COMP 550 - Incremental Knowledge Graphs

GROUP 7: Ashita Diwan (260906861), Nishant Mishra(260903177)
Priyesh Vijayan (260911934)

Abstract

Knowledge graphs (KGs) succinctly represent real-world facts as multi-relational graphs. A plethora of work exists in embedding the information in KG to a continuous vector space in order to obtain new facts and facilitate multiple down-stream NLP tasks. Despite the popularity of the KG embedding problem, to the best of our knowledge, we find that no existing work handles dynamic/evolving knowledge graphs that incorporates facts about new entities. In this project, we propose this problem as an incremental learning problem and propose solutions to obtain representations for new entities and also update the representations of old entities that share facts with these newer entities. We build our solutions with TransE as our base KG embedding model and evaluate the learned embeddings on facts associated with these new entities.

1 INTRODUCTION

Numerous Knowledge Graphs (KGs) are a means of accessible sources for real-world facts that provide information about different entities and their relationships with each other. A fact in a KG is represented by a triplet, (h, r, t), which denotes that a head entity, h is related to a tail entity, t by the relation, r. Numerous NLP models benefit from incorporating external information from these KGs. In order to effectively use KGs for various machine learning-based NLP tasks, it is essential to vectorize the information in KGs. This involves learning continuous representations for the entities and relations in a KG such that the data manifold of the learned representations preserves the inherent neighborhood structure of the KG. While there exist numerous models to learn KG embeddings (Kazemi et al., 2019), they are limited to static KGs and cannot model an evolving KG, where new facts about the world are constantly added or updated. New facts provide additional information about existing entities (new edges) in the KG and also may provide added information about new entities (new nodes). In this project, we focus on providing methodologies to obtain representations (embeddings) for these new entities and also update the representations of existing (old) entities that have received new facts.

Though obtaining new entity representations is the focus of the work, the primary task is still to perform link prediction for KG completion. However, note that it is different from the conventional KG completion task where the graph is static, and new facts/links are predicted only for a fixed set of existing entities. Existing models cannot be used to obtain facts for entities that are not present in the graph. The typical naive solution is to train the KG embedding model from scratch every time a new set of facts arrive, which is computationally expensive with large graphs.

Hence, it is ideal to look forward to solutions where the old model can quickly adapt to new training data without forgetting the information learned in the past. This is the classical problem of incremental learning, which is also commonly referred to as continual learning and online learning (Chen and Liu, 2016). We pose the problem of continually incorporating facts about new entities in the KG model as an incremental learning problem. Note that the parameters of the model are the embedding of entities themselves and their relations. Thus, the problem is more than learning a mere prediction function to obtain representations for new entities, but it also requires to update representations of existing entities that are related to the new entities with minimal gradient updates.

The primary motive of this setup is to avoid relearning the knowledge graph embedding altogether with the occurrence of every new set of facts (triplets). To this aim, we formulated two solutions; the first approach followed a fine-tuning based transfer-learning solution, and the second followed a model-agnostic meta-learning based approach with Graph Convolutional Networks (GCN). While our model-specific fine-tuning approach fared well, the proposed model-independent approach failed to learn representations for a new entity. Owing to space constraints, herein this report, we only discuss the results of a fine-tuning based Incremental learning approach.

We organize our paper as follows. First, in Section 2, we provide the background on TransE, a popular KG embedding model, and discuss a few other related works. Then, we formally pose the Incremental KG embedding problem and discuss the proposed methodologies to solve the problem. Then we provide the dataset and experiment details in Section 4 and discuss the observed results in Section 5. Finally, we conclude our report discussing our findings in detail in Section 6, and note areas for future work.

2 RELATED WORK

Several models and paradigms have been proposed in the literature to learn embeddings for KGs (Wang et al., 2017). Herein we build over the popular, TransE, a translation based model.

2.1 Background on TransE

TransE (Bordes et al.) is used to learn embeddings for entities and relations of multi-relational KG data in low-dimensional vector spaces. TransE models the embeddings by regarding a relation as translation from head entity to tail entity - if (h, r, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r. The scoring (energy) function for TransE is given by,

$$E(h, r, t) = ||h + r - t||$$

Given a training set S of triples (h, r, t), where $h, t \in E$ (E is the set of entities) and $r \in R$ (R is the set of relations), the embeddings are learnt by minimizing a margin-based ranking criterion over the training set.

