Fact Validation with Knowledge Graph Embeddings

Ammar Ammar[1] and Remzi Çelebi[2][0000−0001−7769−4272]

[1] Maastricht Centre for Systems Biology, Maastricht University, Maastricht, Netherlands a.ammar@student.maastrichtuniversity.nl
[2] Institute of Data Science, Maastricht University, Maastricht, Netherlands remzi.celebi@maastrichtuniversity.nl

Abstract. Fact validation in a knowledge graph is a task to determine whether a given fact (subject, predicate, object) should appear in the knowledge graph. In this paper, we have described our approach for the fact validation task in the context of the Semantic Web Challenge 2019. We used embedding features with machine learning to predict facts that were missing from the knowledge graph. The embedding features were generated applying a knowledge graph method known as the RDF2Vec method on the knowledge graph with only positive statements. To improve our machine learning model, we added the test facts that we could validate via the public sources into the positive knowledge graph. We trained a Random Forest classifier on the training data (positive and negative statements) plus the verified test statements and made predictions for test data.

Keywords: Fact validation · Fact checking · Knowledge Graph Embedding.

1 Introduction

Knowledge graphs are currently among the most prominent implementations of Semantic Web technologies. The task of fact checking in knowledge graphs, which is to decide whether a fact t is missing from a given a knowledge graph G, is among the cornerstones of knowledge base management. The verified facts can be used to (1) incomplete knowledge graphs refining (2) knowledge graphs violation detection, (3) improve the quality of knowledge search, and (4) multiple knowledge graphs integration [1]. This year, the International Semantic Web Conference revealed a dataset for the fact validation challenge in the context of the Semantic Web Challenge. The challenge task is to assess the correctness of a given statement about drugs, diseases, products. The challenge participants are asked to assign a trust score for each of the statements with (i.e., a numerical

value between 0 and 1), where 0 means that they are sure that the statement is false and 1 means that they are sure the statement is true. We propose a machine learning model using embedding features for this challenge, to predict if a given statement is true or not (i.e. validate the correctness of the statement).

2 Dataset

The core dataset consists of a graph of entities (drugs, diseases and products) and information linking these entities. The dataset is created by extracting information from a well-known source and identifying links between entities. The dataset contains both training and test set parts in which the training data was made available for building a system to make prediction on test data. Both the training and testing sets consist of 25k examples with positive and negative statements, equally distributed among each of the following five properties:

- http://dice-research.org/ontology/drugbank/interactsWith
- http://dice-research.org/ontology/drugbank/hasCommonIndication
- http://dice-research.org/ontology/drugbank/hasSameState
- http://dice-research.org/ontology/drugbank/hasIndication
- http://dice-research.org/ontology/drugbank/hasCommonProducer

While the challenge organizers generated positive statements by identifying the entities for which the proposed properties hold, they generated the negative statements by replacing the entities in the positive statements such that the generated triples are false or invalid.

3 Methods and Results

We used embedding features to train our machine learning model to predict trust scores for test facts as described in Figure 1. In order to learn embedding features, the training KG, given in the form of reified statements, was converted to the positive explicit statements (eg. $< drug, hasIndication, disease >$). We trained our classifier on the training data (positive and negative statements) plus the verified test statements and made predictions for test data.

We added the verified facts to the positive knowledge graph to generate a better embedding feature vector. We used the external resource "DrugBank" to check if the DrugBank relations might hold for the test facts. This can be seen as the enrichment of the training knowledge graph. While enriching with the DrugBank, only the properties that are relevant with the challenge dataset were included. The properties extracted from Drugbank XML (v5.1.1) are:

