Cognitive Computation
https://doi.org/10.1007/s12559-019-09686-4

Knowledge Fusion via Joint Tensor and Matrix Factorization

Zengguang Hao[1] · Yafang Wang[1] · Zining Liu[1] · Gerard de Melo[2] · Zenglin Xu[3]

Received: 13 October 2018 / Accepted: 23 September 2019

Abstract
We consider the task of knowledge fusion, an important aspect of cognitive intelligence, with the goal of combining *part-of* knowledge drawn from different sources. For this, entities and relations are cast into matrix-based representations. Unlike previous work on relation prediction, we consider the challenging setting of graphs with large amounts of completely separate connected components and no overlap between the training and test set entities. In order to address these challenges, we propose a novel cognitively inspired factorization method that jointly factorizes a *subject–predicate–object* tensor via RESCAL and a similarity matrix via matrix factorization. Our experimental results show that our method significantly outperforms several strong baseline models, including RESCAL and several TransE-style models. The proposed joint factorization of a *subject–predicate–object* tensor while applying matrix factorization to a similarity matrix obtains substantially higher average accuracy rates than previous approaches. This shows that it can successfully address the challenge of knowledge fusion of disconnected data.

Keywords Knowledge fusion · Connected components · Entity overlap · Tensor factorization · Word similarities

Introduction

In the last decade, substantial progress has been made in cognitive computing [38], and much of this has rested on the ability to draw on large amounts of data to overcome the traditional knowledge acquisition bottleneck. Such knowledge can be provided in more explicit forms, drawing for instance on the emergence of large knowledge graphs, or can be mined from unstructured sources using machine learning and information extraction. Despite the substantial growth of available knowledge, downstream applications often need to cull pertinent information from multiple sources, because no single source is sufficiently comprehensive. This is not only true for information mined from text, but also often applies to crowdsourced and other kinds of data. In practice, disparate sources tend to have heterogeneous distributions and properties, which means that it is non-trivial to suitably combine information across sources, a problem that is often referred to as knowledge fusion [31]. This is akin to how humans acquire knowledge from multiple sources and may need to figure out how they fit together, and at times also determine which source is more likely to be correct. Knowledge fusion is important whenever one wishes to draw on multiple sources, and has found application in massive information retrieval, knowledge management, e-learning, and knowledge acquisition, among others [41]. It can also serve as a means of discovering and cleaning errors present in individual knowledge sources [9].

In this paper, we focus on the fusion of knowledge pertaining to the *part-of* relation, which reflects prototypical mereological relationships between objects, such as between *wheel* and *bicycle*. Such knowledge is an essential form of commonsense knowledge, given its prominent

✉ Yafang Wang
yafang.wang@sdu.edu.cn

Zengguang Hao
hzg@mail.sdu.edu.cn

Zining Liu
liuzining@mail.sdu.edu.cn

Gerard de Melo
gdm@demelo.org

Zenglin Xu
zlxu@uestc.edu.cn

[1] Shandong University, Jinan, 250101, China

[2] Rutgers University, New York, NY 10002, USA

[3] University of Electronic Science and Technology of China, Chengdu, China

Published online: 18 November 2019

role in enabling us to understand the composition of the world. These sorts of relationships are also of prime importance in lexical semantics, and, thus, can be sourced from lexical resources providing meronymic information, such as WordNet [11], from resources mined from text and images, such as PWKB [36], from crowdsourced data, such as VisualGenome [18], or from large knowledge graphs, such as GeoNames.[1] Although *part-of* relationships are abundant in such sources, the underlying definitions are inconsistent. Given a generic *part-of* relationship from one data source, we may not know specifically which kind of *part-of* relationship holds. Thus, knowledge fusion in this case entails discerning between different sorts of relationships that are easily confused.

This form of knowledge fusion bears a relationship to the task of knowledge graph completion, which focuses on the prediction of missing knowledge. In recent years, embedding-based approaches for knowledge graph completion have excelled at the task, exhibiting both strong generalization capabilities and robustness. Two main groups can be distinguished: tensor factorization approaches such as RESCAL [27] and translation-based models such as TransE [3] and its numerous variants (as discussed in Section "Related Work"). We show that neither framework can effectively cope with our knowledge fusion task, which involves graphs that harbor large numbers of completely separate connected components, lacking any overlap between the entities from training and test sets. We will expound on this in further detail in Section "Problem Definition".

To overcome the aforementioned challenges, we propose a cognitive-inspired joint decomposition of a *subject–predicate–object* tensor and a similarity matrix. The main contributions of this paper are as follows:

- We put forward a new task and corresponding datasets in the area of knowledge fusion, aiming at the important challenge of integrating knowledge graphs with large amounts of isolated connected components, with non-overlapping training set and test set entities.
- We propose a cognitively inspired method that exploits a similarity matrix as side information, providing a link between entities in the graph. The proposed method fuses the graphs via joint factorization of the tensor and the similarity matrix, optimized via an Alternating Direction Method of Multipliers strategy.
- In our experiments, we evaluate our method against state-of-the-art baselines. The results show that our proposed method substantially outperforms all baselines in terms of the achieved accuracy levels.

Related Work

Knowledge Fusion Knowledge fusion is a long-standing but increasingly important problem, which aims at identifying a true and coherent set of knowledge in the form of *subject–predicate–object* triples given input triples extracted from heterogeneous knowledge bases [8]. There has been substantial research in this broad area of inquiry. For example, Nengfu et al. [26] proposed a rule-based knowledge fusion method, which provides its answers by combining different answers from different sources, and developed fusion methods to select rules that meet specific user preferences. Dong et al. [8] proposed adapting traditional data fusion techniques, including voting and Bayesian analysis methods, to the task of knowledge fusion. Their work at Google aimed at building a large-scale Knowledge Vault using a multi-source knowledge fusion method based on supervised learning [7]. Thoma et al. [37] introduced an approach for cross-modal knowledge fusion that integrates visual and textual latent representations with embeddings of knowledge graph concepts. Our paper studies a distinct setting, where the goal is to fuse knowledge while distinguishing different kinds of relations.

Knowledge Graph Completion In recent years, there has been substantial research on knowledge graph completion. The well-known TransE method [3] maps entities to vectors and regards relations r as translations from a head entity h to a tail entity l. Based on TransE, a number of improved models have been proposed, such as TransH [39], TransR [22], CTransR [23], PTransE [21], and TranSparse [14]. Specifically, TransE attempts to make $h + r$ and l be as close as possible by adjusting the vectors for the head h, relation r, and tail l. In order to better model N–1, 1–N, and N–N relations, the TransH method [39] instead models relations as hyperplanes with an associated translation operation. TransE and TranH both embed the entities and relations into the same space. The TransR [22] method instead considers separate entity and relation spaces to better capture the differences between entities and relations. Based on TransR, Lin et al. proposed the CTransR [23] model, which clusters and groups the head–tail entities. PTransE [21] is based on relation paths, exploiting possible paths of linking the two entities as features to predict the relation. Ji et al. [14] proposed TranSparse to solve the challenge of heterogeneous and unbalanced objects (entities and relations) in a knowledge graph. The KG2E method [13] uses Gaussian distributions to reflect the data uncertainty and improve the link prediction accuracy. An alternative direction is to focus on tensor or matrix decomposition. In particular, [28] factorized the large YAGO 2 core ontology using a technique called RESCAL [27], a restricted Tucker

[1] http://www.geonames.org/

decomposition for link prediction. He et al. [12] proposed a Bayesian neural tensor decomposition approach to model the deep correlations or dependency between the latent factors in knowledge base completion. As we show, most such methods struggle when faced with graphs that consist of numerous separate connected components.

