Natural Language Incorporation in Knowledge Graph Completion

Piyush Chawla

[1] Smart Data Analytics, Universität Bonn, Germany
piyush.bphc@gmail.com
[2] Birla Institute of Technology & Science, Pilani - Hyderabad, India
f20140120@hyderabad.bits-pilani.ac.in

Abstract

Knowledge Graph (KG) research has increased meteorically throughout the last decade. Google's integration of KG to their search engine has sparked research in areas like KG-Completion and KG-Enrinchment in semantic web and machine learning community. In the recent years many state-of-the-art models have been proposed to predict missing facts in KGs (KG-Completion). One class of such models is embedding models which map the KG entities to a low-dimensional vector space and use these representations to predict the new facts. Even though these models acheive good results in predicting unseen facts, they suffer from the out-of-vocabulary problem. A KG completion model can predict new facts only for the entities and relations present in its vocabulary, that is the entities and relations present in the KG triples. This work proposes a simple KG embedding learning approach which tries to tackle the out-of-vocabulary problem by adding the natural language knowledge from word embedding models like word2vec to the entity embedding vectors.

1 Introduction

This work aims to learn low-dimensional vector-space representations for entities/resources in Knowledge Graphs, and at the same time attempts to eliminate the out-of-vocabulary resource problem in latent embedding based Knowledge Graph completion models. Knowledge Graph research has increased meteorically throughout the last decade. In recent years many approaches have been proposed to predict the missing facts in KGs. One such scalable model is TransE (Bordes et al.) [15]. TransE predicts the new facts about KG by learning latent embeddings for the entities and relations. Even though these models have achieved good results in the predicting unseen relationships between entities, they suffer from the out-of-vocabulary problem. A KG-completion model can predict new facts only for the entities that are present in the KG triples.

In other words, any standard KG completion model learns the entities and relations representations only by utilizing the graph structure (nodes and edges). For a triple $(entity1, relation, entity2)$, fed to KG completion model the model produces embeddings for $entity1$, $relation$ and $entity2$ such that they conform to the loss function. The model does not differentiate (is not capable of differentiating) between two entities/relations based on the label that they hold. For instance, even though 'height' and 'elevation' might represent same relation, the model might assign embeddings that are very far from each other in the vector space.

To overcome this issue this work proposes a new model, nlTransE, which uses the word embeddings generated by models like word2vec (Mikolov et al.) [16] to add natural language knowledge in the vanilla TransE model. The models achieves this by forcing TransE model to learn embeddings such that they are as close as possible to the word embeddings of entity labels. This idea is very similar to the alignment model proposed by Chen et al. [6] in their model MTransE. The new model now learns similar embeddings for the entities whose labels have similar meaning in the natural language that is 'height' and 'elevation' will now have close embeddings. The word2vec embedding will now be a good approximation for an OOV resource.

The main objective of this project is to predict the embeddings for entities which are not contained in the KG, in other words, those entities which the model is oblivious to. This is done by using word embeddings from a pretrained module word2vec. The out-of-vocabulary scope of the model (KG model) can always be extended by using a word embedding generation model with richer vocabulary or training the word embedding module with a bigger dataset.

2 Literature Review

Wang et al. [3] propose a novel model for embedding a large scale knowledge graph composed of entities and predicate in a vector space. They discuss mapping properties of relations which must be considered in embedding, namely reflexive, one-to-many, many-to-one, many-to-many. TransE does not perform well dealing with these properties. The new model, TransH, makes a trade-off between model capacity and efficiency and models a relation as hyperplane together with translation operation on it. They use a one-to-many and many-to-one mapping property of a relation to reduce the possibility of false negative labelling (negative data generation).