- indication
- state
- drug-interactions
- packagers

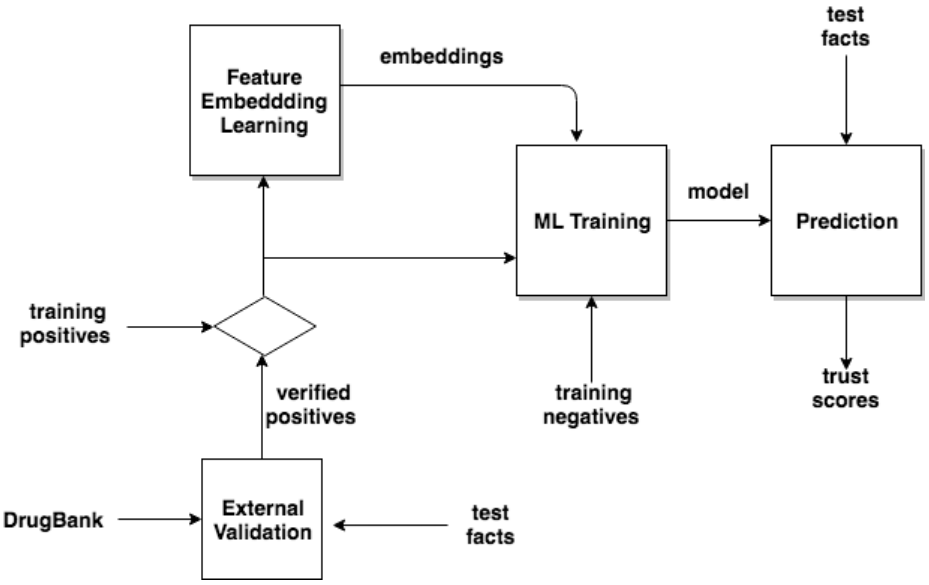

Fig. 1. The overview of our approach for fact validation.

To validate test facts, we link the challenge data to the Drugbank dataset by mapping the drug and disease entities to DrugBank drugs and Human Disease Ontology (DO) diseases respectively. The challenge use the same unique identifiers with Drugbank for drugs. In order to link the challenge disease entities, each text in the indication section in Drugbank was annotated with Human Disease Ontology using BioPortal API.

After disease annotations were obtained for both training and test datasets drugs, the normalized Levenshtein similarity was computed between disease names to match the annotated diseases (DO) with the challenges diseases. A test fact will be considered as verified if the proposed relation between the identified entities holds in the DrugBank.

For feature learning, we need a proper representation of the entities in the knowledge graph that reflects their features. Here, we used the RDF2Vec approach [2] in which the "Random Walks" algorithm was used to generate certain number of walks for each entity and for a specific depth. The parameter used for random walks are: number of walks of 200 for each entity with a depth of 5. We learned from a previous study [3] that the value of 5 for the random walk depth parameter for knowledge graph learning gives the best results. Next, the random walks were fed into another algorithm "Word2Vec" where it is split into terms (i.e. subjects, predicates and objects) and then the Word2Vec algorithm was applied using the "CBOW" neural networks with a layer size of 200 and the graph embeddings were generated. We have noticed that the dimension of the features vector is not critical if it is in the range of 100-500 from our experiments.

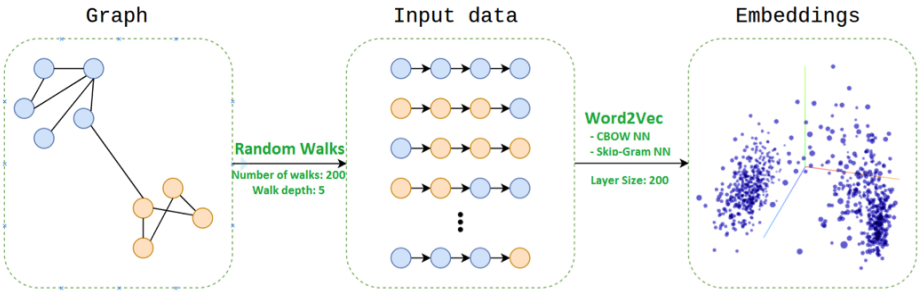

Fig. 2. The workflow of generating embeddings.

The Fig. 2 shows the workflow of generating the embeddings. After that, a Random Forest classifier was trained on the training set plus the verified statements. A number of estimators of 200 was used for the Random Forest classifier and the remaining parameters were left as default. To represent the feature vector of a statement, we concatenated embedding vectors of the subject and object entities, and a numerical value encoding predicates between the entities. The final model was used to predict a probability for each statement in the test set. The predictions were submitted to the challenge website and reported an AUC of 0.99971976. The reported AUC was the highest obtained score among all the participants in the challenge as provided through the leader board on the challenge website [3]. The same method was also applied on the original dataset without performing the enrichment using Drugbank and reported an AUC of 0.9926. From these results, a conclusion can be drawn that the major contributor to the performance is the proposed embedding-based method over enriching the data with an external resource.

Acknowledgments This work was supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/3454-01-01

References

1. Lin, P. : act Checking in Knowledge Graphs with Ontological Subgraph Patterns. Data Science and Engineering (2018)
2. Ristoski, P. and Paulheim, H., 2016, October. Rdf2vec: Rdf graph embeddings for data mining. In International Semantic Web Conference (pp. 498-514). Springer, Cham.
3. Celebi, R and Yasar, E and Uyar, G H and Gumus, O and Dikenelli, O and Dumontier, M, 2018. Evaluation of Knowledge Graph Embedding Approaches for Drug-Drug Interaction Prediction using Linked Open Data. Semantic Web Applications and Tools for Healthcare and Life Sciences.

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