$RN \leftarrow U_k$;

Output: the reliable negative example set RN.

We use TSLRF in an iterative manner until the convergence of the unlabeled example set or achieving the max iteration number. The overall algorithm is schematically shown in fig. 2, and a pseudo-code is provided in Algorithm 1.

In the first stage, the input unlabeled set of the $(k-1)^{th}$ iteration U_{k-1} and the labeled positive set P are used to learn a logistic regression classifier g_1. The model parameters are tuned based on the validation set containing only labeled positive examples [26] (the evaluation metric is $recall^2/P[f(\boldsymbol{X}) = 1]$, where $P[f(\boldsymbol{X}) = 1]$ is the probability that a sample is judged to be positive). g_1 is then used to classify U_{k-1} and get a better unlabeled example set U'_{k-1}.

In the second stage, the logistic regression model is trained based on P and U'_{k-1}. The model parameters are tuned according to the MRR (mean reciprocal rank) metric on the validation set. Afterwards, the model calculates the scores of $\sigma(\boldsymbol{w}\mathbf{x}^T + w_0)$ for every unobserved triple and regards the triples that are assigned with the scores below a certain threshold as the effective negative examples for downstream training. They will make up the next iteration's unlabeled example set U_k. Then U_k will be used to compare with the temporary unlabeled example set TU. If the model does not converge, U_k will be used as the input of the next iteration. After several numbers of iterations, the reliable negative example set RN will be generated and used to train the embedding-based prediction model in our proposed framework.

Note that besides logistic regression, we can also use other classification modes in our proposed architecture, such as SVM, decision trees, and Naive Bayes. However, logistic regression utilizes the logistic transform to relate the predictor and the approximate conditional probability it represents, which retains good probabilistic interpretability for the negative example selection.

B. The Semantically Weighted Prediction Model

Equipped with *RN* and *P*, we devise a properly weighted margin-based loss function for the latent feature models (see fig. 1). To score possible triples, we take circular correlation introduced by Nickel et al. [19] as our choice. Let **h**, **t**, and **r** denote the embedding vectors of the head entity, tail entity and relation, the probability of a triple can be modeled as:

$$f_r(\mathbf{h}, \mathbf{t}) = \sigma(\mathbf{r}^T(\mathbf{h} \star \mathbf{t})) \tag{8}$$

where σ denotes the logistic function and $\star$ denotes *circular correlation*:

$$[\boldsymbol{h} \star \boldsymbol{t}]_k = \sum_{i=0}^{d-1} h_i t_{(k+i) mod\ d} \tag{9}$$

The *circular correlation* operator can capture complex interactions of entity embeddings without increasing the representation dimensionality. Besides, it can be effectively computed via:

$$\boldsymbol{h} \star \boldsymbol{t} = \mathcal{F}^{-1}(\overline{\mathcal{F}(\boldsymbol{h})} \odot \mathcal{F}(\boldsymbol{t})) \tag{10}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are *fast Fourier transform* and its inverse respectively, $\bar{\mathbf{x}}$ denotes the complex conjugate, and $\odot$ is the entrywise Hadamard product.

We use the margin-based loss function [4] that preserves certain preferences between positive and negative triples. As the negative candidates do not have the equal semantic distance to any particular positive one, we introduce the semantic weights into different example pairs. That is to say, we reformalize the embedding learning problem as that of minimizing the following loss function:

$$L = \sum_{s \in P} \sum_{s' \in RN} \nu(s, s') \cdot [f_r(s') + \gamma - f_r(s)]_+, \tag{11}$$

where P denotes the positive example set, RN represents the reliable negative example set, γ is the margin value, $[x]_+ = max\{0, x\}$, f_r is the score function, the weight function $\nu(s, s')$ quantifies the semantic distance between any two triples of our concern. Intuitively, the larger the semantic distance between the positive and the negative examples, the more important it is to predict the preference between them correctly.

Formally, we define the weight function according to the predefined observed features which provide various views on the semantic differences. For example, the feature $\mathbf{1}(e_i = s \& r_k)$ measures the distance from the entity-type level, while the feature $\mathbf{1}(e_i = s)$ does so from the path level. Apparently, the latter can capture more complex dependencies. To distinguish their contributions to the weight function, we have:

$$\nu(s_1, s_2) = 1 - \frac{\sum\limits_{z \in Z_1} \frac{\cos_z(s_1, s_2)}{|Z_1|} + \delta \cdot \sum\limits_{z \in Z_2} \frac{\cos_z(s_1, s_2)}{|Z_2|}}{1+\delta} \tag{12}$$

where δ is a weight parameter chosen through cross validation, $cos_z(s_1, s_2)$ denotes the cosine distance between s_1 and

s_2 in the space of feature z. All six features are divided into two sets: Z_1 denotes the feature set comprising feature 3 and 4, Z_2 denotes the feature set comprising feature 1, 2, 5 and 6.