Tensor Decompositions Standard matrix factorization approaches have been generalized to tensor factorization for 3-dimensional data. There are two main algorithms for tensor decomposition [17]: CP (CANDECOMP/PARAFAC) [16] and Tucker decomposition [6, 42]. Besides these, INDSCAL [35], PARAFAC2 [5], and PARATUCK2 [10] are further classic algorithms for tensor decomposition. In terms of applications, Dong et al.'s work on the Google Knowledge Vault [7] relied on tensor low-rank decompositions for link prediction. Liu et al. [24] propose a neural model connecting neural networks with a Bayesian tensor decomposition to effectively model complex nonlinear relationships. Solé-Casals et al. [34] use tensor completion applied to electroencephalography data to improve the classification performance in a motor imagery brain–computer interface system with corrupted measurements.

However, simple tensor decompositions are unable to cope particularly well with graphs that contain numerous separate connected components. Hence, we rely on a joint decomposition method that exploits additional side information. An approach related to this is that of Acar et al. [1], who formulate a data fusion model via the Coupled Matrix and Tensor Factorization (CMTF) framework, in which the model uses CP decomposition. However, according to our experimental results, CP decomposition is less appropriate for our model.

Problem Definition

Setting

As an important component in cognitive intelligence, knowledge fusion aims to solve the task of integrating a large amount of knowledge from heterogeneous sources. In this paper, we focus on the integration of mereological knowledge, i.e., of *part-of* relationships. The *part-of* relation is one of the most fundamental ones both in formal ontology and in cognitive science, while meronymy is of prime importance in lexical semantics. Unfortunately, the *part-of* relationship is also particularly challenging, because one can in fact distinguish several different kinds of *part-of* relationships.

In particular, we consider the *physicalPartOf* relationship data from PWKB [36], as well as pertinent *part-of* relationships from VisualGenome [18], WordNet [11], and GeoNames as knowledge sources. VisualGenome has many noisy triples, and several different sorts of relationships, including *has-a*, *is-a*, *located-in*, and orientation-based labels (e.g., *in*, *on*, *behind*). Some relations in VisualGenome are overlapping, such as the two relations *on face of* and *part of* in triples such as *(nose, on face of, dog)* and *(nose, part of, dog)*. GeoNames provides mereological data pertaining only to geopolitical entities. Finally, WordNet covers entities from diverse domains, in categories that include *physical entities* (e.g., *animal*, *plant*, *substance*, *physical process*) and *abstract entities* (e.g., *quantity*, *attribute*, *psychological feature*). The WordNet Tensor Data [2] uses a single label *part of* across these different sorts of entities.

The relation definition for the *part-of* relationships across PWKB, VisualGenome, and GeoNames is inconsistent. To demonstrate the merits of our method, we sampled objects to cover a wide spectrum of domains, including animals, plants, artifacts, and locations from PWKB, VisualGenome, and GeoNames. For convenience, in the following, the four domains will be referred to as Animal *part-of* (e.g., nose and dog), Plant *part-of* (e.g., leaf and tree), Artifact *part-of* (e.g., leg and chair), and Located-in *part-of* (e.g., Beijing and China), respectively.

Goal

We consider these as separate knowledge fusion targets with separate relation labels. The goal will be to take WordNet data, which is not classified, and fuse it into four domain-specific knowledge sources by predicting specific kinds of *part-of* relationships, with respect to these four targets.

In our experiments, these four domains each will have a training and test set. Data sampled from PWKB, VisualGenome and GeoNames will be divided into two parts: 75% as training set, and the remainder as the validation set. Data sampled from WordNet will serve as the respective test sets. These datasets have two important characteristics: (1) Because the WordNet dataset is a fairly large one in terms of the number of words (entities), entities in the test set will be mostly non-overlapping with those in the training set. (2) The amount of connected components is very large in the graphs for the respective two datasets, meaning that there are many isolated parts. Throughout this paper, connected components refer to weakly connected components in a directed graph.

Comparison with Knowledge Graph Completion

The task we consider in this paper bears a relationship to the task of knowledge graph completion, which focuses on predicting missing knowledge given input knowledge as training data. Although there are numerous strong

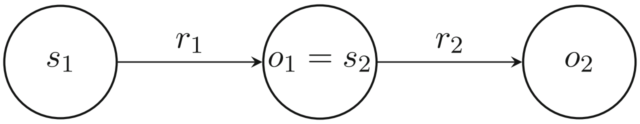

Fig. 1 Example of connectivity between triples

knowledge graph completion and link prediction methods, including RESCAL- and TransE-style approaches, such methods typically can only predict an unseen relation if substantial correlations or links are present. When such information is missing, the methods often fare very poorly.

We consider the widely usd FB15k and WN18 datasets from the TransE paper [3] to shed further light on this. Table 1 provides statistics of these datasets. First of all, we find that both FB15k and WN18 have high proportions of overlapping entities between the training and test sets, as given in the *Entity Overlap* columns. In Table 1, the *Entity Overlap* column for the training set provides the number of entities in the training set that can also be found in the test set. The *Entity Overlap* column for the test set provides the number of entities in the test set that can also be found in the training set.

A second observation is that both FB15k and WN18 have very few connected components. Since there must exist a link from subject to object in each triple, the number of connected components depends directly on the number of entity links across different triples. Hence, a simple means of assessing a dataset is to determine the number of entities that exist in both subject and object positions in that dataset. Figure 1 shows two triples (s_1, r_1, o_1) and (s_2, r_2, o_2), where $o_1 = s_2$. Graph-theoretically, this scenario corresponds to having additional links between triples, which entails a reduction in the number of connected components. Thus, the more we encounter this sort of triple pattern, the lower the amount of connected components. According to the two respective *Fig. 1 Pattern* columns in Table 1, both FB15K and WN18 have high proportions of links occurring in this sort of configuration. This means that there are few connected components and the overall graph is fairly well-connected.

While knowledge graph completion methods work well on FB15k and WN18, they face significant difficulties when applied to datasets that have non-overlapping entity sets and large numbers of separate connected components. We study this in Section "Experiments".

To solve the above two issues, we propose a joint decomposition technique based on a *subject–predicate–object* tensor and a separate similarity matrix. Tensors are selected as a structure of choice to capture the knowledge graph, because they provide a more convenient and more cognitively inspired way to describe multi-source data and to suitably capture their multi-linear structure [32]. In the proposed method, similarities may serve as bridges between triples. In cognitive science, similarity-based associations between concepts are widely viewed as essential ingredients of human cognition. In particular, humans often rely on notions of similarity to infer information when limited prior knowledge is available.

Proposed Method

In this paper, we propose to rely on a 3rd-order tensor model to predict the relation and thereby select appropriate relationships for data fusion, with said tensor denoted as $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. As discussed, we additionally rely on entity similarities as side information to assist in the process of relation prediction. These similarities are assumed to be given in the form of a similarity matrix $\mathbf{P} \in \mathbb{R}^{I \times J}$.

Model

Figure 2 provides an illustration of our model, including the Subject × Object × Relation tensor and the Subject × Object similarity matrix.