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h',r,t')}} [\gamma + d(h+r, t) - d(h'+r, t')]_+$$

where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyperparameter and $S'_{(h',r,t')}$ is the set of corrupted triplets.

$$S'_{(h',r,t')} = \{(h', r, t)|h' \in E\} \cup \{(h, r, t')|t' \in E\}$$

The set of corrupted triplets is composed by replacing either the head or tail entity with a random entity. The loss function favors lower energy values for training triplets than for the corrupted ones. This model requires reduced set of parameters as it learns only one low-dimensional vector for each entity and each relationship. We used OpenKE's [1] implementation for setting our model. For our task, we made changes to the TransE model, so that it can learn the representations of the new entities.

2.2 Other related works

TransE works well in modelling 1-to-1 relations, but the performance is unsatisfactory in case of modelling 1-to-N, N-to-1, and N-to-N relationships. To overcome this, (Wang et al., 2014) proposed TransH, which models the relations as hyper-planes with translation operations on them. TransE and TransH project the entities and relations in the same semantic space. (Lin et al., 2015) proposed TransR to build entity and relation embeddings in separate entity space and relation spaces. (Xie et al., 2016) introduced a representation learning based approach to learn embeddings for both entities and relations in a common vector space using triplets as well as additional semantic information in the form of entity description.

(Kazemi et al., 2019) provides a comprehensive review of the advances in representation learning for dynamic graphs. The work points out that the literature on dynamic graphs is limited to only a few and that too only for temporal graphs where new facts or added or old facts dropped in a KG with static number of entities. The review also acknowledges the need for models that can learn representations for new entities and that there is no existing work that focuses in this setup though this is natural in the real-world.

The scarcity of models for incremental learning in KGs can be attributed to the complexity of multi-relational graphs. Only in the recent past, embedding models for simple dynamic graphs have surfaced. In the NLP side of research, (Kaji et al) explored an incremental training strategy for training word embeddings using skip-gram models. We adopt ideas from this paper for setting up our incremental approach to dynamically upgrade the embedding dictionary and update the negative sampling probability.

3 Experiment Details

In this section, we elaborate on the dataset and experimental setup used in our project.

[1]https://github.com/thunlp/OpenKE

Triples in training	Triples in validation	Triples in testing	Triples in test_zeroshot	No. of Entities	No. of Relations
4,83,142	50,000	59,071	31,078	19,970	1,345

Table 1: Statistics of FB20K dataset.

3.1 Dataset setup

Freebase provides general facts of the world, and is an extensively used dataset for knowledge completion tasks. We employed the FB20K dataset (Xie et al., 2016) for our task. In addition to containing all the entities and relations from the FB15K dataset, this dataset also contains new entities which was required for our setup. As a first step, we analyzed and visualized this dataset to gain insights. The statistics of the dataset are given in Table 1.

The FB20K dataset contains 1,345 relations and 19,970 entities in total, out of which 14,951 are present in the training set. The test_zeroshot file contains all the new entities that are not seen in the training data, while the test file contains the triples between the existing entities (e-e). The zeroshot setting contains a total of 9,217 entities, from which 5,019 are the new ones. The triplets in test_zeroshot can be split in three categories: (e-d), (d-e), and (d-d); where e stands for existing and d stands for the new entities. (e-d) means a triple with an existing entity as its head and a new entity as its tail, (e-d) implies that the tail is a new entity but the head is not, and (d-d) implies that both the head and the tail are new entities. The statistics on the zero-shot setting are given in Table 2.

	e-d	d-e	d-d
Triplets	11,880	19,047	151
Total Entities	7,916	9,176	110
New Entities	4,155	5,012	110

Table 2: Statistics of test_zeroshot.

The original FB20K dataset was proposed for Zero-shot learning setup where the data for learning representations for new entities did not have any new edges and the model leverage textual information about the entities to learn representations. To adapt this data to our setup where we don't leverage any additional side information such as text and leverage only limited edge/facts of new entities.

For the task of incremental learning of knowledge graph embeddings on new entities, we took the whole training data, and obtained embedding using TransE. We split the facts in test_zeroshot data into two sets such that both the set had the same set of new entities (4,427) and removed all entities which had only one fact associated with them. We used one for obtaining embeddings for new entities and the other for evaluating their performance.

3.2 Evaluation

We evaluate the models for link prediction, which aims to predict the missing h or t for a relation fact (h, r, t). For each missing entity, we rank the a set of candidate entities from the KG, rather than predicting just the best candidate. In testing phase, for each test triple (h, r, t), we replace the head/tail entity by all entities in the knowledge graph, and rank these entities in descending order of similarity scores calculated by score function. We used the Mean Rank of correct entities, and the proportion of correction entities in top-K ranked entities (Hits@K) measures as our evaluation metric (Lin et al., 2015). A good link predictor should achieve lower mean rank or higher Hits@K. We conduct experiments on the FB20K dataset, and compute the MR, Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10 to compare the performance of the models.

