Translating Embeddings for Modeling Multi-relational Data

Bordes et al., Neurips 2013

Presented by Nishant Mishra(260903177)

Outline

- INTRODUCTION
- METHODOLOGY
- EVALUATION AND RESULTS
- DISCUSSION

INTRODUCTION

CONTEXT/BACKGROUND

Knowledge Graphs are directed multi-relational graph which represent facts using entitities and different relations between them

KGs have massive real world applications such as information retrieval, semantic parsing, social network analysis etc. E.g. Freebase, Dbpedia, wordnet.

Each edge is of the form of a triplet (head,label,tail), also called a fact, which indicates that there exists a relationship of name label between the entities head and tail.

INTRODUCTION

PROBLEM/MOTIVATION

This paper deals with the problem of Knowledge Graph Embedding.

KGs are are symbolic in nature and hence it is difficult to manipulate them and perform computation.

The key idea is to embed components of a KG into continuous vector spaces, to simplify the manipulation while preserving the inherent structure of the KG.

This facilitates their use in a number of downstream tasks such as KG completion, node classification etc.

INTRODUCTION

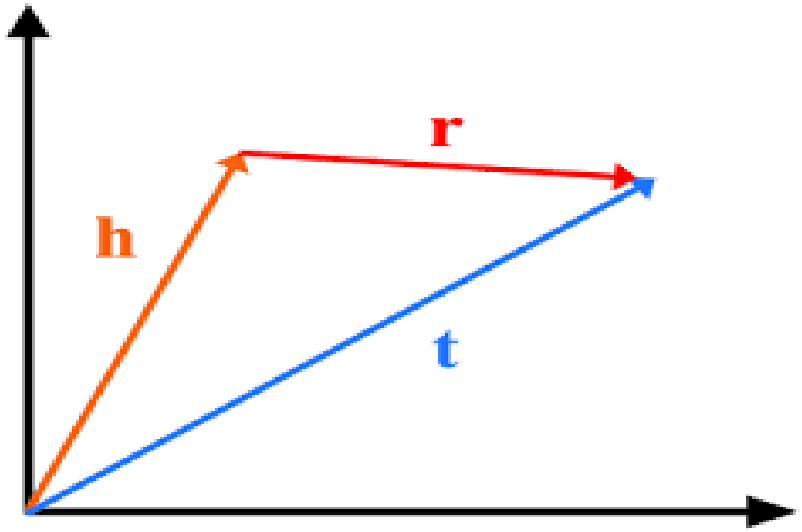
What's TransE?

The paper proposes the Translating Embeddings(transE), a method to solve this problem

TransE models relation embeddings by interpreting them as translations operating on the low dimensional entities embeddings.

The basic notion is highly intuitive as it can easily be represented geometrically in the form of vector translations.

INTRODUCTION



If the fact(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

INTRODUCTION

Why TransE?

Translations are natural transformations for modeling hierarchical data, which are common in KGs.

Light parametrization leads to better scalability allowing training on large amount of data.

Also takes inspiration from Mikolov et al's seminal paper word2vec

METHODOLOGY

This is represented by the following margin based ranking criteria, used as the objective function for training

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

Where d is a dissimilarity measure (L1 or L2 norm), γ >0 is a margin hyperparameter.

The corrupted triplets were constructed using the below equation, with either head or tail replaced from original training triplets

$$S'_{(h,\ell,t)} = \left\{(h',\ell,t)|h' \in E\right\} \cup \left\{(h,\ell,t')|t' \in E\right\}.$$

METHODOLOGY

The entities are normalised in order to ensure that the training process does not trivially minimize L
The optimization is carried out using a mini batch gradient descent update

The algorithm is stopped based on its performance on a validation set.

For a given entity, its embedding vector is the same when the entity appears as the head or as the tail of a triplet.

METHODOLOGY

TRAINING

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

$\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$

$\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

$\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

$S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

$T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

for $(h, \ell, t) \in S_{batch}$ **do**

$(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

$T_{batch} \leftarrow T_{batch} \cup \left\{ \left((h, \ell, t), (h', \ell, t') \right) \right\}$

end for

Update embeddings w.r.t. $\sum_{\left((h,\ell,t),(h',\ell,t')\right) \in T_{batch}} \nabla \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'}) \right]_+$

end loop

EVALUATION

The evaluation task was basically Link prediction i.e predicting a given entity that is related to another entity by a particular relation.

For each test triplet, the head was replaced with all the entities in the dictionary. Dissimilarities (or energies) of those corrupted triplets are first computed by the model and then sorted by ascending order; the rank of the correct entity is finally stored.

This was repeated by replacing the tails and mean of the ranks of correct triplet was stored.

Another metric apart from mean rank was the hits@10 which basically denotes the number of times the correct triplet was among the top 10 ranked ones.

EVALUATION

TransE, is evaluated on data extracted from Wordnet and Freebase15 against several recent methods from the literature which were shown to achieve the best current performance.

