


<b>ITC 4/49</b> <b>Information Technology and Control</b> <b>Vol. 49 / No. 4 / 2020</b> <b>pp. 622-642</b> <b>DOI 10.5755/j01.itc.49.4.25366</b>	<b>Integrated Embedding Approach for Knowledge Base Completion with CNN</b>	
	Received 2020/02/27	Accepted after revision 2020/10/13
	 <a href="http://dx.doi.org/10.5755/j01.itc.49.4.25366">http://dx.doi.org/10.5755/j01.itc.49.4.25366</a>	

**HOW TO CITE:** Chen, S., Xie, S., Chen, Q. (2020). Integrated Embedding Approach for Knowledge Base Completion with CNN. *Information Technology and Control*, 49(4), 622-642. <https://doi.org/10.5755/j01.itc.49.4.25029>

# Integrated Embedding Approach for Knowledge Base Completion with CNN

## Samuel Chen

Faculty of Information Technology; Melbourne Polytechnic; Preston Victoria 3072; Australia;  
e-mail: samuelchen@melbournepolytechnic.edu.au

## Shengyi Xie

Experimental Training Center; Fujian Vocational College of Agriculture; Fuzhou 350181; Fujian;  
phone: +86 153 5673 2821; e-mail: fjnyxsy@qq.com

## Qingqiang Chen

Information Science and Engineering College; Fujian University of Technology; Fuzhou 350118; Fujian;  
phone: +86 131 7811 4567; e-mail: 516040610@qq.com

Corresponding author: 516040610@qq.com

In knowledge graph embedding, sophisticated models may suffer from over-fitting and high computational costs. On the contrary, transitional models come with lower complexity but struggle with complex relations while integrating relational attributes or semantic information could help with embedding representation. Convolutional neural networks employed in recent researches are able to model interactions between entities and relations efficiently but may ignore global dependencies. To tackle such problems, a model called Integrated Embedding Approach for Knowledge Base Completion (IEAKBC) is proposed. In this model, embedding representations of entities and relations are put together to constitute a three-column,  $k$  dimensional matrix for each triplet. Afterwards, features from different relations are integrated into head and tail entities thus forming fused triplet matrices. Both sets of matrices are used as inputs to a convolutional neural network (CNN) framework. In CNN, kernels go over each row of the matrices for feature extraction. Feature maps are subsequently concatenated and weighted for output scores to discern whether the original triplet holds or not.

Experiments on four benchmark datasets show that our model performs well on complex relations while retaining transitional characteristics. Finally, we apply the model to a personalized search application, verifying its practicality in real-world scenarios.

**KEYWORDS:** Knowledge Representation, Knowledge Base Completion, CNN, Translation Mechanism, Link Prediction.

## 1. Introduction

Knowledge Base (KB) [35] encodes structured facts in the form of triples, each of which is composed of a head entity, a relation and a tail entity, denoted as (h, r, t). In knowledge graph (KG) structure, nodes refer to head entities h and tail entities t, while edges are relations r between nodes. KBs, such as Freebase [5], WordNet [19] and NELL [41], are widely used in semantic searches [77], question answering [7, 23], visual detection [1], etc.

There are a lot of missing facts or incomplete triples in existing KBs, lacking entities or relations [72]. Knowledge Base Completion (KBC) [51] aims to solve this problem by inferring new facts with known information, or by introducing external resources. Embedding methods are among the mainstream research directions for KBC; after entities / relations are embedded into tensor / matrix / vector spaces, local / global patterns and semantic features of facts are extracted with tensor / matrix factorization or vector translation / rotation operations so as to handle downstream tasks such as link prediction and triple classification. Classic embedding models include RESCAL [48-50], NTN [12], TransE [8], DistMult [78], etc.