Consider a positive triple s *(Barack Obama, place of birth, Honolulu)*, and assume that triple s' *(Hillary Clinton, place of birth, Honolulu)* and s'' *(David Beckham, place of birth, Honolulu)* are two negative candidates. According to Eq. (12), the triple pair (s, s') should get smaller weight value in comparison with the triple pair (s, s'') due to feature 5. Since the entity pairs *(Barack Obama, UnitedStates)* and *(Hillary Clinton, UnitedStates)* are already directly connected by the relation nationality in the KGs.

After the training process, the optimized models calculate the probabilities for any unobserved triples w.r.t. the existing KGs data by means of the score function $f_r(\mathbf{h}, \mathbf{t}) = \sigma(\mathbf{r}^T(\mathbf{h} \star \mathbf{t}))$, and produce a ranked list of the predicted triples.

V. Experiments

A. Datasets and Evaluation Metrics

We choose two well-known datasets to evaluate our proposed model. **FB15k** [6] is a subset of Freebase (a collaborative knowledge base of general facts). **WN18** [6] is a subset of WordNet (a knowledge graph about words, which provides the information of synonym and antonym). For each dataset, we measure the performance of our algorithm averaged over five folds which are divided randomly.

For every triple (h, r, t) in the testing set (Under the *closed world assumption* and *Filter* setting), we remove the head entity h and sort all of the candidate entities including h according to the model's predicting scores. The similar operation is taken on the tail entity as well. We evaluate models based on their output ranking positions of testing ground truth entities. Specifically, we use MRR and Hit@10 as evaluation metrics: MRR is the average of the reciprocal rank results for the triples containing removed items (h or t); Hit@n is the number of triples containing removed items in the testing set that appear within the top n ranks. We compute the metrics for removing head or tail entities separately, which are then averaged.

B. Experimental Setup

We chose several common baseline methods as comparisons to our proposed method. These include latent feature models TransE [6] and HOLE [19], and observed feature model Node+LinkFeat [8]. TransE is one of the most widely used latent feature models that regard relations as translations from a head entity to a tail entity. HOLE is a recent latent feature model based on the circular correlation of vectors, which has the loss function most similar to our proposed method. Node+LinkFeat which can extract effective features from the local graph structure.

Our frameworks applied to the latent feature models and the observed feature models are respectively appended with the subscripts "L" and "O". $\mathrm{TSLRF_L}$ benefits from the **n**egative example **r**efining scheme (**NR**) and the **s**emantic **w**eighting scheme (**SW**). We examine the individual contribution of NR and SW components in $\mathrm{TSLRF_L}$. For clarity, we express the methods to be compared in explicit combining forms, including TransE+NR, TransE+SW, HOLE+NR and HOLE+SW. Parameters for TransE, HOLE, and Node+LinkFeat were respectively set as in the original implementations [6], [8], [19]. For $\mathrm{TSLRF_L}$ and $\mathrm{TSLRF_O}$, we use adaptive gradient stochastic gradient descent (AdaGrad SGD) [27] as the optimization algorithm. The dimension of latent features is set to 150. The margin parameter γ is the best value chosen through grid search. The max number of training epochs is set to 1000, and the saved model with the best MRR performance on the validation set is evaluated on the testing set.

TABLE I
THE TRIPLE PREDICTION RESULTS OF EMBEDDING-BASED MODELS.

	FB15 MRR	FB15k Hit@10	WN18 MRR	WN18 Hit@10
TransE [6]	0.295	0.512	0.394	0.895
TransE+NR	0.297	0.534	0.392	0.904
TransE+SW	0.319	0.562	0.395	0.917
HOLE [19]	0.377	0.575	0.831	0.921
HOLE+NR	0.384	0.617	0.834	0.933
HOLE+SW	0.381	0.609	0.833	0.941
$\mathrm{TSLRF_L}$	**0.392**	**0.645**	**0.840**	**0.944**

TABLE II
THE TRIPLE PREDICTION RESULTS OF OBSERVED FEATURE MODELS

	FB15 MRR	FB15k Hit@10	WN18 MRR	WN18 Hit@10
Node+LinkFeat [8]	0.806	0.865	0.932	0.941
$\mathrm{TSLRF_O}$	**0.814**	**0.887**	**0.942**	**0.948**

C. Results and Analysis

Table 1 displays triple prediction results of latent feature models. Benefiting from the semantic distance weighting and the reliable negative example set, our proposed $\mathrm{TSLRF_L}$ achieves the best performance in the triple prediction experiment. Models with the reliable negative example set ($\mathrm{TSLRF_L}$, HOLE+NR, and TransE+NR) perform better than corresponding models without the reliable negative example set (HOLE+SW, HOLE, and TransE) on most of the datasets. It indicates that using the generated reliable negative examples could improve the predicting ability of the latent feature models. Analogously, models that are with the semantic weighting scheme (TransE+SW, HOLE+SW, $\mathrm{TSLRF_L}$) achieve better prediction performances compared with TransE, HOLE, and HOLE+NR. This result shows the effectiveness of the semantic distance weighting strategy in our framework. It should be noticed that NR brings about more improvement to HOLE compared with SW. The reason might be that HOLE can exploit *Similarity Component* [19] through circular correlation.

