Effects of Negative Sampling on Knowledge Graph Completion

Betül Bayrak
Department of Computer Engineering
Gazi University
Ankara, Turkey
betulbayrak@gazi.edu.tr

***Abstract*—Knowledge graph embedding concepts have become popular in recent years. The most common usage area of this concept is knowledge graph completion because ordinarily, knowledge graphs consist of a large amount of structured data, but they are not complete. To perform the completion task, embedding models use positive triples, which are in the knowledge graph and some negative triples which artificially generated for each positive triples. This study is aimed to investigate the effects of negative sampling on the knowledge graph completion task. Several experiments are realized with well-known knowledge graph embedding models on different sized knowledge graphs to show these effects. We applied TransE, ComplEx, DistMult, and ConvKB models to Kinship and FB15k-237 knowledge graphs. In addition, as negative sampling method, we used random-corruption negative sampling.**

***Keywords*—negative sampling, corrupting triples, knowledge graph completion (KGC), knowledge graph embedding (KGE)**

I. Introduction

Knowledge graphs consist of entities and relations, $i.e.$ $KG = (\mathcal{E}, \mathcal{R})$. Generally, they include large amounts of structured data as triples. These triples are like (s,p,o), s for subject, p for predicate, and o for object, $e.g.$ $(Oliver, friend\_of, Alima)$. Subject and object are also an entity, predicate is a relation, $i.e.$ $s, o \in \mathcal{E}$, $p \in \mathcal{R}$. There are lots of applications of knowledge graphs. The most popular ones are knowledge graph completion [1] [2], entity alignment [3], question answering [4], and item recommendation [5]. However, there is a common thread to all these applications, vectorizing the triples, to perform their main tasks. Creating vectors from the triples are known as knowledge graph embedding in literature. There are many knowledge graph embedding models, $e.g.$ TransE [6], ComplEx [1], ConvKB [7], TransR [8], DistMult [9], Rescal [10], RotatE [11].

As mentioned before, knowledge graphs are large and open to extract new knowledge without external data. Therefore, knowledge graph completion is a critical issue. For example, $(Oliver, friend\_of, Alima)$ and $(Oliver, friend\_of, Sophia)$ triples exist in the knowledge graph. However, there is not a $friend\_of$ relation between $Alima$ and $Sophia$, even though that may be correct. Whether $(Alima, friend\_of, Sophia)$ is correct or not can be extracted using knowledge graph completion methods. Moreover, if it is correct, it can be added to knowledge graph.

A knowledge graph consists of triples. These triples are used for the knowledge graph completion tasks. Commonly, these triples are named as positive triples or golden triples. However, artificiall created negative samples can also be used to enhance the success rate of the tasks. The number of created negative samples for each triple is an essential parameter on success. Negative triples are created by various techniques (e.g. random [12], adversarial [11], offset-based [13], nscaching [14]).

The rest of this study is organized as follows. In section-II we explain the negative sampling method used in this study, and in section-III, there is a detailed explanation of the experiments, datasets, parameters, evaluation metrics, and results. In the end, you can fin the conclusion of the study and also future work ideas.

II. Randomly Corrupted Negative Triples

Random negative triple generation (i.e. random sampling) is a traditional and the most commonly used negative sampling method. We can split random sampling into two different branches, fully-random sampling and random corruption of triples. In this study, we used the second one.

(n_s: number of negative samples for each positive triples, S: set of positive triples, S': set of negative triples)

To generate S' from S, for every (s, p, o) create n_s negative triples.

A. Fully-random Sampling

To create (s', p', o'), choose randomly s' and o' from entities, p' from relations. $s', o' \in \mathcal{E}$, $p \in \mathcal{R}'$ and $(s', p', o') \notin S$.

B. Random Corruption Sampling

In random corruption, to create corrupted triples, corruption side is important. Corruption side can be 's', 'o', 's, o', '$s+o$'. If corruption side:

- s: (s', p, o), $s' \in \mathcal{E}$, $(s', p, o) \notin S$
- o: (s, p, o'), $o' \in \mathcal{E}$, $(s, p, o') \notin S$
- s, o: (s', p, o) or (s, p, o') choose randomly, $s', o' \in \mathcal{E}$, $(s', p, o), (s, p, o') \notin S$
- $s + o$: (s', p, o'), $s', o' \in \mathcal{E}$, $(s', p, o') \notin S$

For all corruption sides, s' and o' are chose randomly from all entities.