Subject x Object x Relation Tensor We establish a Subject × Object × Relation tensor, denoted as $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, where I, J, and K represent the number of subjects, objects, and relations, respectively. Any entry $\mathcal{X}_{ijk}$ in $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ captures a score that characterizes the probability of a relation k between subject i and object j. The higher this score is, the more likely the relationship is taken to hold by the model. In practice, the scores of training triples with known true relationships are initialized to a common larger constant such as 1, while other entries are initialized to

Table 1 Statistics for standard knowledge graph completion datasets

Datasets	No. of entities in training set			No. of entities in test set		
	Entity overlap	Fig. 1 Pattern	Total	Entity overlap	Fig. 1 Pattern	Total
FB15k	481,316	483,129	483,142	59,071	58,589	59,071
WN18	58,302	141,440	141,442	4998	2128	5000

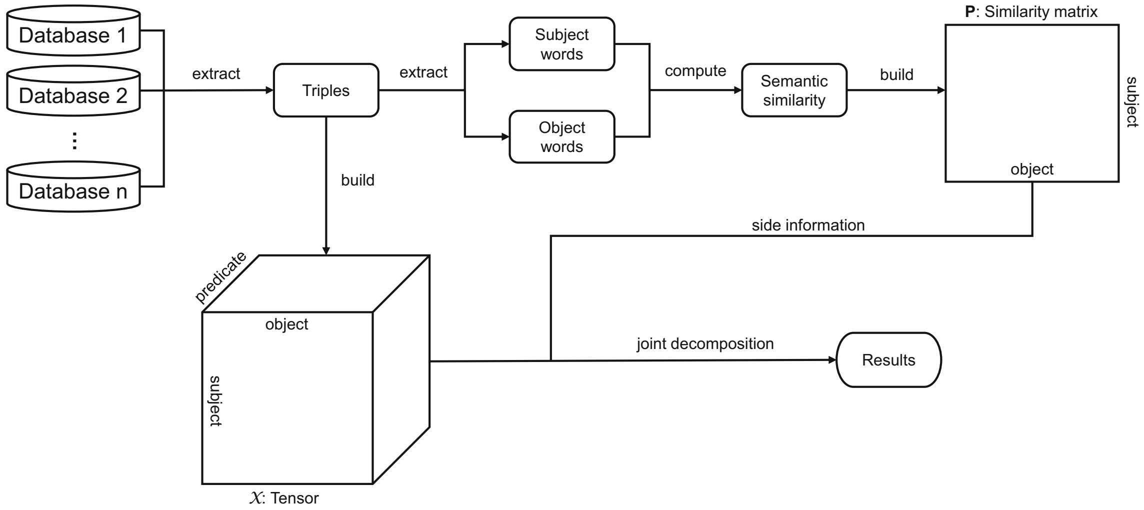

Fig. 2 Tensor and similarity model. The triples are taken from multiple databases, and serve as a source of subject words as well as object words. We compute the semantic similarity between any two words as side information. Finally, the joint decomposition model predicts the scores of all relations for every pair of subject word and object word

global smaller constant such as 0. Due to the small size of the training set, $\mathcal{X}$ is a sparse tensor.

Subject × Object Similarity Matrix As the graph consists of a large amount of isolated connected components, it is not sufficient to directly predict relations based on a very sparse tensor of this sort. Thus, a Subject × Object similarity matrix is constructed to serve as side information to the model, in order to assist in better predicting the relations. The similarity matrix is denoted as $\mathbf{P} \in \mathbb{R}^{I \times J}$. Due to the nature of similarity scores, this matrix is dense. A careful analysis of the model shows that within the side information, only similarities between any two subject words or any two object words are effectively used to determine the relationships. Similarities between subject and object are not required. Hence, the similarities between any two subjects and any two objects are computed by an appropriate similarity metric. We will analyze similarity computation methods in Section “Similarity”.

Given these two inputs, we rely on tensor decomposition with side information to optimize for the objective function in Section “Objective Function” and complete $\mathcal{X}$ via an ADMM approach [4] in Section “Model Solution Based on ADMM”. Finally, we select the relation with the highest score as the relation of the corresponding pair of subject and object.

Objective Function

The goal of the model is a joint analysis of the information captured by the sparse tensor $\mathcal{X}$ and the side information matrix $\mathbf{P}$ to predict the score of every candidate relationship. Hence, we define an objective function in which the main tensor and side information matrix are decomposed simultaneously, and then seek to minimize this objective. The joint objective simultaneously decomposes the tensor $\mathcal{X}$ via the RESCAL [27] method, while for the side information matrix, it relies on standard matrix factorization. To achieve this, the objective function Z is composed of a tensor function denoted by Z_1 as well as a side information matrix function Z_2:

$$Z = Z_1 + Z_2, \tag{1}$$

where Z_1 denotes the least squares error with respect to the RESCAL objective

$$\begin{aligned} Z_1 = & \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{X_k} - \mathbf{A_1}\mathbf{R_k}\mathbf{B_1}^{\mathsf{T}}\|_{\mathsf{F}}^2 \\ & + \frac{\lambda_3}{2}\sum_{k=1}^{K}\|\mathbf{R_k}\|_{\mathsf{F}}^2 + \frac{\lambda_1}{2}(\|\mathbf{A_1}\|_{\mathsf{F}}^2 + \|\mathbf{B_1}\|_{\mathsf{F}}^2) \end{aligned} \tag{2}$$

In the above equation, $\mathbf{X_k}$ denotes the kth slice of the $\mathcal{X}$ tensor. It is decomposed as a matrix multiplication of three matrices. $\mathbf{A_1} \in \mathbb{R}^{I \times R}$ captures the latent component representation of subjects, while R is the rank of the tensor. $\mathbf{B_1} \in \mathbb{R}^{J \times R}$ will contain the latent component representation of objects. $\mathbf{R_k}$ is an asymmetric $R \times R$ matrix that models the interactions of $\mathbf{A_1}$ and $\mathbf{B_1}$ in the kth predicate. The final two items in Eq. 2 are the regularization terms, in which λ_1 and λ_3 serve as regularization parameters.

Similarly, Z_2 represents the least squares error with regularization term for decomposing the similarity matrix

$\mathbf{P}$ into $\mathbf{A_2}$ and $\mathbf{B_2}$ using standard matrix factorization. It is defined as

$$Z_2 = \frac{1}{2}\|\mathbf{P} - \mathbf{A_2}\mathbf{B_2}^{\mathsf{T}}\|_{\mathsf{F}}^2 + \frac{\lambda_2}{2}(\|\mathbf{A_2}\|_{\mathsf{F}}^2 + \|\mathbf{B_2}\|_{\mathsf{F}}^2), \tag{3}$$

where the definitions of $\mathbf{A_2}$, $\mathbf{B_2}$ are the same as for $\mathbf{A_1}$ and $\mathbf{B_1}$. λ_2 is a regularization parameter.

Considering that the side information matrix $\mathbf{P}$ is coupled with the main tensor $\mathcal{X}$, they share some common factors. As a result, $\mathbf{A_1}$ and $\mathbf{A_2}$ should be equal to the extent possible, and the same applies accordingly to $\mathbf{B_1}$ and $\mathbf{B_2}$. To encourage this, we introduce global factor variables denoted by $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ and corresponding constraints defined as Eq. 4. Furthermore, to some extent, global factors help to accelerate the convergence of the model, because $\mathbf{A_1}$ and $\mathbf{A_2}$ possibly have a substantial difference given that they represent different information.

$$\begin{aligned} \mathbf{A_i} - \overline{\mathbf{A}} &= 0 \qquad \forall i \in \{1, 2\} \\ \mathbf{B_i} - \overline{\mathbf{B}} &= 0 \qquad \forall i \in \{1, 2\} \end{aligned} \tag{4}$$