[7] proposes yet another KG completion model, TransR, to build entity and relation embeddings in a separate entity and relation space. An entity may have multiple properties and the predicates may focus on different aspects of the entities, which renders common continuous vector space insufficient for such kind of modelling. This is a limitation of models like TransE and TransH. The model TransR learns relation embeddings in different vector spaces. Afterwards, they learn embeddings by first projecting entities from entity space to corresponding relation space and then making translation between projected entities

Nickel et al. give an overview of the knowledge graph completion algorithms in their work [8]. They describe the relational machine learning problem of statistical analysis of relational and graph-structured data. The paper reviews statistical models for learning new facts (entity/resource links) about the data. They mainly talk about two broad categories of models , namely latent feature model (like tensor factorization) and multiway neural networks. And the second type is mining observable features in the graph structure. This work was the starting point of this thesis since, it succinctly describes the state-of-the-art (at that time) KG completion model.

Kristiadi et al. discuss the incorporation of literal data in knowledge graph completion in their work [10]. Literals are the non-URI (unformatted) data present in knowledge bases. Most KG completion techniques filter out the literal data in the pre-processing steps. This results in loss of important and interesting properties/information that the literals have. The traditional knowledge graph training algorithms focus on the relations between the entities. To add literal information they come up with LiteralE which is an extension to the latent feature models in general (can be used extend the popular embedding models like transE, DistMult etc.). The experiments show an increase in the link prediction accuracy upon addition of these literal entities knowledge. Toutantova et al. [2015] [11] and Tu et al. [2017] [12] attempt the literal incorporation problem by employing the textual literal data along with the relation embedding. Their model learns additional embedding from the textual data and then add that to the score function in the latent feature learning models.

[13] Xie et al. project entity image features into an entity embedding space and thuse incorporate image literals in the model. Though these models use additional knowledge in the literals, they do not tackle the OOV problem. Pezeshkpour et al. extends the DistMult model

(MultiModal) to predict likeliness of triples (subject, predicate, object) by using the literal embedding instead of object embedding in the vanilla DistMult. This adds literals to the entity embeddings. In contrast, LiteralE combines the literals into the entity embedding directly and thus no additional task is required. In contrast to these approaches of adding natural language knowledge to KG embeddings, nlTransE's ultimate goal is to predict a KG embedding (close approximation) for entities which are not present in the knowledge graph vocabulary.

3 Model

Since nlTransE is an extension of TransE [15] model it is necessary to briefly outline the idea and mathematics behind it. TransE is a simple, but yet powerful, model for knowledge graph completion. The entities and relations are mapped into a vector space of dimension N. For every entity $e \epsilon E$ and every relation $r \epsilon R$ the model learns their vector representations $e\prime$ and $r\prime$. Multi-relational data is mapped into low-dimensional vector space. Even though this model is simplistic we would be remiss to ignore the results that it achieves on benchmark datasets.

The model treats relations as translations operating on the entity vectors. If a triple $[s; p; o]$ exists in the KG, the model brings the embedding of object as close as possible to the subject embedding plus the relation embedding. This can be seen in the diagram below.

The loss function, 1, tries to do achieve the same goal, i.e. it brings s + p closer to o. Where subject, relation and object (s,p and o) are the corresponding embedding vectors. Stochastic gradient descent (mini batches) is used for the optimization and the additional constraints that the L2 norm of embeddings of entities is 1 are added as proposed by Bordes et al. 2013.

$$loss = |s + p - o| \tag{1}$$

$$\zeta = \Sigma_{(h,l,t)\epsilon S} \Sigma_{(h',l',t')\epsilon S'_{(h,l,t)}} [\lambda + d(h + l, t) - d(h^{'} + l^{'}, t^{'})] \tag{2}$$

$$S^{'}_{(h,l,t)} = (h^{'}, l, t) | h\epsilon E \quad \cup (h, l, t^{'}) | h\epsilon E \tag{3}$$

The new model, nlTransE, is implemented by extending the vanilla TransE model and adding the new loss functions. 4 and 5 show the loss functions that are used in the current version of nlTransE model. 4 uses multiplication of transE embedding and word2vec embedding for the entity as the loss function, on the other hand 5 uses L2 distance between the these embeddings. λ is the weight added to control effect of word2vec embeddings on nlTransE. If $\lambda = 0$, the model becomes vanilla TransE.