3.3 Skyline Model

In order to comprehend the performance of the proposed models, we compare the model against TransE trained from scratch on the entire training data and the observed facts of new entities, i.e, first set prepared from test_zeroshot. The model is evaluated for discovering the remaining facts in the second triplet set. The aim of the proposed the models discussed next is to close the gap in performance achieve this skyline performance.

4 Proposed Work

4.1 Problem of Incremental Learning

We formulate the problem of Incremental learning on KGs as follows. At time-step, $t+1$, if $\mathcal{T}_t$ denotes the facts set at t and $\mathcal{Z}_t \in \mathbb{R}^{|\mathcal{E}_t|*d}$ denote the embeddings for the entity set $\mathcal{E}_t$ at t, then, the task is to obtain embeddings, $\mathcal{Z}_{t+1} \in \mathbb{R}^{|\mathcal{E}_{t+1}|*d}$ based on the new (updated) facts set, $\mathcal{T}_{t+1}$. Note that, $\mathcal{T}_{t+1} - \mathcal{T}_t$ corresponds to the new facts added and $\mathcal{T}_t - \mathcal{T}_{t+1}$ corresponds to the deleted facts. We restrict the scope of project to cases where $|\mathcal{E}_{t+1}| > |\mathcal{E}_t|$ and to only one future prediction, i.e $t \in \{0, 1\}$ as we don't intend to capture the temporal dynamics.

4.2 Incremental learning via fine-tuning

We setup our problem as that of fine-tuning existing knowledge graph embeddings upon introduction of new entities. Therefore, with the introduction of new entities, new facts are added which need to be assigned a suitable representation in the same space as the entities and relations in the old graph. Not only this, addition of new nodes also affects the embeddings of entities associated with them. So, we intended to solve the problem of efficiently extrapolating our knowledge graph embeddings to accommodate new entities being added without having to be retrained from scratch. One of the important assumptions with this was that no new relation was being added.

We began by training a base model on the training data, on the 14,951 entities and 1,345 relations and saved the embeddings. We used the OpenKE toolkit (Han et al., 2018) for training TransE embeddings. The model took 1,000 iterations to converge to an optimum loss on the given triplets. The implementation of TransE uses the *'bern'* strategy for negative sampling introduced in (Wang et al., 2014) by corrupting entities and relations to obtain negative samples.

Once we had our base embeddings, we proceeded to fine-tune it to the new entities. We used the first set of test_zeroshot triplets as our training data for fine-tuning. Since no new relations were added, we froze the weights associated with relation embedding in the base model. Since we want to get the new node embeddings and the updated representation of old nodes associated with them, we found the neighbor nodes of the new nodes introduced according to the new triplets, which was basically the set of all old entities related to a new node. We froze the weights of all the entities that are not one of the new nodes or their neighbor in order to speed up the training. This selective freezing of weight vectors was done by changing the trainer model in OpenKE, setting a gradient mask that forces the gradients of weights associated with these entities to be zero at each epoch. Then, the pre-trained model with partially frozen weights was trained on the new triplets using the same strategy. It was allowed to learn for 2,000 epochs to converge.

We also fine-tuned a version of the pre-trained model without freezing any of the weights for the relations or the neighboring entities, it did not perform as well as the one with frozen weights.

4.3 Incremental learning via Meta learning

We also trained a GCN based model to learn an embedding-model agnostic approach for incremental learning. The idea here is to learn a GCN to predict the embeddings of new nodes given the old embeddings of it is neighboring entities in the old graph and similarly obtain an updated representation of old entities based on the recently learned embedding of new entities. These two predictions are jointly iterated. This can be viewed as learning to learn problem (meta-learning). While the model was able to learn to update embeddings of old entities, it failed at learning representations for new entities, i.e. the training loss did not reduce to a satisfactory level. To train this model, we prepared multiple sets of 2 time-stamped graphs and learned a GCN to learn embeddings for new entities when a new time-stamped graph set is provided. The code for this model is also submitted.

5 RESULTS

We gauge the effectiveness of the proposed Incremental learning approaches with the TransE model, under three different experimental settings. a) the computationally inefficient, *skyline*, that trains from scratch, b) fine-tuning all the weights on the new facts, and c) fine-tuning only entity embeddings associated directly with the facts of new entities. The results are tabulated in Table: 3.