Relations of triples could be divided into four categories according to their cardinalities, 1-1, 1-M, M-1 and M-M [8, 42]. The last three kinds of relations are usually referred to as complex relations. Predicting head on 1-1 and 1-M relations and predicting tail 1-1 and M-1 relations are called prediction on “side 1” of triples; predicting head on M-1 and M-M relations and predicting tail on 1-M and M-M relations are called prediction on “side M” of triples, which are obviously more difficult. Relations could also be classified as symmetric, antisymmetric, etc. It is not rare that simple assumptions (e.g., TransE) perform well on relations of 1-1 type. However, it usually goes beyond TransE’s capabilities to appropriately learn complex relationships. Most of subsequent models focus on integrating various role information brought by differ-

ent relational attributes into entity representations for improvement. Lao et al. [33-34] and Takahashi et al. [59] verify the effectiveness of applying contextual information contained in relationship paths to KBC, demonstrating the value of relational features from another perspective. However, exquisitely designed models with stronger expressivity often come with higher computational costs. In ConvE [15] a parameter-efficient convolutional neural network (CNN) is employed to extract merely local patterns of triplets, yet yielding state-of-the-art (SOTA) performance.

Our initiative is to integrate diverse semantics brought by multiple relational attributes into entity representations with a simple but highly efficient CNN framework while following the idea of TransE, which is, if triplet (h, r, t) holds, the corresponding vector representations of its elements  $v_h, v_r, v_t$  should comply with one rule:  $v_h + v_r \approx v_t$ . In IEAKBC, firstly entities / relations are embedded into low-dimensional vectors, i.e., each original triplet is represented in the form of a three-column,  $k$  dimensional matrix. Then the relation features are integrated into head / tail entities, forming feature-fused matrices of the same size used as another input channel for convolution; afterwards local / global patterns and semantic features are extracted in CNN, and finally the score is computed to verify the validity of the triplet. We perform link prediction and triple classification tasks on benchmark datasets, making comparison between IEAKBC and several mainstream models.

In addition, IEAKBC is applied to a search personalization problem with the objective that ranking orders of search results could meet users’ personal preferences accurately.

In Section 2 relevant work is discussed, in Section 3 IEAKBC model is proposed, in Section 4 experimental results are presented and analyzed, in Section 5 application of IEAKBC in the personalized search system is discussed, and in Section 6 we conclude and discuss future plans.

## 2. Related Work

Multiple literature surveys [13, 45-46, 54, 70] have been conducted on knowledge graph embedding models. Such models based on latent features learn low-dimensional representations of nodes and edges in KGs and can be roughly divided into three categories: tensor / matrix factorization based approaches, translation / rotation based methods and models that employ neural networks.

### 2.1. Tensor / Matrix Factorization Based Approaches

Such approaches utilize semantic similarities. RESCAL, a bilinear model computes the outer product of head / tail entity and relation matrix with Alternating Least Squares (ALS) updating entities and relations alternatively. TATEC [21] adds two way interactions to RESCAL and brings forward an idea that entity representations could be relation specific. Subsequent models expect to acquire a balance between complexity, performance and scalability. DistMult inherits the merits of NTN and TransE to represent relations as diagonal matrices and entities vectors, taking Hadamard product of entity / relationship embeddings as the triplet score. However, since the result is not affected by the order of parameters, DistMult could not identify anti-symmetric relational patterns. ComplEx [64] disables the commutativity by embedding entities / relations into complex spaces. DistMult and ComplEx could be regarded as dimension reduction versions of RESCAL. Other SOTA models include HolE [47], Simple [28], ANALOGY [38], Tucker [63], etc.

### 2.2. Translation / Rotation Based Methods

Translation / rotation based methods use distance based score functions. SE [9, 6] takes the idea that if one triplet holds, the head mapping should be close to the tail vector in the relation specific subspace. However, the correlation between entities and relations is weak and costs of optimization on matrix projection are rather high.