$$loss = |s + p - o| + \lambda(multiply(s, s_{w2v}) + mutliply(o, o_{w2v})) \tag{4}$$

$$loss = |s + p - o| + \lambda(L2(s, s_{w2v}) + L2(o, o_{w2v})) \tag{5}$$

4 Experiments

FB15k (Bordes et al.), a widely used subset of freebase [2], was used as a test case. To test the capabilities of nlTransE model this dataset was trained, and the link prediction was tested using hits@N. The dataset consists of 14,541 entities, 1,345 relations and 592,213 triples. Entities are in the form of corresponding MIDs which look like "/m/xyz", and the relations are human readable IDs like "/common/webpage/category". In order to get the labels for all entities, "freebase links", a subset from DBpedia [1], was used. This subset contains mappings between freebase and DBpedia entities. For instance "dbr:CentralIndianForests" is mapped to /m/0cnkvs". After mapping all the freebase entities to DBpedia entities their labels were used to compute word embeddings. For instance the word embeddings for "dbr:CentralIndianForests" was computed as $(wordembedding(Central) + wordembedding(Indian) + wordembedding(Forests)/3)$.

Component Predicted	hits@1	hits@5	hits@10
Relation	0.544	0.849	0.89
Head	0.095	0.292	0.397
Tail	0.148	0.373	0.485

Table 1: Prediction accuracy for vanilla TransE

Once the mappings are done the final dataset contained 1,345 relations, 14,439 entities and 516,874 triples. The test and train data were created by segregating these mapped triples in $3:7$ ratio (30% test and 70% train).

Component Predicted	hits@1	hits@5	hits@10
Relation	0.555	0.8335	0.8835
Head	0.0944	0.290	0.395
Tail	0.1472	0.371	0.4849
Head (w2v)	0.029	0.098	0.146
Tail (w2v)	0.0778	0.174	0.233

Table 2: Prediction accuracy for multiplication loss function, $\lambda = 1$

Word embedding computation was done using a pretrained word2vec model. This model was trained using the gensim library with 10,000 word vocabulary and the embedding size of 100 (which was also size of KG embeddings). Model evaluation was performed using hits@N, which is defined as average number of times the component was correctly predicted. Since the aim of this work is adding label knowledge to the existing latent feature models (TransE in our experiments), the model should perform well on the standard link prediction tasks. These tasks envolve predicting either head (subject), tail (object) or predicate (relation) given the other two. If the knowledge graph contains a triple [Delhi, capital of, India], *relation* prediction means predicting 'capital of' given 'Delhi' and 'India', *tail* prediction means predicting 'India' given 'Delhi' and 'capital of' and *head* prediction means predicting 'Delhi' given 'capital of' and 'India'.

Component Predicted	hits@1	hits@5	hits@10
Relation	0.5554	0.8435	0.883
Head	0.0936	0.290	0.3961
Tail	0.1471	0.371	0.484
Head (w2v)	0.0326	0.102	0.1482
Tail (w2v)	0.0791	0.1785	0.236

Table 3: Prediction accuracy for multiplication loss function, $\lambda = 5$

Apart from these, two more predictions are defined. For the above example $Tail_{w2v}$ prediction is defined as predicting 'India' given 'Delhi' and 'capital of' but using the word2vec embedding for entity 'Delhi'. This tests the out-of-vocabulary resource prediction capability (preliminary test) of the model. If the model can use word2vec embeddings instead of the transE embeddings for fact prediction, then OOV can be tackled. For instance, in the above example if KG doesn't have 'Germany' in its vocabulary, but has 'Berlin' then the fact [Berlin, capital of, Germany] can be predicted using the word2vec embedding for 'Germany', that is , word2vec embeddings will become a close approximation of the out-of-vocabulary resource embeddings (vector space representations).

