Jointly Embedding Knowledge Graph and Word Vectors

Tomu Kadoki^{a,*}, Runhe Huang^b, Satrou Fujita^b

^aGraduate School of CIS, Hosei University, 3-7-2 Kajino-cho, Koganei, Tokyo 184-8584 Japan

^bFaculty of CIS, Hosei University, 3-7-2 Kajino-cho, Koganei, Tokyo 184-8584 Japan

*Corresponding Author: tomu.kadoki.6e@stu.hosei.ac.jp

Abstract

Knowledge graph embedding is a model for embedding components of a knowledge graph into a low-dimensional vector space. This model can improve the capability of relational calculation used in search engines. Traditional approaches to knowledge graph embedding model could hardly represent complex relationships among entities and their meanings, because there is no direct way to represent the degree of similarity between entities and between relations. In this paper, we propose a Word-Knowledge model (WK-model) that embeds similarity measures of word vectors into the knowledge graph embedding space. We compared WK-model with TransE, which is a pioneer among the existing knowledge graph embedding models, in terms of the capability of calculating in the knowledge graph and similarities between the embedding model and representations in the word vector space. In particular, we tested the ability of the two methods to embed similarities of word vectors into the knowledge graph space and achieve a balanced performance in entity prediction tasks.

Keywords: Knowledge graph, Knowledge embedding model, Word vector, Entity prediction, Word similarity.

1. Introduction

Enhancing the capability of search engines can greatly contribute to the improvement of user experience when interacting with the Internet. A typical search engine by TF-IDF generates search results by calculating the occurrences and matching between the words in the user query and documents published on the Internet using the term frequency-inverse document frequency method. Google has recently introduced a search engine mechanism that can generate search results considering also the meaning and context of searched words. This mechanism makes the search engine displaying related contents and information of

the word.

One of the approaches for extending the search engine capability is a knowledge graph embedding model. For example, TransE is a knowledge embedding model based on vector operations performed over a knowledge graph by embedding relationships of knowledge into a vector space⁽¹⁾. Several extensions of TransE have been proposed, including TransH⁽²⁾ and TransR⁽³⁾. TransH maps the multidimensional vector space of entity relationships to hyperplanes in lower dimensions to deal with multiple meanings of the relationships⁽²⁾. TransR projects the vector space of entity relationships into another vector space to express a different abstraction of the relationships⁽³⁾.

Traditional knowledge graph embedding models focus on establishing adequate relations between concepts; thus, these models cannot directly represent similarities and distances between concepts. For instance, the degree of similarity between two concepts cannot always be embedded to build the vector space as expected because a single relation cannot express diverging meanings and different levels of abstraction. If there were a way to directly embed the similarity between concepts into a knowledge graph, the vector space could be structured so it could deal with multiple meanings of the relations among the concepts.

This paper proposes a model for structuring knowledge graphs by representing the degrees of similarities among entities in word vectors. This model minimizes the loss functions of knowledge and similarity as defined below to improve the knowledge graph structure. Section 2 discusses recent studies related to knowledge graphs. Section 3 describes the proposed method called a Word-Knowledge model (WK-model) in detail. Section 4 outlines experiments and their results, and Section 5 discusses them. Finally, Section 6 concludes the paper.

2. Related Work

2.1 Knowledge Graph

A knowledge graph is a directed graph that represents the relationship between objects in the real world, where a real-world object is called an “entity” and a relationship between entities is called a “relation.” A piece of knowledge is represented as a triplet that links two entities with a relation.

Generally, a triplet is expressed as (h, r, t) , where the source entity of relation (r) is called a head entity (h) and the destination entity of relation (r) is called a tail entity (t). For instance, knowledge about “Apple is a type of fruit” can be expressed as a triplet $(apple, is_a, fruit)$.