TransE, which is designed based on the ideas that hierarchical relationships could be naturally represented by translations and in word embedding studies some 1-1 relationships may be represented by translations in embedding spaces as well, maps the relations as translational vectors and uses a margin-based

ranking criterion with a dissimilarity scoring function  $d = \|\mathbf{v}_h + \mathbf{v}_r - \mathbf{v}_t\|_p$ , in which  $\|\cdot\|_p$  is the L1 or L2 norm for  $p \in \{1, 2\}$ . Stochastic Gradient Descent (SGD) is employed for parameter updating iteratively. Local features of triplets are preserved in the mapping of the same dimension of vectors. TransE featuring effective translations and model simplicity often performs better than SE despite its lower expressivity. However, TransE is inappropriate for learning complex relationship types due to the convergence problem, i.e., entity representations tend to be similar or even identical even though the discrepancy of their semantics could be significant. TransE is widely applied for embedding initialization. For instance, knowledge validity is often constrained by time and Jiang et al. [27] learns the temporal features of relationships with TransE. Lin et al. [36] and Luo et al. [40] further improve the model expressivity by combining semantic information from relation paths with TransE.

Variants of TransE lay stress on embedding mechanisms. TransH [71] maps entities to relation specific hyperplanes to embody their role differences. In TransR [37] the hyperplanes are replaced with relation specific matrices and the semantic spaces of entities / relations are separated to enhance expressivity. STransE [43] further addresses the problem that head / tail entities of different attributes still share the same relational projections. There are more such models [76].

However, higher expressivity is accompanied by increasing number of parameters [80]. TransD [25] decomposes the projection matrices into vectors for simplification while TransSparse [26] introduces sparse matrix to address the imbalance in head / entity proportions.

Related models include TransM [18], ManifoldE [75], FT [20] and TransA [74], improving expressivity by relaxing the constraints imposed on distance based scoring functions. TorusE [17], RotatE [58] and CrossE [81] employ a torus, rotations and composite operations to improve expressivity respectively. TransG [22] and KG2E [24] map entities and relations into to random variables according to their uncertainties.

### 2.3. Models with Neural Networks

NTN takes the strengths of multiple models, in which head / tail entity representations are concatenated as input and bilinear tensors are employed to replace the

traditional linear transformation layer. NTN holds the strongest expressivity in almost all models. However, with high computational costs it is difficult to appropriately train NTN model, over-fitting frequently spotted especially in sparse or small KGs. ProjE [57] could be regarded as a simplified NTN.

In recent years, CNN, originally designed for computer vision [65] with less parameters than fully connected neural networks, is in the spotlight of Natural Language Processing (NLP) [30, 56]. ConvE is the first to apply CNN to knowledge representation. ConvE argues that a padded two-dimensional (2D) convolution on 2D reshaping of concatenated  $\mathbf{v}_h$  and  $\mathbf{v}_r$  could model more interactions and extract more features than previous models without increasing the embedding size or number of parameters, thus lowering probability of memory overflow. ConvE is also considered to be more expressive than HolE, since it captures non-linear features. The authors find that filters with smaller sizes could better extract local patterns but also report predictive accuracy degradation with one-dimensional (1D) convolution. Presumably, it is related to the lack of relational attribute integration. Unintuitively the convolving across reshaped embeddings promotes such integration [2], while 1D convolution fails to do so. In addition, ConvE does not observe the valuable transitional characteristic employed by TransE and its multiple variants.

There are several studies aiming at simplifying ConvE with 1D convolution, including Hyper [2] and ConvKB [42], the latter hoping to improve ConvE by generalizing transitional characteristics, i.e., by capturing global relationships among same dimensional entries of vector representations of triplet elements. While there's obvious improvement on specific KBC tasks, performance inconsistency is observed, for which our hypothesis is as discussed above: lack of relational integration. While 1D convolution takes care of the global dependencies, in ConvKB entities and relations are represented separately, leading to the loss of various properties of entities pairing with different relations. CapsE [44], a recent study, addresses this problem by employing a capsule layer to process the output from the hidden layer. SACN [55] also extends ConvE by learning relation path representations while retaining translational features.

