Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 345–354

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

# Learning Entity and Relation Embeddings for Knowledge Resolution

Hailun Lin, Yong Liu*, Weiping Wang, Yinliang Yue, and Zheng Lin

Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
linhailun@iie.ac.cn

**Abstract**

Knowledge resolution is the task of clustering knowledge mentions, e.g., entity and relation mentions into several disjoint groups with each group representing a unique entity or relation. Such resolution is a central step in constructing high-quality knowledge graph from unstructured text. Previous research has tackled this problem by making use of various textual and structural features from a semantic dictionary or a knowledge graph. This may lead to poor performance on knowledge mentions with poor or not well-known contexts. In addition, it is also limited by the coverage of the semantic dictionary or knowledge graph. In this work, we propose ETransR, a method which automatically learns entity and relation feature representations in continuous vector spaces, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution. Experimental results on two benchmark datasets show that our proposed method delivers significant improvements compared with the state-of-the-art baselines on the task of knowledge resolution.


Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

*Keywords:* knowledge graph, knowledge resolution, knowledge representation, entity embedding, relation embedding

## 1 Introduction

Access to an organized knowledge graph is critical for many real-world tasks, such as query suggestion and question answering. Most real-world information is unstructured, interconnected, noisy, and often expressed in the form of text. This inspires constructing an organized knowledge graph from the large volume of noisy text data. A large number of knowledge graphs have been constructed, such as Freebase [1], Knowledge Valut [8], YAGO [9], Probase [18]. An important component in constructing knowledge graphs is knowledge resolution. Given the knowledge mentions (e.g., entity mentions and relation mentions) in unstructured text data, the goal of knowledge resolution is to cluster knowledge mentions into disjoint groups with each group representing a unique knowledge.

*Corresponding author: Yong Liu (liuyong@iie.ac.cn)

1877-0509 
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.045

The knowledge resolution task is challenging due to the fact that many knowledge mentions are ambiguous: the same mention can refer to various different real world entities or relations when they appear in different contexts, and many knowledge has various mention forms. Knowledge resolution plays a critical role in high-quality knowledge graph construction. When knowledge extracted from text data is ready to be inserted into the knowledge graph, it is necessary to know which real world knowledge this piece of information should be associated with. If a wrong decision is made here, the knowledge graph will not only lose some information, but also introduce errors.

Traditional approach to knowledge resolution usually makes use of various textual and structural features from a semantic dictionary or a knowledge graph (e.g., [5], [16]), which may lead to poor performance on knowledge mentions with poor not well-known contexts. Moreover, its performance is also limited by the coverage of the semantic dictionary or knowledge graph. Recently, a promising approach for the task is embedding knowledge into a continuous vector space to learn feature representations, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution. Following this approach, many methods have been explored, which will be introduced in detail in Section 2. Among these methods, some notable works, including TransE [3], TransH [17], TransR [11] are effective and efficient. TransE represents entities as points and relations as translations from head entities to tail entities in a vector space. TransH models a relation as a hyperplane together with a translation operation on it. TransR models entity and relation embeddings in separate vector spaces, which are bridged by relation-specific matrices. These methods mainly focus on modeling single knowledge in continuous space, which ignore modeling the semantic relatedness between knowledges.

In order to learn better knowledge representations to model the complicated semantic correlations between knowledge triples, we propose a new model, named ETransR. ETransR is the extension form of TransR, which models entities and relations in distinct vector spaces, i.e., entity space and multiple relation spaces, and performs translation in the corresponding relation space. The basic idea of ETransR is illustrated in Figure 1. In ETransR, for each knowledge tripe $(h, r, t)$, it first embeds entities in entity space, relations in relation space. And then it projects entity embeddings into relation space by relation-specific matrix $M_r$, which can obtain $\mathbf{h}_r+\mathbf{r} \approx \mathbf{t}_r$. The model can make the head/tail entities that actually hold the relation (denoted as circles) close with each other, and get far away from those that do not hold the relation (denoted as triangles).

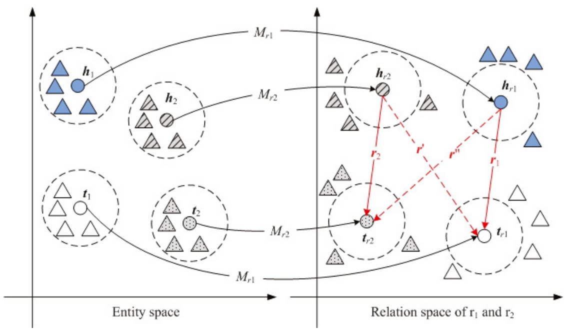


Figure 1: The illustration of ETransR

Given two knowledge triples $(h_1, r_1, t_1)$ and $(h_2, r_2, t_2)$, if they are two mention forms of a unique knowledge, then they will satisfy $\mathbf{t}_{r_1}$- $\mathbf{h}_{r_1} \approx \mathbf{t}_{r_2}$- $\mathbf{h}_{r_2}$. For head entities, if $h_1$ is the same as $h_2$, it will have $\mathbf{h}_{r_2}+ \mathbf{r}_1 \approx \mathbf{t}_{r_1}$, $\mathbf{h}_{r_1}+ \mathbf{r}_2 \approx \mathbf{t}_{r_2}$. And for tail entities, if $t_1$ is the same as $t_2$, it