2.2 Knowledge Graph Embedding

A knowledge graph embedding model encodes knowledge relationships in a multidimensional vector space. TransE embeds entities and relations in a low-dimensional vector space, where a triplet (h, r, t) is embedded as entities h and t located at $\mathbf{h}$ and $\mathbf{t}$ to satisfy a vector operation of $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ ⁽¹⁾. In other words, TransE learns to construct a vector space, where $\mathbf{t}$ is located close to $\mathbf{h} + \mathbf{r}$, $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ is minimized if (h, r, t) is given in the knowledge graph, and $\|\mathbf{h}' + \mathbf{r} - \mathbf{t}'\|$ is maximized if (h', r, t') is not given in the knowledge graph. Here (h', r, t') is called a negative sampling, where either h or t is replaced with another entity from the original triplet. The loss function of TransE for learning is defined as

$$L_k = \sum_K \sum_{K'} [\gamma_k + d(\mathbf{h}, \mathbf{r}, \mathbf{t}) - d(\mathbf{h}', \mathbf{r}, \mathbf{t}')]_+ \quad (1)$$

$$d(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L1/L2}, \quad (2)$$

where γ_k denotes the margin for max-margin-based learning, $L1$ and $L2$ are the norms, $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ expresses the distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$ using $L1$ or $L2$ norm, K denotes the set of triplets in the knowledge graph, and K' denotes the set of triplets generated by negative sampling. $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ becomes smaller if the triplet is in the original knowledge graph, whereas $d(\mathbf{h}', \mathbf{r}, \mathbf{t}')$ becomes larger if the triplet is not in the knowledge graph as a result of minimizing Eq. (1). TransE locates $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$ close to each other for accomplishing $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

Several extensions of TransE have been proposed, including TransH and TransR. TransH embeds multiple meanings of the relations by projecting $\mathbf{h}$ and $\mathbf{t}$ into the hyperplanes making the relation accomplished⁽²⁾. On the other hand, TransR embeds the relation between entities by

Fig.1. WK-model and loss functions.

transforming a vector space into another vector space using a transformation matrix⁽³⁾.

2.3 Jointly Embedding Knowledge Graph and Other Knowledge Sources

There are methods that embed the knowledge graph and other knowledge sources into a vector space. For example, Wang *et al.* proposed a model that embeds Wikipedia anchors and the co-occurrence of texts into a space⁽⁴⁾. Fang *et al.* proposed another model that embeds descriptions of entities, Wikipedia URL anchors, and a knowledge graph into a space⁽⁵⁾.

3. WK-model

3.1 Overview of WK-model

WK-model extends the model that embeds a knowledge graph with other knowledge sources. In particular, the proposed model jointly embeds triplets of a knowledge graph and similarity grades of word vectors into a low-dimensional vector space. Word vectors are low-dimensional vectors whose distance expresses semantical similarity between words⁽⁶⁾. WK-model allocates entities that have a higher similarity degree in word vectors closer to each other. For the final configuration, WK-model configures clusters of semantically similar entities that also satisfy relations in the knowledge graph.

WK-model defines two types of loss functions, for knowledge constraints L_k and similarity constraints L_v , and learns a proper allocation of entities by minimizing the loss functions using a stochastic gradient descent mechanism (Fig. 1). The locations of entities in the vector space are normalized when every epoch for learning is completed. The learning in WK-model utilizes negative sampling as used in TransE, TransR, TransH, and other models.

3.2 Knowledge Loss Function

Similar to TransE, WK-model uses the knowledge loss function represented in Eqs. (1) and (2). $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ of TransE is calculated using L1 or L2 norm⁽¹⁾, whereas $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ of WK-model is calculated using only L1 norm. WK-model also minimizes the knowledge loss function for learning to accomplish a proper allocation of entities in a vector space.

3.3 Similarity Loss Function

WK-model defines the loss function L_v as a sum of L_{pv} and L_{nv} that calculate the differences between cosine similarities in both the knowledge graph embedding model and word vectors:

$$L_{pv} = \sum_K [-\gamma_{cs} + |cs(\mathbf{h}, \mathbf{t}) - cs(\mathbf{h}_v, \mathbf{t}_v)|]_+ \quad (3)$$

$$L_{nv} = \sum_{K'} [-\gamma_{cs} + |cs(\mathbf{h}', \mathbf{t}') - cs(\mathbf{h}'_v, \mathbf{t}'_v)|]_+ \quad (4)$$