978-1-7281-7565-2/20/$31.00 ©2020 IEEE

III. Experiments

A. Datasets

We used two different knowledge graphs, FB15k-237 and Kinship.

1) FB15k-237: FB15k [15] is a part of FreeBase and FB15k-237 [16] dataset is deleted version of FB15k where FB15k- 237 [9] has reverse relations. As shown in Table-I, there are 310116 triples in the dataset.

2) Kinship: Kinship dataset[1] is a real-world dataset. It patterns 25 types of kinship terms among 104 people. (See Table-I)

In this study, we chose the datasets described above because they are well-known knowledge graphs. And also, we wanted to show the differences in effects according to dataset size and type.

TABLE I. Datasets Details

KG Name	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Train	Test	Total
FB15K-237	14541	237	289650	20466	310116
Kinship	104	25	9612	1074	10686

B. Experimental Setup

We used the same parameters for each model because we wanted to stabilize all the situations and make fair comparisons on the results. We used 'adam' optimizer with $1e-4$ learning rate and multi-class negative log-likelihood loss function with $\lambda = 1e-5$. Our embedding dimension was 200, the number of the iterations was 400, and the batches count was 100. In the experiments, although we tried to show how using negative sampling affects the results, we wanted to show how the number of negative samples affects the results too. Therefore, [0,1,5,10,20,30,40,50,60,70,80,90,100] were used as the number of negative sampling parameter.

C. Evaluation Metrics

$r_{(s,p,o)}$ function is a ranking function; it calculates ranking of the (s,p,o) triple and returns a ranking.

1) Mrr: Mean reciprocal rank

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{r_{(s,p,o)_i}} \tag{1}$$

2) Mr: Mean rank

$$MR = \frac{1}{|Q|}\sum_{i=1}^{|Q|} r_{(s,p,o)_i} \tag{2}$$

3) Hits@N: Ratio of triples which have top-N ranking to all the other test triples.

$$Hits@N = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\begin{cases}1 & if(r_{(s,p,o)_i} \leq N)\\ 0 & otherwise\end{cases} \tag{3}$$

TABLE II. ComplEx Results

ComplEx	**FB15k-237**				
n_s	***Mrr***	***Mr***	***H@10***	***H@3***	***H@1***
0	0.00066	7146.91372	0.00061	0.00017	0.00002
1	0.29056	202.57789	0.46942	0.32301	0.20060
5	0.30423	193.76227	0.49408	0.34285	0.20698
10	0.30395	191.23291	0.49868	0.34192	0.20566
20	0.30353	191.84659	0.50198	0.34216	0.20331
50	0.30086	192.88532	0.50179	0.33673	0.20116
100	0.29686	193.95315	0.49966	0.33519	0.19588
	Kinship				
0	0.05985	47.64665	0.12104	0.03911	0.01350
1	0.49168	6.10568	0.80540	0.52048	0.37104
5	0.48598	6.97207	0.75605	0.50140	0.37989
10	0.47961	7.35242	0.73464	0.49628	0.37430
20	0.47614	7.83101	0.70391	0.48743	0.37663
50	0.47066	8.47300	0.67039	0.47998	0.37523
100	0.46692	9.03771	0.64665	0.47812	0.37523

TABLE III. DistMult Results

DistMult	**FB15k-237**				
n_s	***Mrr***	***Mr***	***H@10***	***H@3***	***H@1***
0	0.00070	7126.22029	0.00059	0.00012	0.00010
1	0.27561	232.06079	0.44097	0.30178	0.19257
5	0.29380	209.32098	0.47368	0.32771	0.20246
10	0.29726	206.29730	0.48317	0.33143	0.20373
20	0.29693	207.08097	0.48718	0.33218	0.20065
50	0.29489	207.93187	0.49317	0.33292	0.19590
100	0.29312	210.57853	0.49366	0.32939	0.19387
	Kinship				
0	0.05259	49.65875	0.09544	0.03585	0.00978
1	0.50950	5.04749	0.87011	0.57263	0.35661
5	0.51442	5.08845	0.86499	0.57263	0.36778
10	0.50708	5.16993	0.86127	0.56378	0.35894
20	0.50742	5.21136	0.85987	0.56192	0.36127
50	0.50451	5.27700	0.85894	0.55680	0.36080
100	0.50160	5.29842	0.85801	0.55074	0.35615