Among the fine-tuning approaches, the model that freezes the relation embedding and fine-tunes only the entities is superior to the other model that fine-tunes the relation embedding too. The gap in performance among them narrows down with relaxed higher hits@k rates. The poor performance of the later model despite tuning the relation embeddings can be attributed to the reason that other entities embeddings may no more be re-

Model	MRR	MR	Hits@10	Hits@3	Hits@1
Fine-tuning TransE	0.2993	1872	0.5449	**0.4801**	0.1026
Fine-tuning TransE - static relation embeddings	**0.3727**	1387	**0.5676**	**0.5148**	0.2152
Skyline model (Training from scratch)	0.3959	1244	0.5678	0.4783	0.2815

Table 3: Evaluation results on TransE with different settings.

lated to each other by the updated translation vectors. Thus by preserving/fixing the relation vectors, the first model outperforms the later though it takes a little longer to converge, as shown in the convergence plots, Figure: 1.

In comparison to the skyline, our best-performing fine-tuning model significantly outperforms the skyline on Hits@3 with very less compute time. Our model achieves comparable performance on Hits@10 and MRR metric while faring poorly on the stricter Hits@1. The time taken for fine-tuning the pre-trained models was ***13 minutes for 1,000 epochs on GPU,*** *which is considerably faster than the time taken for* ***training the model from scratch for the same number of epochs (56 minutes)*** *for nearly equivalent results.*

6 Conclusions

- We formulated the problem of updating KG embeddings with new facts as an Incremental learning problem and suggested two paradigms of solution approaches to solve this problem, i.e., fine-tuning and meta-learning.
- We experimentally showed the effectiveness of incrementally learning the embeddings to be computationally efficient to learn from scratch without comprising heavily on the performance.
- We observed that fine-tuning only the associated old entities and keeping the relation embeddings intact is necessary to avoid forgetting old facts.
- Though our current attempts at solving with meta-learning were not successful, we believe this is the right way to solve this problem, primarily to obtain model-agnostic solutions.
- Apart from the clearly visible scope for performance improvement (hits@1) for the setup with only past graph state (embedding), it is also a crucial problem to model the temporal nature of the evolution of facts over multiple timesteps and also extend the model to incorporate new relations.

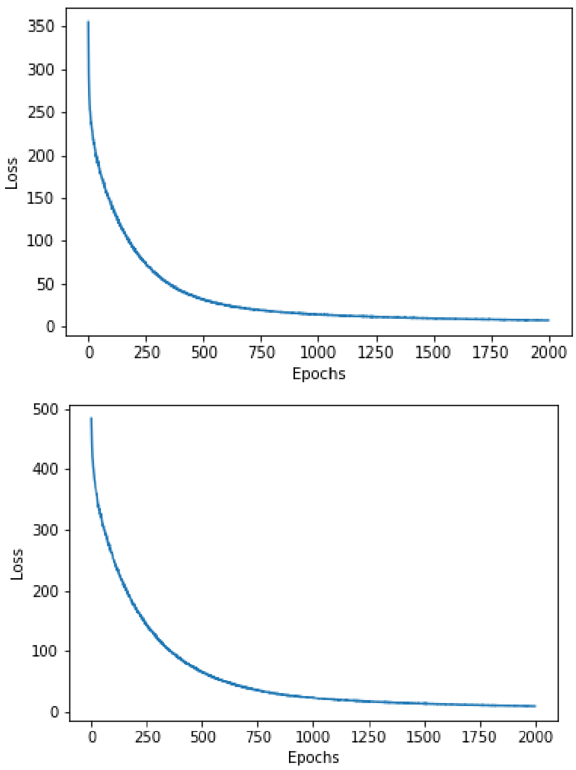

Figure 1: Loss plot for fine-tuning pre-trained model a) without frozen weights, b)with frozen weights.

7 STATEMENT OF CONTRIBUTIONS

This project is a collaborative effort by all three team members. The work was evenly distributed and completed with utmost sincerity and honesty. Each member contributed equally throughout the project.

References

Nobuhiro Kaji et al. Incremental skip-gram model with negative sampling. In *EMNLP 2017*. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS 2013*.

Zhiyuan Chen and Bing Liu. 2016. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 139–144.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. 2019. Relational representation learning for dynamic (knowledge) graphs: A survey. *CoRR*, abs/1905.11485.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *AAAI 2019*.