$$L_v = L_{pv} + L_{nv}. \quad (5)$$

Cosine similarity between $\mathbf{v}_1$ and $\mathbf{v}_2$ is expressed as $cs(\mathbf{v}_1, \mathbf{v}_2)$ for the vectors in word vectors of $\mathbf{h}_v, \mathbf{t}_v, \mathbf{h}'_v, \mathbf{t}'_v$ and the corresponding vectors in WK-model of $\mathbf{h}, \mathbf{t}, \mathbf{h}', \mathbf{t}'$. γ_{cs} denotes the margin for max-margin learning according to Eq. (1). L_{pv} denotes the loss function for a triplet $(h, r, t) \in K$. This function calculates the difference of similarities in word vectors and WK-model. Thus, to minimize L_{pv} , WK-model allocates vectors $\mathbf{h}$ and $\mathbf{t}$ that have the same similarity grade as they have in word vectors. L_{nv} is a function similar to L_{pv} but for negative sampling.

3.4 Negative Sampling

WK-model uses negative sampling to improve the resolving capability as used in TransE, TransH, and TransR. Since the mechanism of replacing h or t was not described for TransE⁽¹⁾, we used two options for negative sampling proposed by Wang *et al.*⁽²⁾.

The first method for negative sampling chooses h or t for replacement with the same likelihood labeled as “unif.” Although some relations appear in triplets in a form of one-to-many correspondence and the others in a form of many-to-one, a negative sampling rate might not correspond to the actual occurrence if h and t are always chosen with the same probability. To solve it, the second method for negative sampling chooses h or t with the probability corresponding to the actual occurrence labeled as “bern.” Both options are tested in the experiments.

4. Experiments

4.1 Evaluation Criteria

We conducted two types of experiments to compare WK-model and TransE. First, we evaluated the capability of the generated knowledge graph and another is an implication of similarity between word vectors and the models.

4.1.1 Capabilities

Two types of capabilities are expected from a knowledge graph in generating a knowledge graph embedding model:

- (1) The ability of the knowledge embedding model to keep and regenerate a triplet of $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ even if the entity of the triplet is removed;
- (2) The ability of the knowledge embedding model to estimate and generate an unknown triplet of $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ by referring to the surrounding relations.

These two capabilities are almost equivalent to predicting a wildcard entity of $*$ in $(h, r, *)$ or $(*, r, t)$, which is called an entity prediction in TransE⁽¹⁾, TransH⁽²⁾, and TransR⁽³⁾.

First, we generated a expression having wildcard such as $(h, r, *)$ or $(*, r, t)$ from (h, r, t) in the test data for the entity prediction. Second, we generated a new set of triplets by respectively choosing an entity e from all entities in the knowledge graph and setting it to $*$. Third, we calculated $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ for all generated triplets and ranked them for all $(h, r, *)$ or $(*, r, t)$. Finally, we calculated the average ranks for all test data as MeanRank and probabilities involved in the top 10 as Hit@10.

For the entity prediction, Bordes *et al.*⁽¹⁾ pointed out that it must be counted as an error if there are generated triplets in the training, test, and validation data⁽¹⁾. Therefore, we tested two options: removing such triplets from ranking labeled as “Filter” and not removing them from ranking labeled as “Raw.”

4.1.2 Implication of Similarity

The implication of similarity can be measured as a difference between cosine similarities of two entities in the word vectors and generated knowledge graph space. We calculated the root mean square error (RMSE) and mean absolute error (MAE) to evaluate the implication level of similarity of word vectors with the following three classes:

- (a) Combinations of two entities that exist only in the test or validation data;

- (b) Combinations of two entities generated by negative sampling;
- (c) Combinations of two entities that exist in the training data.

There are no chances that two entities in class (a) simultaneously appear in learning, a few chances in class (b), and more chances in class (c).

4.2 Dataset and Preprocessing

We used the WN18 dataset generated by Bordes *et al.*⁽¹⁾, which is a part of WordNet, as the knowledge dataset. WordNet is a conceptual dictionary that classifies words into groups, “synsets”, of synonyms and describes definitions and relationship between “synsets”. WN18 has 151,442 triplets configured with 18 relations and 40,943 entities.

We used the Wikipedia model⁽⁸⁾ pre-trained using fastText⁽⁷⁾, an extension of Word2Vec⁽⁶⁾, as word vectors.