TABLE IV. TransE Results

TransE	**FB15k-237**				
n_s	***Mrr***	***Mr***	***H@10***	***H@3***	***H@1***
0	0.00502	7209.92874	0.00494	0.00453	0.00445
1	0.29471	191.44706	0.47862	0.32724	0.20339
5	0.30998	188.63386	0.49565	0.34530	0.21584
10	0.30879	190.51284	0.49821	0.34493	0.21346
20	0.31029	193.03498	0.50213	0.34970	0.21271
50	0.30643	195.82937	0.50313	0.34948	0.20498
100	0.30257	198.04132	0.50450	0.34784	0.19879
	Kinship				
0	0.04051	49.36453	0.08007	0.01909	0.00000
1	0.18019	29.46229	0.36080	0.21369	0.07775
5	0.19270	28.65642	0.37989	0.22160	0.09125
10	0.20958	28.37803	0.38361	0.23557	0.11453
20	0.21679	28.52933	0.39479	0.23836	0.12244
50	0.22180	28.65270	0.39525	0.23929	0.13035
100	0.22359	28.73184	0.39199	0.24162	0.13222

TABLE V. ConvKB RESULTS

ConvKB	FB15k-237				
n_s	***Mrr***	***Mr***	***H@10***	***H@3***	***H@1***
0	0.00052	7162.09729	0.00027	0.00005	0.00002
1	0.24620	286.64712	0.40601	0.26760	0.16799
5	0.26049	266.68648	0.42952	0.28536	0.17682
10	0.26139	261.00815	0.43662	0.28634	0.17604
20	0.26501	257.31347	0.44097	0.29113	0.17848
50	0.26309	263.10519	0.43889	0.28901	0.17694
100	0.26128	268.16329	0.44063	0.28612	0.17428
	Kinship				
0	0.05308	48.44786	0.10754	0.02840	0.00978
1	0.15892	30.28911	0.32542	0.16387	0.07123
5	0.16793	29.74953	0.33659	0.17365	0.08240
10	0.16616	29.42877	0.34451	0.18250	0.07169
20	0.16705	29.58566	0.34358	0.17970	0.07542
50	0.16648	29.49488	0.34730	0.18948	0.06797
100	0.17012	29.48883	0.34963	0.18622	0.07635