Since complex methods are potentially subject to either over-fitting or under-fitting, we hope to bring

forward a simple but effective solution that could achieve trade-offs between accuracy and scalability for multi-relational domains.

We consider the multiplicative similarity calculation employed by DistMult and similar models are essentially a way to extract potential semantics; on the other hand, feature extraction based on transitional distance is highly efficient and adept at dealing with sparse datasets while solutions to the convergence problem adopted by multiple successors to TransE are in nature also extracting semantic attributes. We hope to apply this idea to our model.

The CNN framework could extract more non-linear features than shallow models [15]. However, compared to the 2D kernels with larger receptive fields, more 1D kernels with smaller sizes may better recognize latent patterns and reduce computational costs.

Comprehensively speaking, our initiative is to adopt a CNN framework with 1D convolution, succeed the idea of transitional constraint and combine embeddings with relation specific integration so as to address the problem of semantic loss, stabilizing model performance while keeping the computational cost reasonable in real world scenarios.

---

### 3. Integrated Embedding Approach for Knowledge Base Completion (IEAKBC)

In KB  $\mathcal{G}$ , triplets denoted as  $(h, r, t)$  refer to facts containing semantic information, where  $h, t \in \mathcal{E}$ ,  $r \in \mathcal{R}$  and  $\mathcal{E}, \mathcal{R}$  refer to a set of entities and relations, respectively. The objective of model design is to find a reasonable score function to measure the plausibility or implausibility for triplets.

Here we use  $k$  to represent the dimensionality of embeddings of entities and relations produced by TransE, so the original triplet could be displayed in the form of a matrix  $\mathbf{A} = [\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t] \in \mathbb{R}^{k \times 3}$ , and  $\mathbf{A}_{i,:} \in \mathbb{R}^{1 \times 3}$  means the  $i$ th row of  $\mathbf{A}$ . Representations of entities are independent of their position as well as combination with different relations so information is propagated among multiple triplets, i.e., global dependency is retained while local patterns are kept in different entries of entity and relation representations.

Our core feature is an operation named relational fusion for incorporating relational features into em-

beddings of head and tail entities so as to retain the integrity of triplets, i.e., to keep the complex pairing condition between entities and relations. The formula is as follows, in which  $\mathbf{v}'_h$  is the head entity representation after relational fusion,  $\mathbf{v}_h$  is the original  $k$ -dimensional vector produced by TransE,  $\cdot$  means dot product operation,  $\mathbf{w}_l$  refers to parameter vectors gained through learning processes in a feedforward neural network,  $\mathbf{v}_r$  represents the original  $k$ -dimensional relation vector and  $b_l$  is a bias term. For tail entity representation, the process is alike. Set  $\mathbf{v}'_r = \mathbf{v}_r$ . We hope this set of procedures could help to implant relational features into entities so that same entities with variant properties for combination with different relations could be expressed more affluently, covering all kinds of relations with higher expressivity.

$$\mathbf{v}'_h = \mathbf{v}_h \cdot (\mathbf{w}_l \cdot \mathbf{v}_r + b_l). \quad (1)$$

After the relational integration, we get  $\mathbf{A}'$ , variation of  $\mathbf{A}$  as the second input channel with relation attributes embedded into entities, defined as follows.

$$\mathbf{A}' = [\mathbf{v}'_h, \mathbf{v}'_r, \mathbf{v}'_t] \in \mathbb{R}^{k \times 3}. \quad (2)$$

Given  $\boldsymbol{\omega} \in \mathbb{R}^{1 \times 3}$  represents the filters operating on the two channels with multiple kernels allocated to each