Model Solution Based on ADMM

As mentioned above, our overall optimization problem is modified to minimize the objective function Z in Eq. 1 with the constraints in Eq. 4. Hence, the resulting constrained optimization problem is

$$\begin{aligned} &\min_{\mathcal{F}} \quad Z \\ s.t.\ &\mathbf{A_i} - \overline{\mathbf{A}} = 0 \qquad \forall i \in \{1, 2\} \\ &\mathbf{B_i} - \overline{\mathbf{B}} = 0 \qquad \forall i \in \{1, 2\} \end{aligned} \tag{5}$$

where $\mathcal{F} = \{\mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{R}_k\}$, and $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$ denotes the global factor matrices. To simplify the objective function Z with the constraints, Eq. 5 can be transformed into an unconstrained optimization objective via Lagrangian augmentation. The transformed objective function $L_\rho(\cdot)$ is

$$\begin{aligned} L_\rho(\cdot) = Z &+ \sum_{i=1}^{2} \mathrm{tr}([\Theta_A^i]^{\mathsf{T}}(\mathbf{A_i} - \overline{\mathbf{A}})) + \frac{\rho}{2}\sum_{i=1}^{2}\|\mathbf{A_i} - \overline{\mathbf{A}}\|_{\mathsf{F}}^2 \\ &+ \sum_{i=1}^{2} \mathrm{tr}([\Theta_B^i]^{\mathsf{T}}(\mathbf{B_i} - \overline{\mathbf{B}})) + \frac{\rho}{2}\sum_{i=1}^{2}\|\mathbf{B_i} - \overline{\mathbf{B}}\|_{\mathsf{F}}^2 \end{aligned} \tag{6}$$

where ρ denotes the penalty parameter. $L_\rho(\cdot)$ represents $L_\rho(\mathcal{F}, \mathbf{\Theta_A^1}, \mathbf{\Theta_A^2}, \mathbf{\Theta_B^1}, \mathbf{\Theta_B^2}, \overline{\mathbf{A}}, \overline{\mathbf{B}})$. Finally, $\mathbf{\Theta_A^1}$, $\mathbf{\Theta_A^2}$, $\mathbf{\Theta_B^1}$, $\mathbf{\Theta_B^2}$ are Lagrange multiplier parameters.

We adopt the ADMM technique [4] to optimize for this objective with Lagrangian augmentation. In particular, the iterative solution of Eq. 6 is as follows.

$$\begin{aligned} \mathcal{F}^{k+1} &\leftarrow \arg\min_{\mathcal{F}} L_\rho(\mathcal{F}, \Theta_A^1, \Theta_A^2, \Theta_B^1, \Theta_B^2, \overline{\mathbf{A}}, \overline{\mathbf{B}}) \\ \overline{\mathbf{A}}^{k+1}, \overline{\mathbf{B}}^{k+1} &\leftarrow \arg\min_{\overline{\mathbf{A}}, \overline{\mathbf{B}}} L_\rho(\mathcal{F}, \Theta_A^1, \Theta_A^2, \Theta_B^1, \Theta_B^2, \overline{\mathbf{A}}, \overline{\mathbf{B}}) \\ (\Theta_A^i)^{k+1} &= (\Theta_A^i)^k + \rho\left(\mathbf{A}_i^{k+1} - \overline{\mathbf{A}}^{k+1}\right) \quad \forall i \in \{1, 2\} \\ (\Theta_B^i)^{k+1} &= (\Theta_B^i)^k + \rho\left(\mathbf{B}_i^{k+1} - \overline{\mathbf{B}}^{k+1}\right) \quad \forall i \in \{1, 2\} \end{aligned} \tag{7}$$

Updating $\mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}$ We obtain the update formulae by taking the partial derivatives of L_ρ in Eq. 7 with regard to $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{B_1}$, $\mathbf{B_2}$, respectively, and set these derivatives to zero. The results are as follows, where $\boldsymbol{I}$ denotes the identity matrix.

$$\mathbf{A_1} = \left(\sum_{k=1}^{K} \mathbf{X}_k\mathbf{B_1}\mathbf{R}_k^{\mathsf{T}} - \Theta_A^1 + \rho\overline{\mathbf{A}}\right)\left(\sum_{k=1}^{K} \mathbf{R}_k\mathbf{B}_1^{\mathsf{T}}\mathbf{B_1}\mathbf{R}_k^{\mathsf{T}} + (\rho + \lambda_1)\boldsymbol{I}\right)^{-1} \tag{8}$$

$$\mathbf{B_1} = \left(\sum_{k=1}^{K} \mathbf{X}_k^{\mathsf{T}}\mathbf{A_1}\mathbf{R}_k - \Theta_B^1 + \rho\overline{\mathbf{B}}\right)\left(\sum_{k=1}^{K} \mathbf{R}_k^{\mathsf{T}}\mathbf{A}_1^{\mathsf{T}}\mathbf{A_1}\mathbf{R}_k + (\rho + \lambda_1)\boldsymbol{I}\right)^{-1} \tag{9}$$

$$\mathbf{A_2} = \left(\mathbf{P}\mathbf{B_2} - \Theta_A^2 + \rho\overline{\mathbf{A}}\right)\left(\mathbf{B}_2^{\mathsf{T}}\mathbf{B_2} + (\lambda_2 + \rho)\boldsymbol{I}\right)^{-1} \tag{10}$$

$$\mathbf{B_2} = \left(\mathbf{P}^{\mathsf{T}}\mathbf{A_2} - \Theta_B^2 + \rho\overline{\mathbf{B}}\right)\left(\mathbf{A}_2^{\mathsf{T}}\mathbf{A_2} + (\lambda_2 + \rho)\boldsymbol{I}\right)^{-1} \tag{11}$$

Updating $\mathbf{R}_k$ Following previous work [27], by holding $\mathbf{A_1}$ as well as $\mathbf{B_1}$ constant, and vectorizing $\mathbf{X}_k$ together with $\mathbf{R}_k$, the function for minimizing $\mathbf{R}_k$ can be transformed as

$$f(\mathbf{R}_k) = \frac{1}{2}\|\mathrm{vec}(\mathbf{X}_k) - (\mathbf{B_1} \otimes \mathbf{A_1})\,\mathrm{vec}(\mathbf{R}_k)\|_2^2 + \frac{\lambda_3}{2}\|\mathrm{vec}(\mathbf{R}_k)\|_2^2 \tag{12}$$

where $\otimes$ denotes the Kronecker product. Minimizing Eq. 12 can be viewed as regularized linear regression. As a result, the solution of Eq. 12 is

$$\mathrm{vec}(\mathbf{R}_k) = \left(\mathbf{M}^{\mathsf{T}}\mathbf{M} + \lambda_3\boldsymbol{I}\right)^{-1}\mathbf{M}^{\mathsf{T}}\mathrm{vec}(\mathbf{X}_k) \tag{13}$$

where $\mathbf{M}$ denotes $\mathbf{B_1} \otimes \mathbf{A_1}$. However, computing $\mathbf{M}^{\mathsf{T}}\mathbf{M}$ is very inefficient. This can be solved by singular value decomposition (SVD) of $\mathbf{A_1}$ and $\mathbf{B_1}$, defined as $\mathbf{A_1} = \mathbf{U}_A\mathbf{S}_A\mathbf{V}_A^{\mathsf{T}}$ and $\mathbf{B_1} = \mathbf{U}_B\mathbf{S}_B\mathbf{V}_B^{\mathsf{T}}$ [27]. Then Eq. 12 can be cast as

$$f(\mathbf{R}_k) = \frac{1}{2}\|\mathbf{U}_A^{\mathsf{T}}\mathbf{X}_k\mathbf{U}_B - \mathbf{S}_A(\mathbf{V}_A^{\mathsf{T}}\mathbf{R}_k\mathbf{V}_B)\mathbf{S}_B^{\mathsf{T}}\|_{\mathsf{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{V}_A^{\mathsf{T}}\mathbf{R}_k\mathbf{V}_B\|_{\mathsf{F}}^2 \tag{14}$$

According to Eq. 13, the solution of Eq. 14 is

$$\begin{aligned} \mathrm{vec}(\hat{\mathbf{R}}_k) &= (\mathbf{N}^{\mathsf{T}}\mathbf{N} + \lambda_3\boldsymbol{I})^{-1}\mathbf{N}^{\mathsf{T}}\mathrm{vec}(\hat{\mathbf{X}}_k) \\ \mathbf{R}_k &= \mathbf{V}_A\hat{\mathbf{R}}_k\mathbf{V}_B^{\mathsf{T}} \end{aligned} \tag{15}$$

where $\hat{\mathbf{X}}_k = \mathbf{U}_A{}^{\mathsf{T}}\mathbf{X}_k\mathbf{U}_B$, $\hat{\mathbf{R}}_k = \mathbf{V}_A^{\mathsf{T}}\mathbf{R}_k\mathbf{V}_B$, and $\mathbf{N} = \mathbf{S}_B \otimes \mathbf{S}_A$.