Except for generating the reliable negative example set, our proposed TSLRF model can directly generate the two-class prediction score based on observed features for unknown triples as well. We define the final saved logistic regression

classifier g_2 as $\mathrm{TSLRF_O}$. Then we compare it with a state-of-the-art observed feature model Node+LinkFeat in MRR and Hit@10 measures. Table 2 shows triple prediction results of the two observed feature models. $\mathrm{TSLRF_O}$ achieves better performances than Node+LinkFeat. It denotes that generating a reliable negative example set for the observed feature prediction model is also helpful.

VI. Conclusion

In this paper, we first propose a PU-learning framework to iteratively improve the negative candidate pools for training triple prediction models. The experimental results validate the effectiveness of introduced negative selection scheme for both the latent feature models and observed feature models. Then, we devise a semantic distance weighting scheme to better the pairwise loss function, which is widely used in many triple prediction models. This weighting strategy effectively exploits additional observed features to improve the latent feature model further. Besides, experimental results also show that the combination of the two proposed schemes brings about substantial improvements over state-of-the-art methods.

Acknowledgment

This work was supported in part by National Key R&D Program of China(No. 2017YFC0803703), Beijing Municipal Natural Science Foundation under Grant No. 4172063 and the National Natural Science Foundation of China under Grants 61472423, 61432008 and U1636220.

References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.*, 2008, pp. 1247–1250.

[2] M. Fabian, K. Gjergji, and W. Gerhard, "YAGO: A core of semantic knowledge unifying wordnet and wikipedia," in *Proceedings of International World Wide Web Conference.*, 2007, pp. 697–706.

[3] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: a web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 601–610.

[4] A. Bordes and J. Weston, "Learning Structured Embeddings of Knowledge Bases," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2011, pp. 301–306.

[5] R. Socher, D. Chen, C. Manning, D. Chen, and A. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion," *Neural Information Processing Systems*, pp. 926–934, 2013.

[6] A. Bordes, N. Usunier, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-Relational Data," in *Proceedings of Advances in Neural Information Processing Systems.*, vol. 26, 2013, pp. 2787–2795.

[7] N. Lao and W. W. Cohen, "Fast query execution for retrieval models based on path-constrained random walks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, p. 881.

[8] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 2015, pp. 57–66.

[9] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2014, pp. 1112–1119.

[10] K.-W. Chang, W.-t. Yih, B. Yang, and C. Meek, "Typed tensor decomposition of knowledge bases for relation extraction," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.*, 2014, pp. 1568–1579.

[11] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2 SPEC. ISS., pp. 107–136, 2006.

[12] M. Bröcheler, L. Mihalkova, and L. Getoor, "Probabilistic Similarity Logic," in *Proceedings of the 2010 Conference on Uncertainty in Artificial Intelligence*, 2010.

[13] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *Proceedings of Artificial Intelligence and Statistics*, vol. 22, 2012, pp. 127–135.

[14] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A. Bordes, 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, 2013.

[15] I. Sutskever, "Modelling relational data using Bayesian clustered tensor factorization," in *Proceedings of Advances in Neural Information Processing Systems.*, 2009, pp. 1–8.

[16] M. Nickel, V. Tresp, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data," in *Proceedings of International Conference on Machine Learning.*, 2011, pp. 809–816.

[17] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2015, pp. 2181–2187.

[18] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge Graph Embedding via Dynamic Mapping Matrix," in *ACL (1)*, 2015, pp. 687–696.

[19] T. P. Maximilian Nickel, Lorenzo Rosasco, "Holographic Embeddings of Knowledge Graphs," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2016, pp. 1955–1961.

[20] N. Lao, T. Mitchell, and W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing.*, no. March, 2011, pp. 529–539.

[21] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation Extraction with Matrix Factorization and Universal Schemas," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*, 2013, pp. 74–84.

[22] C. Elkan and K. Noto, "Learning Classifiers from Only Positive and Unlabeled Data Charles," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*, 2008, pp. 213–220.

[23] R. Xia, X. Hu, J. Lu, J. Yang, and C. Zong, "Instance selection and instance weighting for cross-domain sentiment classification via PU learning," in *Proceedings of International Joint Conference on Artificial Intelligence.*, 2013, pp. 2176–2182.

[24] Y. Liu, S. Qiu, P. Zhang, P. Gong, F. Wang, G. Xue, and J. Ye, "Computational Drug Discovery with Dyadic Positive-Unlabeled Learning," in *Proceedings of International Conference on Data Mining.(SIAM)*, 2017, pp. 45–53.

[25] B. Kotnis and V. Nastase, "Analysis of the impact of negative sampling on link prediction in knowledge graphs," *arXiv preprint arXiv:1708.06816*, 2017.

[26] W. S. Lee and B. Liu, "Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression," *Algorithmic Learning Theory*, vol. 348, no. 1, pp. 71–85, 2003.

[27] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2010.