Additionally a separate large dataset was created called FB1M to test scalability

The hyperparameters were learning rate, the margin(γ), the embedding dimensions(20 or 50), and dissimilarity measure(L1 or L2)

EVALUATION

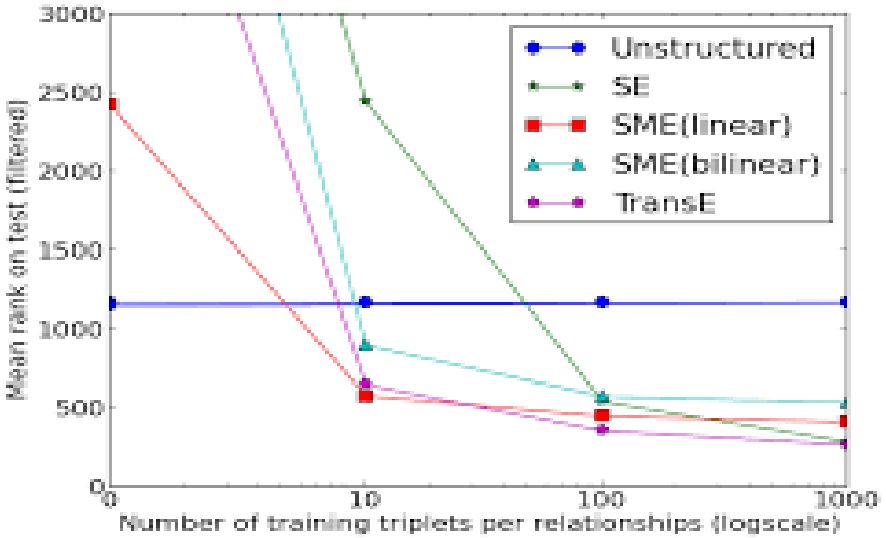
Table 3: **Link prediction results.** Test performance of the different methods.

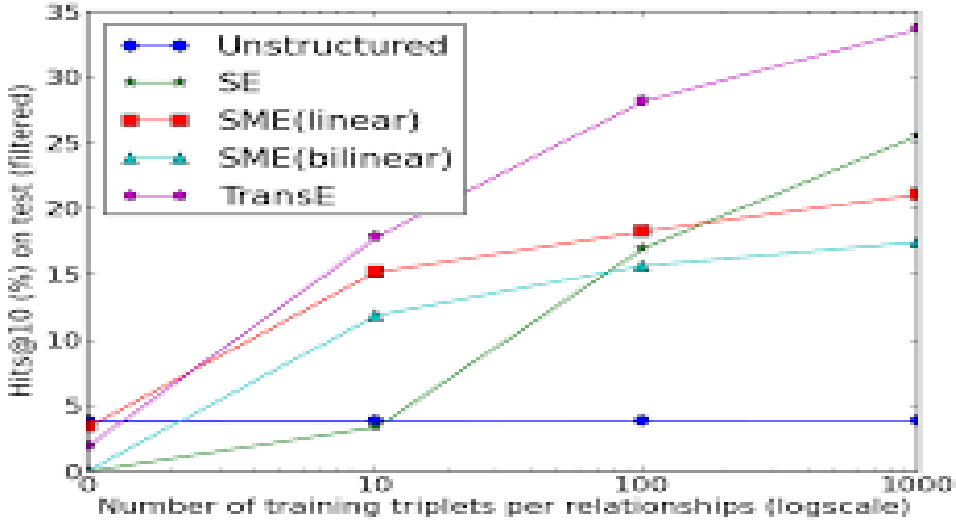
DATASET	WN				FB15K				FB1M	
METRIC	MEAN RANK		HITS@10 (%)		MEAN RANK		HITS@10 (%)		MEAN RANK	HITS@10 (%)
Eval. setting	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Raw*
Unstructured [2]	315	304	35.3	38.2	1,074	979	4.5	6.3	15,139	2.9
RESCAL [11]	1,180	1,163	37.2	52.8	828	683	28.4	44.1	-	-
SE [3]	1,011	985	68.5	80.5	273	162	28.8	39.8	22,044	17.5
SME(LINEAR) [2]	545	533	65.1	74.1	274	154	30.7	40.8	-	-
SME(BILINEAR) [2]	526	509	54.7	61.3	284	158	31.3	41.3	-	-
LFM [6]	469	456	71.4	81.6	283	164	26.0	33.1	-	-
TransE	**263**	**251**	**75.4**	**89.2**	**243**	**125**	**34.9**	**47.1**	**14,615**	**34.0**

TransE, outperforms all counterparts on all metrics, usually with a wide margin, and reaches some promising absolute performance scores

EVALUATION

A further test of generality was also conducted to see how well the different methods learn embeddings of new unseen relations.





As can be seen TransE was the fastest learner requiring the least number of training triplets with the new relations to converge to better rank.

CONCLUSION AND MISC

The good performance of TransE is due to an appropriate design of the model according to the data, but also to its relative simplicity. This means that it can be optimized efficiently with stochastic gradient unlike other models which underfit.

It is highly scalable.

It fails in modeling data where 3-way dependencies between h, l and t are crucial.

TransE does not make use of any additional information such as free text, description, entity type, only based on observed facts, which if implemented can improve the performance. E.g TKRL, DKRL

Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to-N, N-to-1, and N-to-N relations, as it learns similar embeddings to different entities. This has been overcome to some extent in TransH(relation specific entity embeddings), TransR, ManifoldE etc