Enhancing Knowledge Graph Embedding with Probabilistic Negative Sampling

Vibhor Kanojia Hideyuki Maeda Riku Togashi Sumio Fujita
Yahoo Japan Corporation
Tokyo, Japan
{vkanojia, hidmaeda, rtogashi, sufujita}@yahoo-corp.jp

ABSTRACT

Link Prediction using Knowledge graph embedding projects symbolic entities and relations into low dimensional vector space, thereby learning the semantic relations between entities.

Among various embedding models, there is a series of translation-based models such as *TransE*[1], *TransH*[2], and *TransR*[3]. This paper proposes modifications in the *TransR* model to address the issue of skewed data which is common in real-world knowledge graphs. The enhancements enable the model to smartly generate corrupted triplets during negative sampling, which significantly improves the training time and performance of *TransR*.

The proposed approach can be applied to other translation-based models.

Keywords

Knowledge Graph Embedding; TransR; Probabilistic Negative Sampling.

1. INTRODUCTION

Existing translation-based embedding models follow a similar principle. Given a fact of knowledge base, represented by a triple *(h, r, t)* where h, r, t indicate a head entity, a relation and a tail entity respectively, these models aim to learn embedding vectors h_r, r, and t_r such that $h_r + r \approx t_r$ when projected with respect to the relation space.

Knowledge bases such as Word-net (WN18) and Freebase (FB15K) have been proven to be very useful as training datasets for graph embedding but suffer from skewness of data.

For example, in the Freebase training dataset, relation */award/award_nominee/award_nominations* has $15,998$ triplets, whereas */award/award/category* has only 144 triplets. Similarly, Figure 1 shows the distribution of triplets for each relation in the WN18 training dataset. The fluctuations in Figure 1 show the scale of skewness in the dataset which eventually leads to poor performance of translation-based embedding models for relations with less data.

WWW'17 Companion, April 3–7, 2017, Perth, Australia.

ACM ISBN 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054238

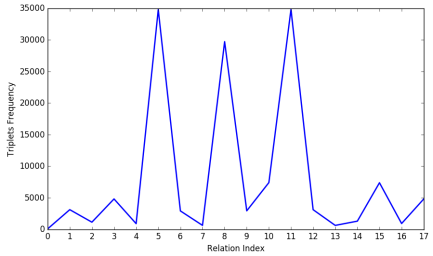

Figure 1: Distribution of triplets for the 18 relations in the WN18 training dataset

2. RELATED WORK

Existing embedding methods define the score function as

$$f_r(h,t) = \|h_r + r - t_r\|_2^2 \quad (1)$$

where h_r and t_r are relation-specific entity embedding vectors.

TransE(Bordes Et al., 2013) does not differentiate between entity and relations, and lays them into the same space: $h_r = h$, $t_r = t$, and tries to minimise the score function in order to make $h_r + r \approx t_r$. Training phase uses margin-based ranking loss to encourage discrimination between golden triplets and incorrect triplets:

$$\mathcal{L} = \sum_{(h,r,t)\in\triangle} \sum_{(h',r',t')\in\triangle'} [f_r(h,t) + \gamma - f'_r(h',t')]_+ \quad (2)$$

where $[x]_+ \triangleq max(0,x)$, $\triangle$ is the set of positive(golden) triplets, $\triangle'$ denotes set of negative triplets generated by randomly corrupting (h, r, t). This step is known as *Negative Sampling*. γ is the margin separating positive and negative triplets.

TransH(Wang et all., 2014) makes use of hyperplanes to address the issue of complex relation embedding. Entities are projected into the hyperplane before the score function is applied: $h_r = h - w_r^\top h w_r$, $t_r = t - w_r^\top t w_r$. In order to reduce False Negative Labels, TransH uses the type of the relation(one-to-one, many-to-one, etc) to decide whether to replace head or tail.

TransR(Lin et al., 2015b) addresses the same issue of complex relation embedding by transforming the entity embeddings by the same relation-specific matrix: $h_r = M_r h$ and $t_r = M_r t$.

All of the above mentioned methods make use of Negative Sampling, and none of them address the issue of skewness of data. This paper proposes *Probabilistic Negative Sampling* to deal with this issue.

3. PROBABILISTIC NEGATIVE SAMPLING

Drawbacks of Current Approach: We observed that the model is able to learn the latent semantics for each relation after just 200 epochs. That is, the model is able to predict that the relation *spouse* is followed by a *human-name* type tail, and *birthdate* is followed by a *date-format* type entity as tail.