Updating $\Theta_A^1, \Theta_A^2, \Theta_B^1, \Theta_B^2$ According to Eq. 7, the formulae for updating $\mathbf{\Theta_A^1}$, $\mathbf{\Theta_A^2}$, $\mathbf{\Theta_B^1}$, $\mathbf{\Theta_B^2}$ are easily computed.

Updating $\overline{A}$, $\overline{B}$ As for $\overline{\mathbf{A}}$, according to Eq. 7, its partial derivative is computed and set to zero. Then we obtain the update formula for $\overline{\mathbf{A}}$.

$$\overline{\mathbf{A}} = \frac{1}{2\rho}\left(\Theta_A^1 + \Theta_A^2 + \rho\mathbf{A_1} + \rho\mathbf{A_2}\right) \tag{16}$$

For computational convenience, the initial iterative value of Θ_A^i, denoted as $(\Theta_A^i)_0$, is set to zero. We can prove $\sum_{i=1}^{2}(\Theta_A^i)_k = 0$ by mathematical induction. Updating $\overline{\mathbf{B}}$ is similar to updating $\overline{\mathbf{A}}$. Therefore, the final update formulae are

$$\begin{aligned} \overline{\mathbf{A}} &= \frac{\mathbf{A_1} + \mathbf{A_2}}{2} \\ \overline{\mathbf{B}} &= \frac{\mathbf{B_1} + \mathbf{B_2}}{2} \end{aligned} \tag{17}$$

Algorithm Algorithm 1 shows the complete algorithmic procedure for our model. We take $\mathcal{X}$, $\mathbf{P}$, $\boldsymbol{I}$, $I_{\max}$, R, λ_1, λ_2, λ_3, and ρ as input. Before invoking the main algorithmic loop, we initialize relevant matrix variables and set the convergence or stop criterion for the loop. Then the algorithm starts updating these matrix variables recurrently until the convergence or stop criterion is met. Finally, the completed tensor $\mathcal{X}$ is generated by a multiplication of $\mathbf{A_1}$, $\mathbf{R}_k$, and $\mathbf{B_1}$.

Algorithm 1 Tensor decomposition via ADMM.

```
Require: Sparse tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, similarity matrix $\mathbf{P} \in \mathbb{R}^{I \times J}$, identity matrix $\boldsymbol{I}$, the number of iterations $I_{\max}$, tensor rank $R$, regularization parameters $\lambda_1$, $\lambda_2$ and $\lambda_3$, penalty parameter $\rho$
Ensure: Predict the missing values of $\mathcal{X}$
Initialize $\mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}$ randomly and set $\mathbf{\Theta_A^1}, \mathbf{\Theta_A^2}, \mathbf{\Theta_B^1}, \mathbf{\Theta_B^2} = 0$;
Initialize $\mathbf{B_2}$ and $\mathbf{R}_k$;
for i = $1, \cdots, I_{\max}$ do
    update $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{B_1}$, $\mathbf{B_2}$, $\mathbf{R}_k$, $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, $\mathbf{\Theta_A^1}$, $\mathbf{\Theta_A^2}$, $\mathbf{\Theta_B^1}$, $\mathbf{\Theta_B^2}$ in order;
    if convergence/stopping criterion is met then
        break;
return Dense matrix $\mathcal{X}$
```

Time Complexity Table 2 lists the time complexity of each operation in the update steps of $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{B_1}$, $\mathbf{B_2}$, $\mathbf{R}_k$, $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, $\mathbf{\Theta_A^1}$, $\mathbf{\Theta_A^2}$, $\mathbf{\Theta_B^1}$, $\mathbf{\Theta_B^2}$. $\mathbf{X}_k$ has few non-zeros due to sparse tensor $\mathcal{X}$. Therefore, $\mathbf{X}_k$ can be treated as a sparse matrix. In time complexity analysis, we use $O(pIJ)$ as the time complexity for the matrix multiplication of a sparse matrix $\mathbf{U}$ with a dense $I \times J$ matrix $\mathbf{N}$, where p is the number of non-zeros in $\mathbf{M}$ [28]. In addition, R is a hyper-parameter and is always assigned a very small value, because we adopt low-rank decomposition in tensor/matrix. As a result, updating $\mathbf{A_1}$ and $\mathbf{B_1}$ needs $O(IJKR)/O(pKR \cdot \max\{I, J\})$ time, while updating $\mathbf{A_2}$ and $\mathbf{B_2}$ costs $O(IJR)$ time. $O(\cdot)/O(\cdot)$ respectively represents time complexity when $\mathbf{X}_k$ is treated as a dense matrix and a sparse matrix. Fortunately, in Eq. 15, $\mathbf{S}_A$ and $\mathbf{S}_B$ are both diagonal matrices. We could transform these matrices into vectors to compute $(\mathbf{N}^\mathsf{T}\mathbf{N} + \lambda_3\boldsymbol{I})^{-1}\mathbf{N}^\mathsf{T}$. Consequently, updating $\mathbf{R_k}$ costs $O(IJR)/O(pJR)$ time. The time complexity of updating $\mathbf{\Theta_A^1}$, $\mathbf{\Theta_A^2}$, $\overline{\mathbf{A}}$ is $O(IR)$, and updating $\mathbf{\Theta_B^1}$, $\mathbf{\Theta_B^2}$, $\overline{\mathbf{B}}$ is $O(JR)$. In summary, updating all variables requires $O(IJRK)/O(pKR \cdot \max\{I, J\})$ time. Due to $\max\{K, R\} \ll \min\{I, J\}$, the time complexity is actually much closer to $O(IJ)/O(p \cdot \max\{I, J\})$. Moreover, when $\mathbf{X}_k$ is treated as a sparse matrix, the time complexity is linear in $\max\{I, J\}$. Our implementation relies on NumPy package[2] for the matrix operations.

Experiments

Data

For our experiments, following our previous discussion in Section "Problem Definition", we obtained data from PWKB, VisualGenome, GeoNames, and WordNet. The *part-of* relation data is divided into four classes. Specifically, triples with a located-in *part-of* relation in the training set and validation set are obtained from the GeoNames dataset, while triples with the other three forms of *part-of* relation are extracted from PWKB and VisualGenome as the rest of the training set and validation set. In PWKB and VisualGenome, the method of obtaining triples with the other three different *part-of* relations is to extract the hypernyms of their subjects and objects in WordNet, based on Table 3. Subsequently, the test set is sampled in the same way from the hypernym relation tree in WordNet.

In our datasets, each relation has 5096 triples in the training set and 1699 triples in the test set (based on a ratio of 3:1). In total, the training and test sets consist of 20,384 and 6796 triples, respectively. The amount of entities is 16,677, which means that the tensor is extremely sparse. There are also a few negative (incorrect) instances in the training set due to noisy triples stemming from PWKB and VisualGenome. Triples in the test set are all positive examples, since they come from WordNet.