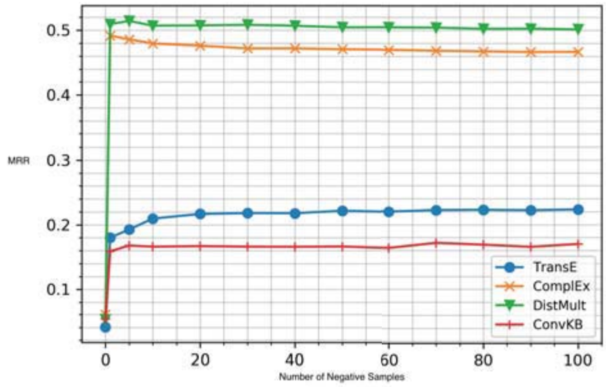

Fig. 1 Kinship dataset MRR results for 4 models

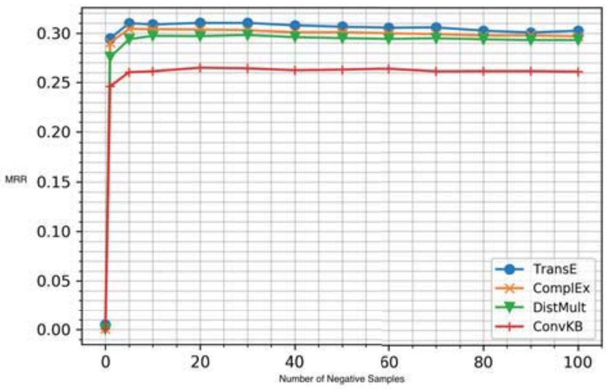

Fig. 2 FB15k-237 dataset MRR results for 4 models

D. Results

We conducted several experiments on ComplEx, DistMult, TransE, and ConvKB models. For each experiment, we changed n_s parameter to show the number of negative samples effects on these models and showed detailed results respectively, in tables II, III, IV, and V. Besides that, there are also two graphs for comparing MRR results of mentioned models on kinship (see Fig-**??**) and FB15k-237 (see Fig-**??**) datasets.

In the experiments, 0, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 were used as n_s parameter. When $n_s = 0$ ($i.e.$ when there is no negative sample created), MRR and Hits values are very close to 0 for all models. As you can see in the tables and figures there is a sharp increase between $n_s = 0$ and $n_s = 1$, for each model and dataset. There is a noticeable increase between $n_s = 1$ and $n_s = 5$, though not as much as between $n_s = 0$ and $n_s = 1$. There are different peak points in the experiments; the peak points change according to model and dataset. Some of the models generally reached the peak point while $n_s = 1$, $n_s = 5$, $n_s = 10$, and $n_s = 20$ in these experiments. There are some exceptions too. For example, in table-IV, the peak point at $n_s = 100$ on kinship dataset but there is not a considerable difference between $n_s = 20$ and $n_s = 100$.

Time and space complexity changes up to n_s value. We summarized the situation in 3 states. First, when n_s value is so small ($i.e.$ $n_s < 5$ for our experiments), time complexity will low, and so will the results. Second, when n_s is an average value, time complexity will be quite close to the firs one, but the results will significantl increase. Lastly, when n_s value is enormous ($i.e.$ $n_s > 30$ for our experiments), time and space complexity will be large, and the results will not change considerably.

IV. CONCLUSION

In this study, several experiments were realized on four knowledge graph embedding models with two datasets for the knowledge graph completion task. These experiments are realized to show how the number of negative samples affects the knowledge graph completion results. There is a direct proportion between n_s and complexity; nevertheless, it is not the same for the n_s and the results. There are several n_s values for the peak points of the results, and it changes up to model and dataset; however, the spread of the n_s values for the peak points is not quite notable. In the results, we showed that there is a significan difference between negative samples not used and negative samples used; besides that, when n_s value is huge, the cost of the process is high but no considerable changes in the results.

For further studies, instead of using only random-corruption negative sampling, other popular negative sampling techniques can be applied to distinct applications on several distinct knowledge graphs, and the effects of negative sampling can be observed more detailed.

[1] github.com/TimDettmers/ConvE

REFERENCES

[1] T. Trouillon, C. R. Dance, É. Gaussier, J. Welbl, S. Riedel, and G. Bouchard, "Knowledge graph completion via complex tensor factorization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4735–4772, 2017.

[2] B. Shi and T. Weninger, "Proje: Embedding projection for knowledge graph completion," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[3] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding." in *IJCAI*, vol. 18, 2018, pp. 4396–4402.
[4] J. Bao, N. Duan, Z. Yan, M. Zhou, and T. Zhao, "Constraint-based question answering with knowledge graph," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2503–2514.
[5] E. Palumbo, G. Rizzo, R. Troncy, E. Baralis, M. Osella, and E. Ferro, "Knowledge graph embeddings with node2vec for item recommendation," in *European Semantic Web Conference*. Springer, 2018, pp. 117–120.
[6] 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.
[7] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," *arXiv preprint arXiv:1712.02121*, 2017.
[8] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
[9] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.
[10] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data." in *Icml*, vol. 11, 2011, pp. 809–816.
[11] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.
[12] 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.
[13] X. Han, S. Cao, X. Lv, Y. Lin, Z. Liu, M. Sun, and J. Li, "Openke: An open toolkit for knowledge embedding," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 139–144.
[14] Y. Zhang, Q. Yao, Y. Shao, and L. Chen, "Nscaching: Simple and efficien negative sampling for knowledge graph embedding," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 614–625.
[15] 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.
[16] 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.