Knowledge Graph Embedding for Triples Fact Validation

Alexis Pister, Ghislain Atemezing

MONDECA, 35 boulevard de Strasbourg 75010 Paris, France.
<firstname.lastname@mondeca.com>

Abstract. This poster[1] presents a methodology for designing and implementing a knowledge graph fact checking using graph embeddings models. The implementation has been tested on the dataset of task 1 of the ISWC 2019 challenge to assess the correctness of a statement. We trained 6 embedding models : DistMult, HolE, TransE, TransR, ComplEx and RDF2VEC. Several machine learning algorithms have been tested to classify the triples given their embeddings using a 4-fold cross validation scheme on the entire dataset. The results indicate that RDF2VEC gives the higher AUC score of 0.877 for the prediction of the correctness of the statements. According to the evaluation report obtained from the challenge board, our team's score came third among nine participating teams to the fact validation task 1 challenge.

Keywords: Knowledge Graph embedding, Fact validation, drugs dataset.

1 Introduction

A knowledge graph (KG) is an oriented graph formed of a set of entities corresponding of the vertices of the graph, and a set of relations which consist of edges connecting the entities. A KG aims at representing entities of a given domain with their relations. We saw a rapid growth of this type of data in the recent years, mainly because of the wide possibilities of applications, such as disambiguation or question answering.

Given a knowledge base constituted of positives and negatives statements, the task 1 of the ISWC 2019 challenge [2] aims to give an idea of the correctness of any new statement similar entities. A training set and a testing set both made of 25,000 statements distributed between positives and negatives triples are given. 11,990 drugs and diseases form the entities of the graph, linked by 5 different predicates giving information on the interactions between the drugs and diseases. Some entities existing in true statements are never seen in false statements, and some entities present in the testing graph are not used in the training set. We present in this paper our machine learning approach to tackle the problem.

[2] https://dice-group.github.io/semantic-web-challenge.github.io/

2 Related Work

Because of their symbolic nature, knowledge graph can be hard to manipulate for some computing tasks. A new strategy has been elaborated to bypass this issue, called knowledge graph embedding [1], which gained massive attention in the recent years for its flexibility and various applications [5]. The aim of this type of method is to transform the components of the graph, i.e. the entities and the relations into continuous vector spaces, which describe the topological and semantic relations of the different components with numerical values.

The majority of knowledge graph embedding methods are based on the maximization of a scoring function, for example the dot product of the embeddings of the subject, object and predicate of a given known statement [1]. However, other types of methods such as RDF2VEC [4] takes into account the semantic and topological local environment of every entity of the graph by generating large amount of random walks and then applying the Word2Vec algorithm on the generated walks.

The embeddings can then be used for many tasks such as link prediction, entity classification or triplet classification. Triplet classification consists of classifying new triplets as true or false. However, the public datasets used for this tasks such as WN11 or FB13 do not have any explicitly false triplets [6], and they have to be generated, contrary to the dataset given in this task.

3 Our Approach for Fact Validation

3.1 Knowledge Graph Embedding on RDF Triples

Our approach has been to transpose the semantic and topology of the entire training graph into a vector space to be able to describe any triple by a set of numeric value. It is then possible to apply statistic and machine learning methods to separate the vector space in the purpose of distinguishing true and false statements by their describing vectors.

To have a vector space describing the graph, we used different methods of knowledge graph embedding found in the literature. We tried the following 6 embedding models : DistMult, HolE, TransE, TransR, ComplEx which are based on a scoring function, and RDF2VEC which is based on a Word2Vec model trained on random walks of the graph. The RDF2VEC model was trained in 10 epochs with a window of 8 using the skip-gram scheme, while we used 200 epochs for all the other models. We chose a dimension of 150 for the vector space, and tried as well a 300-length vector model for RDF2VEC [3].

Once we compute the the embeddings for every entity and relation, we express the embedding of a statement ($\langle subject \rangle$ $\langle predicate \rangle$ $\langle object \rangle$) by combining the three given vectors. We then tackle the problem as a binary classification task,

[3] We used the python library Ampligraph [2] for the implementation of DistMult, HolE, TransE and Complex, pykeen for transR and the official implementation of RDF2VEC

by training different machine learning algorithms on the training dataset, trying to separate true and false statements by their corresponding embeddings in the vector space.

To find the best embedding model and machine learning classifier, we split the released training set into a training and a validation (or development) set with a 0.75/0.25 split. The embedding models were trained on this entire dataset available, while the machine learning models were trained on the training set and tested and optimized on the validation set.

Once the machine learning models were trained, we applied them on the validation set constituted of new true and false statements, previously unseen by the model. A truth score was then given for each statement as the output of the classification.

3.2 Machine Learning Classifiers

We tested two machine learning algorithms to classify the triples given their embeddings for the 6 models : MultiLayerPerceptron (MLP) and RandomForest (RF). The MLP followed a gridsearch parameters optimization for the hidden layers shape, the optimization technique and the hidden layer activation function. The best parameters were 2 hidden layers of size 300 and 100, the ADAM optimization strategy and a ReLU activation function. The models have first been trained on the training set and applied on the development set. They have been implemented with scikit-learn library in Python.

The results are presented in Table 1. The RDF2VEC embedding gives the best results with MLP model, with an AUC score of 0.877 for the prediction regarding the correctness of the development dataset statements.

AUC score	RandomForest	MultiLayerPerceptron
HolE	0.856	0.827
ComplEx	0.855	0.825
TransE	0.848	0.823
DistMult	0.857	0.821
TransR	0.831	0.754
RDF2VEC	0.869	**0.877**

Table 1. AUC scores on the development dataset for the 6 embedding models

4 Evaluation

We used the embedding model which gave the best AUC score in the development data set, i.e. RDF2VEC, and tested it on the test dataset with the same parameters. We trained this model on all the data available, that is the training, validation and test sets, to have an embedding vector for each entity and relation present in the test set. We then trained the machine learning model which gave the best score in the development process, i.e. an MLP, on the triplets embeddings of the training and validation sets and applied it on the embeddings of the test triplets to classify them as true or false.

Our approach has been submitted[4] using the GERBIL interface provided by the organizers of the challenge. The evaluation given by the organizers using GERBIL shows a value of 0.9979 corresponding to AUC.

At the time of writing this paper, our team represented with the nickname "Mdk Team" was ranked third out of nine participants in the leaderboards of the challenge. The AUC difference with the winner team is 0.0018. As four systems have a score higher than 0.9900, we can

5 Conclusion and Future Work

This poster proposes a fact validation workflow for the task 1 of the ISWC 2019 challenge. The presented method consists of the training of an knowledge graph embedding model on the entire dataset, whose inputs are used for machine learning models. Based on our implementation and experiments, RDF2VEC which is based on the generation of random walks outperform all the scoring function based embedding models. As four systems have a score higher than 0.9900 in the competition, we can question the difficulty of this challenge. our system should thus be tested on other datasets. Moreover, the main drawback of the current model is that it can not take as input previously unseen entity or relation to map them to the vector space. New approaches such as graph neural networks could overcome this issue because they directly classify subsets of a graph and seem to give good results on graph classification tasks in the literature [3].

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. L. Costabello, S. Pai, C. L. Van, R. McGrath, and N. McCarthy. AmpliGraph: a Library for Representation Learning on Knowledge Graphs, Mar. 2019.
3. T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
4. P. Ristoski and H. Paulheim. Rdf2vec: Rdf graph embeddings for data mining. pages 498–514, 10 2016.
5. Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
6. Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

[4] http://w3id.org/gerbil/kbc/experiment?id=201906030011