The training and test data have special characteristics that distinguish them from the knowledge graph completion datasets studied in Section "Problem Definition": (1) The

[2] https://www.numpy.org/

Table 2 Time complexity in the update step of variables

Variables	Computation	Time complexity
$\mathbf{A_1}$	$\mathbf{X}_k\mathbf{B_1}\mathbf{R}_k^\top$	$O(IJR) + O(IR^2)$ if $\mathbf{X}_k$ is dense. $O(pJR) + O(IR^2)$ if $\mathbf{X}_k$ is sparse.
	$\mathbf{R}_k\mathbf{B}_1^\top\mathbf{B_1}\mathbf{R}_k^\top$	$O(JR^2)$
$\mathbf{B_1}$	$\mathbf{X}_k^\top\mathbf{A_1}\mathbf{R}_k$	$O(IJR) + O(JR^2)$ if $\mathbf{X}_k$ is dense. $O(pIR) + O(JR^2)$ if $\mathbf{X}_k$ is sparse.
	$\mathbf{R}_k^\top\mathbf{A}_1^\top\mathbf{A_1}\mathbf{R}_k$	$O(IR^2)$
$\mathbf{A_2}, \mathbf{B_2}$	$\mathbf{P}\mathbf{B_2}, \mathbf{P}^\top\mathbf{A_2}$	$O(IJR)$
	$\mathbf{B}_2^\top\mathbf{B_2}$ and $\mathbf{A}_2^\top\mathbf{A_2}$	$O(JR^2)$ and $O(IR^2)$
$\mathbf{A_1}, \mathbf{B_1}, \mathbf{A_2}, \mathbf{B_2}$	Matrix inversion	$O(R^3)$
$\mathbf{R}_k$	SVD of $\mathbf{A_1}$ and $\mathbf{B_1}$	$O(IR^2)$ and $O(JR^2)$
	$\mathbf{U}_A{}^\top\mathbf{X}_k\mathbf{U}_B$	$O(IJR) + O(IR^2)$ if $\mathbf{X}_k$ is dense.
		$O(pJR) + O(IR^2)$ if $\mathbf{X}_k$ is sparse.
	$(\mathbf{N}^\top\mathbf{N} + \lambda_3\boldsymbol{I})^{-1}\mathbf{N}^\top$	$O(R^2)$
	$\mathbf{V}_A\hat{\mathbf{R}}_k\mathbf{V}_B^\top$	$O(R^3)$
$\mathbf{\Theta}_\mathbf{A}^\mathbf{1}, \mathbf{\Theta}_\mathbf{A}^\mathbf{2}, \overline{\mathbf{A}}$	Matrix addition	$O(IR)$
$\mathbf{\Theta}_\mathbf{B}^\mathbf{1}, \mathbf{\Theta}_\mathbf{B}^\mathbf{2}, \overline{\mathbf{B}}$	Matrix addition	$O(JR)$

subjects and objects between the training set and test set are strictly non-overlapping. (2) The training set has 14.23% of triples matching the pattern in Fig. 1, while the test set has 3.72% of such triples.

Evaluation Procedure

We use the training set for training and parameter tuning. The test set is used to evaluate the model by predicting the relations for test set triples. After running Algorithm 1, the model picks the relation with the maximum value among the corresponding four relations as the output relation. As a metric, the standard accuracy measure is used to evaluate the results on each relation of the test set.

Parameter Tuning Our model has 6 parameters: λ_1, λ_2, λ_3, ρ, R, and $I_{\max}$. We need to select the rank R to represent the number of factors and λ_1, λ_2, λ_3 to control the regularization. Furthermore, $I_{\max}$ is selected to control the maximal number of iterations and is set to 100. The model is constructed based on the training set with different choices for model parameters. We compute the prediction accuracy on the held-out set and pick the parameters that maximize the accuracy. In addition, the minimum accuracy value across the four relations is selected as the basis for parameter tuning. The ranges of λ_1, λ_2, λ_3 are all 0–5. The range of ρ is 0–1 and the range of R is 5–15. We set the remaining parameters to fixed values to reduce their impact on the model performance while tuning each parameter. After that, we select the best overall result of our model, in which the values of parameters are $\lambda_1 = 1.5001$, $\lambda_2 = 0.0001$, $\lambda_3 = 5.5001$, $\rho = 2.5001$, and $R = 10$.

Similarity

There are numerous kinds of similarity computation methods for words. Generally, these methods can be divided into two categories, graph-based similarity measures and word embedding–based similarity computation. Table 4 lists some example words with different similarity scores. Words here are identified as “word.POS.sense”, where POS is the part-of-speech of the word and its sense in the WordNet lexical resource is represented by an integer number. For instance, *dog.n.01* refers to the first sense among all noun senses of the word *dog* listed in WordNet.

Table 3 The hypernyms of subjects and objects

Relations	Hypernyms	
	Subject	Object
Animal part-of	Body part/body substance	Animal
Plant part-of	Plant part/plant structure	Plant
Artifact part-of	Artifact/artefact	Artifact/artefact

Table 4 Example word similarities

Word1	Word2	WUP	LCH	ADW	JCN	LIN	GloVe	Remarks
dog.n.01	*cat.n.01*	0.8667	2.0794	0.3815	0.5374	0.8863	0.8017	Animal
dog.n.01	*apple.n.01*	0.4545	1.1239	0.2346	0.0596	0.1404	0.2634	Animal and plant
dog.n.01	*giraffe.n.01*	0.7742	1.6094	0.3810	0.0000	0.0000	0.3556	IC (giraffe)=0
leg.n.01	*foot.n.01*	0.7000	1.7430	0.5716	0.1839	0.6308	0.6396	
leg.n.03	*table.n.02*	0.7000	1.7430	0.6163	0.0863	0.3728	0.3244	

Graph-Based Similarity Measures

Graph-based similarity measures rely on the graph structure in a lexical knowledge graph such as WordNet. Three different kinds of methods can be distinguished.

Path-Based Similarity Lexical knowledge bases are often based on relational hierarchies, such that there must exist a path between any two words of the same part-of-speech. Typically, these hierarchies are based on semantic attributes and ontological considerations, and hence these paths can directly be exploited by similarity methods. Two methods that achieve this are the Wu & Palmer (WUP) [40] method and the LCH [19] method. As shown in the first and second rows of Table 4, both Wu & Palmer (WUP) metric and the LCH [19] metric succeed at distinguishing similar word sense pairs from less similar ones.

Semantic Random Walks Instead of directly using the paths between two nodes, one can also consider random walks to capture the overlap between their respective neighborhoods. A representative semantics-based method is Align, Disambiguate, and Walk (ADW) [30]. However, we observe that ADW does not succeed at discriminating between positive and negative pairs in Table 4.

IC-Based Similarity Resnik [33] used information content (IC) for computing similarity (RES). We also consider two other representative IC-based methods denoted as JCN [15] and LIN [20]. According to Table 4, these two methods perform as well as path-based similarity measures. However, one drawback is that IC scores are computed from a corpus, in which some rare words may be missing, and obtain IC scores of 0. As shown in the third row of Table 4, the IC for *giraffe.n.01* is zero.

Word Embedding-Based Similarity

Word embeddings allow us to map words to vectors of a fixed length by drawing on distributional co-occurrence patterns on large corpora. Subsequently, we can compute the cosine or Euclidean distance between two word vectors to obtain similarity scores. The two most well-known embedding methods are word2vec [25] and GloVe [29]. However, these methods cannot deal with the problem of polysemy, and sense induction variants also tend to be noisy due to the difficulty of word sense disambiguation. As considered in the 4th and 5th row, one may distinguish two senses of the word *leg*, referring to a human limb (leg.n.01) or to one of the supports of a piece of furniture (leg.n.03). The word vector for *leg*, however, is unique, leading to a low similarity score between *leg* and *table*.

For the sake of fairness, we only experiment with WordNet-based similarity approaches rather than word embedding–based similarity approaches due to the inability of word sense disambiguation and the lack of some low-frequency words.