During the Negative Sampling phase, if we replace the heads and tails randomly, the generated corrupted triplets, like $\triangle(Obama,\ spouse,\ Europe)$, do not contribute much to the training phase in the later half because the value, $f'_{spouse}(Obama, Europe)$ is already big enough to have any effect on the loss function.

Motivation: Even after 1000 epochs, the model fails to predict the correct tail amongst the semantically possible options for relations with less triplets. This is because the process of distinguishing between correct and incorrect tails amongst the semantically possible options is very slow. We aim to make this process fast by having the model generate corrupted triplets smartly from the semantically possible options.

Our Approach: In our approach, we use training data to create a list of possible instances of heads and tails for each relation. As a result, the relation *spouse* will have a list of entities of type *human-name*, similarly the relation *gender* will have a list with only two elements *Male* and *Female*. We define a tuning parameter β known as *train bias*, and vary its value in the range $[0.0, 0.4]$ during the training phase. The value of β determines the probability by which our model will be biased towards the training data in the Negative Sampling phase. In other words, $\beta = 0.2$ means, 20% of the times the model will generate a corrupted triplet with head/tail selected from semantically possible options for a particular relation. This ensures that even if the training data has only a few triplets for a relation r, the corrupted triplets generated will be biased towards semantically possible options in the later phase of training, thereby speeding up the distinguishing process. Note that the value of β is not static, but increased gradually during the training phase.

4. EVALUATION AND RESULTS

We evaluated our approach, TransR-PNS, over TransR in Link Prediction tasks, e.g. predicting missing h or t in a triplet (h, r, t), using two datasets, FB15K[1], general fact collection provided by Freebase, and WN18[2], semantic knowledge of words provided by WordNet; both are used in previous studies [1],[2],[3].

We used two measures as our evaluation metric: (1) Mean Rank of correct entities; and (2) proportion of correct entities in top-10 ranked entities (Hits@10). Ranking is decided by score function f_r. A good model should achieve lower mean rank, or higher Hits@10. Note that the results that we achieved vary slightly from the results mentioned in the original papers due to the difference in the configuration and tuning parameters. All the results are obtained over "raw" setting, i.e., we do not remove a correct triplet generated as corrupted triplet during negative sampling phase. For experiments of original TransR and our approach (TransR-PNS), we used Adam Optimiser with initial learning rate λ = 0.001, batch size = 256, dimension of vectors d = 50, dimension of matrix k = 50, varied the margin γ in the range = {0.2, 0.5, 1.0, 2.0}, train bias β in the range = [0.0, 0.4] and used L_2 distance between vectors. For both FB15K and WN18, we trained the model for 2000 epochs. Evaluation results and comparison of the proposed approach with other approaches are given in Table 1, which shows our approach performed significantly better than all the previous approaches in all aspects.

[1]#Rel:1,345; #Ent:14,951; #Train:483,142; #Test:59071

[2]#Rel:18; #Ent:40,943; #Train:141,442; #Test:5000

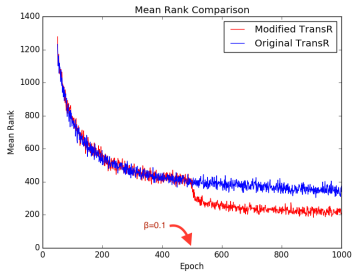

Figure 2: Comparison of Original TransR and TransR-PNS over Mean Rank metric for FB15K dataset

Table 1: Evaluation Results

Approach	WN18		FB15K	
	Mean Rank	Hits@10	Mean Rank	Hits@10
TransE	263	0.754	243	0.349
TransH(bern)	401	0.73	212	0.457
TransR(bern)	205	0.759	191	0.464
TransR-PNS	**15**	**0.764**	**144**	**0.472**

5. CONCLUSIONS

Our experiments prove that instead of generating corrupted triplets randomly, generating them based on a probabilistic model improves the model's learning speed and effectiveness drastically. This can be verified from Figure 2, where changing the value of β results in a drastic performance improvement. With both datasets, our approach outperforms the previous methods on Mean Rank as well as Hits@10, achieving 0.66% and 1.72% gains in Hits@10 against TransR, which is the best among the previous methods. Although these seem to be small, considerable improvements in Mean Rank (190 and 47 position gains) suggest that our method possibly enables users the significant amelioration of service experiences. The proposed probabilistic approach can also be applied to the current state-of-the-art approach TransG[4] to further improve its performance.

6. REFERENCES

[1] 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*, pages 2787–2795, 2013.

[2] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119. Citeseer, 2014.

[3] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[4] H. Xiao, M. Huang, and X. Zhu. Transfig: A generative model for knowledge graph embedding. In *Proceedings of the 29th international conference on computational linguistics. Association for Computational Linguistics*, 2016.