Component Predicted	hits@1	hits@5	hits@10
Relation	0.5564	0.837	0.878
Head	0.0943	0.2878	0.3932
Tail	0.1471	0.3686	0.4811
Head (w2v)	0.0308	0.1011	0.01462
Tail (w2v)	0.0759	0.1764	0.2312

Table 4: Prediction accuracy for multiplication loss function, $\lambda = 10$

Table 1 contains the link prediction accuracies for the vanilla TransE model. It can be observed that the vanilla transE has highest hits values for almost all the components, which is not a surprise as adding label knowledge to the embeddings increases noise. Table 2 shows the prediction results for all the 5 components using nlTransE model. The model uses multiplication loss function 4 and λ value of 1.

Component Predicted	hits@1	hits@5	hits@10
Relation	0.3219	0.610	0.7117
Head	0.088	0.24	0.3336
Tail	0.1374	0.319	0.420
Head (w2v)	0.0363	0.1047	0.1571
Tail (w2v)	0.0745	0.180	0.242

Table 5: Prediction accuracy for L2 loss function, $\lambda = 1$

Table 3 and table 4 contain the prediction results for nlTransE model using mutliplcation loss function and λ values of 5 and 10 respectively. The last table 5 shows results for nlTransE model using L2 loss function 5. Since the normal relation and entity prediction deteriorated too much in this model, it wasn't further investigated. The last model showed an increase in the $Tail_{w2v}$ prediction, but this increase came at the cost of lower Tail prediction accuracy.

5 Conclusion

This work tries to tackle the out of vocabulary problem in Knowledge Graph completion. Word2vec is used as a natural language integration tool and word embeddings are used to add label information into the KG embeddings. It should be noted that These experiments are just preliminary results on the idea. The model can be extended to other latent embedding models like DistMult etc. The aim of this work is not improving link prediction performance (to improve state-of-the-art) but to predict links to resources which are not even present in the knowledge graph. The future work will contain the complete analysis of this approach.

References

[1] Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z. DBpedia: A Nucleus for a Web of Open Data. In: Aberer K. et al. (eds) The Semantic Web. Lecture Notes in Computer Science, vol 4825. Springer, Berlin, Heidelberg.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge Muhao Chen, Tao Zhou, Pei Zhou, Carlo Zaniolo, Multi-graph Affinity Embeddings for Multilingual Knowledge Graphs, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[3] Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen, Knowledge Graph Embedding by Translating on Hyperplanes, AAAI Publications, Twenty-Eighth AAAI Conference on Artificial Intelligence

[4] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu Learning Entity and Relation Embeddings for Knowledge Graph Completion, AAAI Publications, Twenty-Ninth AAAI Conference on Artificial Intelligence

[5] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sren Auer, Chris Bizer DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia Semantic Web Journal, volume 6, number 2, pages 167-195, IOS Press, 2015

[6] Muhao Chen, Yingtao Tian, Mohan Yang, Carlo Zaniolo. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment

[7] H Zafar, G Napolitano, J Lehmann. Formal Query Generation for Question Answering over Knowledge Bases

[8] Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs

[9] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng. Learning Multi-Relational Semantics Using Neural-Embedding Models

[10] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, Asja Fischer. Incorporating Literals into Knowledge Graph Embeddings.

[11] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury. Representing Text for Joint Embedding of Text and Knowledge Bases

[12] Cunchao Tu, Han Liu, Zhiyuan Liu, Maosong Sun. CANE: Context-Aware Network Embedding for Relation Modeling

[13] Ruobing Xie, Zhiyuan Liu, Tat-seng Chua, Huanbo Luan, and Maosong Sun. Image-embodied knowledge representation learning. arXiv preprint arXiv:1609.07028, 2016

[14] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference, 2014

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran. Translating Embeddings for Modeling Multi-relational Data

[16] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space