Baselines

To assess the effectiveness of the proposed method, we compare the results of our approach against three competitive baselines.

Joint Analysis Based on CP Decomposition CP decomposition [16] is a common factorization method, seeking to factorize a tensor into a sum of component rank-one tensors. To evaluate CP decomposition on our task, in our objective function, Z_1 is changed to

$$Z_1 = \frac{1}{2}\|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^{\mathsf{T}}\|_{\mathsf{F}}^2 + \frac{\lambda_1}{2}(\|\mathbf{A}\|_{\mathsf{F}}^2 + \|\mathbf{B}\|_{\mathsf{F}}^2 + \|\mathbf{C}\|_{\mathsf{F}}^2), \tag{18}$$

where $\mathbf{X}_{(1)}$ denotes mode-1 matricization of the $\mathcal{X}$ tensor, $\mathbf{C} \in \mathbb{R}^{K\times R}$ denotes the latent component representation of relations, and $\odot$ denotes the Khatri–Rao product. The subsequent derivation process and algorithm are similar to our model.

Standard RESCAL Method For comparison, we also considered the standard RESCAL decomposition model, in which we disregard the side information matrix.

TransE-Style Models As introduced in the "Related Work" section, models of this sort are well-known approaches

Table 5 Experimental results of our model and baselines. The values in italics represent the maximum accuracy of each category of all models

Models	*Part-of* relation accuracy				
	Animal	Plant	Artifact	Located-in	Average
Our model (WUP similarity)	97.17%	94.52%	92.23%	*100.00%*	95.98%
Our model (LCH similarity)	98.23%	95.94%	96.65%	99.71%	*97.63%*
Our model (ADW similarity)	89.23%	55.80%	78.10%	98.82%	80.49%
Our model (LIN similarity)	72.75%	32.31%	34.37%	*100.00%*	80.48%
Our model (JCN similarity)	51.50%	26.60%	16.30%	38.14%	33.14%
Joint analysis based on CP decomposition	63.86%	68.04%	68.92%	*100.00%*	63.86%
Standard RESCAL decomposition	*100.00%*	0.00%	0.00%	0.00%	25.00%
TransE	0.00%	*100.00%*	0.24%	0.00%	25.06%
TransH	0.00%	0.00%	5.89%	97.35%	25.81%
TransR	0.00%	*100.00%*	0.06%	0.00%	25.01%
CTransR	0.00%	0.00%	*100.00%*	0.00%	25.00%
PTransE_ADD	61.33%	32.25%	0.05%	0.00%	23.41%
PTransE_MUL	24.43%	74.40%	0.06%	0.00%	24.72%
PTransE_RNN	61.09%	46.09%	0.12%	0.00%	26.83%

for knowledge graph completion. We rely on the THU implementation.[3]

Results

In our experiments, we constructed the model and completed the tensor using the parameters tuned via the training set, and then predicted the *part-of* relationships for test set triples. We then use the test set to compute the prediction accuracy comparing the predicted results and the true test set results. Table 5 provides the results achieved by our model and by the baselines in terms of the accuracy on the test set. We observe that our model has a substantial advantage over the baselines.

As evinced by the experimental results in Table 5, our model greatly outperforms the baselines across different kinds of *part-of* relations in terms of the average accuracy. CP factorization can be viewed as factorizing a 3-dimensional tensor into a mathematical operation over three factor matrices, for subject, object, and predicate factors, respectively. Because of the non-overlap of training and test entities and the many separate connected components, the side information provided by the similarity matrix is crucial in providing links between subject words and object words, and plays an important role for relation prediction. However, for CP decomposition, the independent aspects of each different *part-of* relation are not taken into consideration sufficiently well and reflected in the predicate factor matrix, which leads to substandard results in our experiments. Instead, the RESCAL component of our approach deals with the relations by factorizing each slice of the 3-order tensor independently, in a way that can account for the particular differences between relations.

In addition, comparing our model results with relation prediction approaches, we see that the results for the standard RESCAL and TransE-family approaches are particularly unsatisfactory, with many cases of 0% accuracy. It turns out that, without side information, the predicted relation for almost all of the triples in the test set is often the same. It appears that the model selects one or two relations seemingly haphazardly, which is determined by the parameter selection. Due to our graph having (1) numerous isolated connected components, (2) a lack of overlap between the training and test set entities, the model fails to predict the correct relations. Thus, the joint similarity matrix factorization proves crucial in providing the missing connections between entities. Given that (2) means that links exist between the training and test set, one can consider (2) a special case of (1). Hence, overall, what causes bad results for the standard RESCAL and TransE-family approaches is an extreme lack of connections between the entities. For instance, as shown in Fig. 3a, without the link between *dog* and *canine*, a model based on link prediction cannot predict that *nose* is a part of *canine*, because the two entities are disconnected. In the process of training the model, links in the graph serve as constraints. Hence, the trained model fares poorly when there are only few constraints. In our model, as can be seen from Fig. 3b, the similarity matrix serves as a sort of link that bridges the gap between entities. Due to our method's ability to jointly consider the tensor and this similarity matrix, it obtains substantially stronger results than the baselines.

[3] https://github.com/thunlp/KB2E

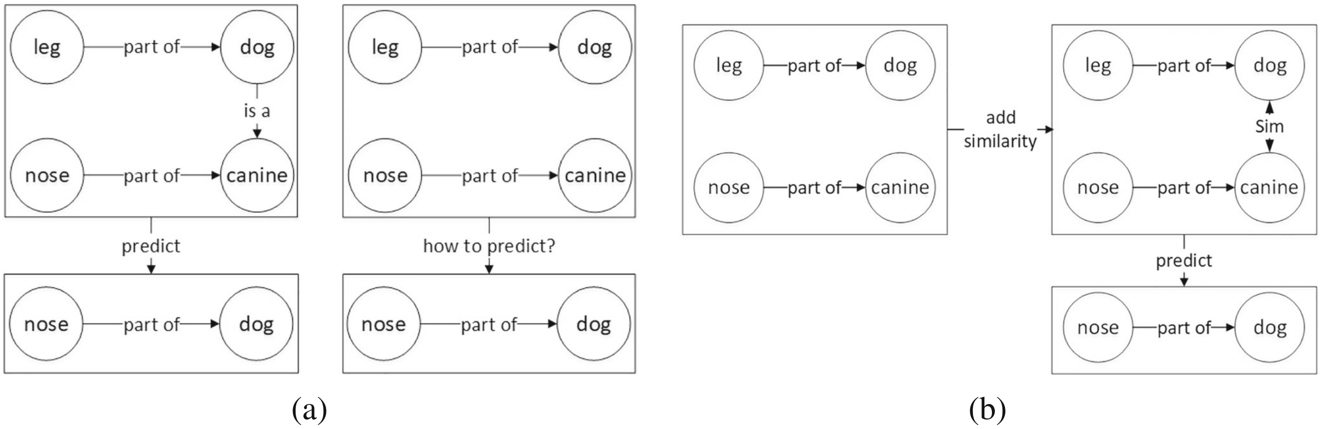

Fig. 3 **a** Knowledge graph completion. The left figure shows an example of the input, while the right side shows the lack of successful link prediction. **b** An example as predicted by our model

Results of Experiments with Other Similarities As shown in Table 5, models with path-based similarity approaches (WUP and LCH) provide the highest levels of accuracy. The accuracy of our model with LCH similarity is higher than when relying on WUP similarity. The main reason is that WUP similarity scores are normalized values ranging from 0 to 1, while LCH similarity ranges from 0 to infinity, which can enlarge the gap between similar word sense pairs and less similar ones. In contrast, ADW similarity obtains inferior results, owing to scores such as for the examples given in Table 4. Finally, due to the lack of IC scores for rare words mentioned in Section "Similarity", models with IC-based similarity (LIN and JCN) obtain very low levels of accuracy.