All entities of WN18 do not necessarily match the words in the pre-trained model. The words containing a delimiter such as “silicon_valley” can be divided into separate words such as “silicon” and “valley” to find a match. However, there are entities that cannot match any word even if this method is applied to the dataset. To maintain the integrity, we removed words that do not exist in the pre-trained model from the knowledge dataset. After removing, the size of the knowledge dataset became 131,990 triplets configured with 18 relations and 39,119 entities.

4.3 Parameters of the Experiments

For TransE, we selected the learning rate λ in $\{0.1, 0.01, 0.001\}$, the margin for learning knowledge γ_k in $\{1, 2, 10\}$, and the latent dimension k in $\{20, 50, 100\}$. The best models were selected by early stopping whose interval was 20 epochs using MeanRank. These parameters are equivalent to the parameters used by Bordes⁽¹⁾. Two options for negative sampling were selected, namely, “unif” and “bern,” and the options for norms were L1 and L2 norms. We replicated TransE using the code provided by Lin *et al.*⁽³⁾ (<https://github.com/thunlp/KB2E>) and added an early stopping function.

For WK-model, we selected the learning rate λ in $\{0.1, 0.01, 0.001\}$, the margin for learning knowledge γ_k in $\{1, 2\}$, and the latent dimension k in $\{50, 100\}$. The best models were selected by early stopping whose interval was 20 epochs using MeanRank. The margin for learning similarity γ_{cs} was 0 or selected from $\{0.05, 0.1, 0.5\}$, and options of negative sampling were “unif” and “bern”.

Table1. Entity prediction results

	Head				Tail			
	MeanRank		Hit@10		MeanRank		Hit@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TE1	439	426	79.2	91.7	414	401	78.2	91.4
TE2	265	253	49.1	53.8	273	261	48.9	53.9
WK1	344	332	74.8	86.9	373	362	74.6	87.3
WK2	275	264	76.4	88.3	262	250	75.7	88.5

Table2. The difference of similarity results

	RMSE			MAE		
	(a)	(b)	(c)	(a)	(b)	(c)
TE1	0.239	0.257	0.309	0.200	0.214	0.257
TE2	0.223	0.237	0.475	0.190	0.209	0.431
WK1	0.090	0.085	0.008	0.071	0.066	0.004
WK2	0.082	0.077	0.039	0.064	0.060	0.037

Fig. 2. Histogram of cosine similarity in Word Vectors

(a) Histogram of cosine similarity in TE1.

(b) Histogram of cosine similarity in TE2.

Fig. 3. Histogram of cosine similarity in TransE

Fig. 4. Histogram of cosine similarity in WK-model

4.4 Results of the Experiments

Tables 1 and 2 list the results of the experiments comparing WK-model and TransE. “TE1” in the tables stands for TransE with “unif” and L1-norm and “TE2” for TransE with “unif” and L2-norm. “WK1” in the tables stands for WK-model with $\gamma_{cs} = 0$ and “unif”, and “WK2” for WK-model with $\gamma_{cs} = 0.05$ and “bern”.

Table 1 demonstrates the capabilities of the knowledge graphs measured with the entity prediction. Smaller values of MeanRank and higher values of Hit@10 represent higher capacities. No notable differences between Head and Tail results can be observed, and all results in “Filter” columns are higher than those in “Raw” columns for the same model. TE1 shows the best performances in Hit@10, whereas TE2 shows the best performances in MeanRank. However, TransE has unbalanced performances because it indicates the worst performance in the columns for the parameters that achieve the best performance in the other columns. On the other hand, WK2 has higher performances close to the best in both MeanRank and Hit@10.

Table 2 demonstrates the implication degrees of similarity between word vectors and the knowledge graph embedding models. Smaller RMSE and MAE represent smaller differences of similarity between word vectors and the models. Columns (a), (b), and (c) in Table 2 indicate classes described in Section 4.1.2. WK-model has the best performance in terms of both the RMSE and MAE. WK1 has

the best performance for class (c), whereas WK1 has worse performance in term for classes (a) and (b). Because of this, the margin γ_{cs} might have contributed to the performance of WK2, which have balanced performance.