Error Analysis Some cases of errors of our model results from incorrect generalizations. Given the similarity matrix, the subject "tail.n.01" and object "ant.n.01" are predicted to stand in a *part-of* relationship for the animal dataset. However, it is common sense that ants do not possess tails. We believe that these sorts of errors occur due to incorrect generalizations, possibly because the similarity links are undirected. In future work, one could consider adapting the matrix or incorporating additional constraints to avoid such errors.

Conclusion

This paper presents a cognitively inspired approach for the fusion of knowledge pertaining to the *part-of* relation from heterogeneous sources. Our approach addresses the challenging setting of operating on graphs with large numbers of isolated connected components, and non-overlapping entity sets between the training and test sets, which are not well-supported by existing knowledge graph completion methods such as RESCAL and the TransE family of neural approaches.

Instead, we propose jointly optimizing for tensor factorization along with matrix factorization of a similarity matrix, via an instantiation of the ADMM technique. In our experiments, we find that our method outperforms all baselines with a substantially higher average accuracy.

Funding Information The authors would like to acknowledge the financial support provided by the National Natural Science Foundation of China (no. 61503217)

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not describe any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was not required as no humans or animals were involved.

References

1. Acar E, Rasmussen M, Savorani F, Næs T., Bro R. Understanding data fusion within the framework of coupled matrix and tensor factorizations 129, 53–63. 2013.
2. Bordes A, Glorot X, Weston J, Bengio Y. A semantic matching energy function for learning with multi-relational data. Machine Learning. To appear. 2013.
3. Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: NIPS; 2013. p. 2787–2795.
4. Boyd SP, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning. 2011;3(1):1–122.
5. Chew PA, Bader BW, Kolda TG, Abdelali A. Cross-language information retrieval using PARAFAC2. In: SIGKDD; 2007. p. 143–152.

6. Correa FE, Oliveira MDB, Gama J, Corrêa PLP, Rady J. Analyzing the behavior dynamics of grain price indexes using tucker tensor decomposition and spatio-temporal trajectories. Comput Electron Agric. 2016;120:72–78.
7. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: SIGKDD; 2014. p. 601–610.
8. Dong XL, Gabrilovich E, Heitz G, Horn W, Murphy K, Sun S, Zhang W. From data fusion to knowledge fusion. PVLDB. 2014;7(10):881–892.
9. Dong XL, Srivastava D. Knowledge curation and knowledge fusion: challenges, models and applications. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, Melbourne, Victoria, Australia, May 31 - June 4, 2015; 2015. p. 2063–2066.
10. Du J, Yuan C, Tian P, Lin H. Channel estimation for multi-input multi-output relay systems using the PARATUCK2 tensor model. IET Commun. 2016;10(9):995–1002.
11. Fellbaum C, (ed). 1998. WordNet: an electronic lexical database. http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20%&path=ASIN/026206197X. Cambridge: The MIT Press.
12. He L, Liu B, Li G, Sheng Y, Wang Y, Xu Z. Knowledge base completion by variational Bayesian neural tensor decomposition. Cogn Comput. 2018;10(6):1075–1084.
13. He S, Liu K, Ji G, Zhao J. Learning to represent knowledge graphs with gaussian embedding. In: CIKM. ACM; 2015. p. 623–632.
14. Ji G, Liu K, He S, Zhao J. Knowledge graph completion with adaptive sparse transfer matrix. In: AAAI; 2016. p. 985–991.
15. Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. arXiv:cmp-lg/9709008. 1997.
16. Kiers H. Towards a standardized notation and terminology in multiway analysis. 2000;14, 105–122.
17. Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Rev. 2009;51(3):455–500.
18. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, Bernstein M, Fei-Fei L. Visual genome: connecting language and vision using crowdsourced dense image annotations. arXiv:1602.07332. 2016.
19. Leacock C, Chodorow M. Combining local context and wordnet similarity for word sense identification. WordNet: an electronic lexical database. 1998;49(2):265–283.
20. Lin D. An information-theoretic definition of similarity. In: ICML; 1998.
21. Lin Y, Liu Z, Luan H, Sun M, Rao S, Liu S. Modeling relation paths for representation learning of knowledge bases. Computer Science. 2015.
22. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: AAAI; 2015. p. 2181–2187.
23. Lin Y, Liu Z, Zhu X, Zhu X, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: AAAI; 2015. p. 2181–2187.
24. Liu B, He L, Li Y, Zhe S, Xu Z. Neuralcp: Bayesian multiway data analysis with neural tensor decomposition. Cogn Comput. 2018;10(6):1051–1061.
25. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781. 2013.
26. Nengfu X, Wensheng W, Xiaorong Y, Lihua J. Rule-based agricultural knowledge fusion in web information integration. NJAS - Wageningen Journal of Life Sciences. 2012;10(1):635–638(4).
27. Nickel M, Tresp V, Kriegel H. A three-way model for collective learning on multi-relational data. In: ICML; 2011. p. 809–816.
28. Nickel M, Tresp V, Kriegel H. Factorizing YAGO: scalable machine learning for linked data. In: WWW; 2012. p. 271–280.
29. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: EMNLP; 2014. p. 1532–1543.
30. Pilehvar MT, Jurgens D, Navigli R. Align, disambiguate and walk: a unified approach for measuring semantic similarity. In: ACL; 2013. p. 1341–1351.
31. Preece AD, Hui K, Gray WA, Marti P, Bench-Capon TJM, Jones DM, Cui Z. The KRAFT architecture for knowledge fusion and transformation. Knowl.-Based Syst. 2000;13(2-3):113–120.
32. Ragusa E, Gastaldo P, Zunino R, Cambria E. Learning with similarity functions: a tensor-based framework. Cogn Comput. 2019;11(1):31–49.
33. Resnik P. Using information content to evaluate semantic similarity in a taxonomy. arXiv:cmp-lg/9511007. 1995.
34. Solé-Casals J, Caiafa CF, Zhao Q, Cichocki AS. Brain-computer interface with corrupted EEG data: a tensor completion approach. Cogn Comput. 2018;10(6):1062–1074.
35. Stegeman A, Berge JT, Psychometrika LDL. Sufficient conditions for uniqueness in candecomp/parafac and indscal with random component matrices. Psychometrika. 2006;71(2):219–229.
36. Tandon N, Hariman C, Urbani J, Rohrbach A, Rohrbach M, Weikum G. Commonsense in parts: mining part-whole relations from the web and image tags. In: Proceedings of the thirtieth AAAI conference on artificial intelligence, February 12–17, 2016, Phoenix, Arizona, USA; 2016. p. 243–250.
37. Thoma S, Rettinger A, Both F. Knowledge fusion via embeddings from text, knowledge graphs, and images. arXiv:1704.06084. 2017.
38. Wang Y, Widrow B, Zadeh LA, Howard N, Wood S, Bhavsar VC, Budin G, Chan CW, Fiorini RA, Gavrilova ML, Shell DF. Cognitive intelligence: deep learning, thinking, and reasoning by brain-inspired systems. IJCINI. 2016;10(4):1–20.
39. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: AAAI; 2014. p. 1112–1119.
40. Wu Z, Palmer MS. Verb semantics and lexical selection. In: ACL; 1994. p. 133–138.
41. Yu Xl, Qiao L. Knowledge fusion methods: a survey. DEStech Transactions on Computer Science and Engineering (smce). 2017.
42. Zhang J, Han Y, Jiang J. Tucker decomposition-based tensor learning for human action recognition. Multimedia Syst. 2016;22(3):343–353.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.