Figures 2 demonstrates the relative histogram of cosine similarities between two words picked up from the word vectors. Although classes (a) and (b) almost randomly select two words from the source of word vectors, their similarities are distributed in a positive-value region. This might be an encoding feature of the word vectors. Figures 3 demonstrates the relative histogram of cosine similarities between entities in TransE. Similarities in classes (a) and (b) are distributed almost in normal distribution around zero. On the other hand, similarities in class (c) is biased to 1.0. This result indicates that head and tail entities in a relation are allocated very close to each other in the vector space. Figures 4 demonstrates the relative histogram of cosine similarities between entities in WK model and words in word vectors. Shapes of graphs are very similar to graphs in Fig. 2. Strong embedding tendency for head and tail entities in a relation is relieved, and characteristics of entity allocations in word vectors are well embedded.

5. Discussion

MeanRank of TransE with L1 norm is 439, although 79% triplets of the test data are at the top 10 in the case of Head and “Raw.” If 79% of triplets were at the rank 10, the remaining 21% triplets would be around the rank 2000. Therefore, MeanRank becomes higher when removing the remaining triplets out of the top 10 in TransE with L1 norm. Similar results are obtained for Tail and “Filter.” MeanRank of TransE with L2 norm is 265, although only 49% triplets of the test data are at the top 10 in the case of Head and “Raw.” If 49% of triplets were at the rank 10, the remaining 51% triplets would be around the rank 510. Therefore, TransE with L2 norm cannot evenly optimize the performance for all possible triplets in the test data. On the basis of these two facts, it can be concluded that the distance measure $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ has a great impact on the performance of TransE.

MeanRank of WK-model with $\gamma_{cs} = 0.05$ is 275, although 76% triplets of the test data are at the top 10 in the case of Head and “Raw.” If 76% of triplets were at the rank 10, the remaining 24% triplets would be at the rank 1100. Therefore, the similarity of word vectors can reduce the impact of the distance measure $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$ and improve the unbalanced performance of TransE.

The differences of similarities between entities in class

(c) are smaller than those in classes (a) and (b). This indicates that the opportunity for learning is related to the implication of similarity.

The differences of similarities between entities in classes (a) and (b) are modified from TransE. These indicate that the similarity of the entities in the model can be close to the word vectors even if the opportunity for learning the words is 0. There are entities in class (a) similar to those in classes (b) and (c); thus, the similarity between entities in class (a) might be formulated during learning the similarity between entities in classes (b) and (c). From these viewpoints, it can be concluded that WK-model can embed the similarities of words without direct relationships by combining the knowledge graph and word vectors.

6. Conclusions

In this paper, we proposed WK-model, a knowledge embedding model that includes similarities of word vectors. According to the experimental results, the similarities in WK-model can be close to the similarities in word vectors, and the performance can be reasonable for any distance function $d(\mathbf{h}, \mathbf{r}, \mathbf{t})$, whereas TransE demonstrated an unbalanced performance against the condition of distance function.

The extensions of TransE such as TransH and TransR reported a way to improve the performance of vector operations by projecting the relations to hyperplanes or mapping them to another vector space. Therefore, we intend to apply the proposed method to the models of TransH and TransR.

References

- (1) A.Bordes, N.Usunier, A.G.Duran: "Translating embeddings for modeling multi-relational data," Advances in neural information processing systems, pp.2787-2795, 2013.
- (2) Z.Wang, J.Zhang, J.Feng, Z.Chen: "Knowledge Graph Embedding by Translation on Hyperplanes," AAAI, Vol. 14, 2014.
- (3) Y.Lin, Z.Liu, M.Sun, Y.Liu, X.Zhu: "Learning entity and relation embeddings for knowledge graph completion," The 29th AAI Conference on Artificial Intelligence (AAAI'15), Vol. 15, 2015.
- (4) Z.Wang, J.Zhang, J.Feng, Z.Chen: "Knowledge graph and text jointly embedding," EMNLP, pp.1591-1601, 2014.
- (5) W.Fang, J.Zhang, D.Wang, Z.Chen, M.Li: "Entity disambiguation by knowledge and text jointly embedding," Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp.260-269, 2016.
- (6) T.Mikolov, K.Chen, G.Corrado, J.Dean: "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
- (7) P.Bojanowski, E.Grave, A.Joulin, T.Mikolov: "Enriching Word Vectors with Subword Information," TACL5, pp.135-146, 2017.
- (8) FacebookResearch, <https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md>, reference date : 28, January, 2019.