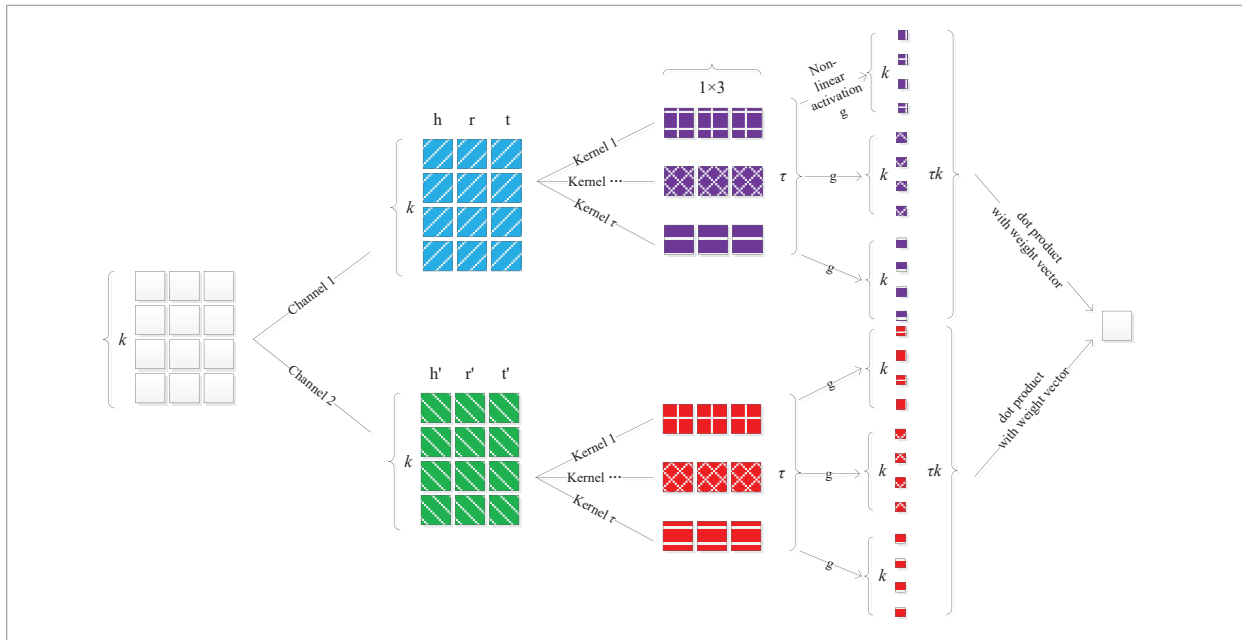
channel, and what  $\boldsymbol{\omega}$  does is to repeatedly go over each row of  $\mathbf{A}$  and  $\mathbf{A}'$ , two channels of the input to generalize the transitional characteristics from embedding representations, to extract global relationships among the same dimensional entries of triplets, and to analyze the variations from entities with different properties due to changing relations. Feature map  $\mathbf{v}$  after convolution processes could be denoted as  $\mathbf{v} = [v_1, v_2, \dots, v_k]$ , formula listed as follows in which  $g$  is a non-linear activation function such as sigmoid and  $b_2$  is a shared bias term while parameters in kernels are different. Here we do not differentiate between  $\mathbf{A}_{i,:}$  and  $\mathbf{A}'_{i,:}$  any more, since they are regarded as two channels processed parallel

$$v_i = g(\boldsymbol{\omega} \cdot \mathbf{A}_{i,:} + b_2). \quad (3)$$

Let  $\Omega$  and  $\tau$  be the kernels of each channel and the number of such kernels, i.e.,  $\tau = |\Omega|$ . Therefore, for each channel there will be  $\tau$  feature maps that are concatenated into one single vector  $\in \mathbb{R}^{\tau k \times 1}$  respectively and processed by a shared weight vector  $\mathbf{w}_2$  via a dot product to generate scores for triplets. Formally, our score function  $f$  could be defined as follows, with  $*$  denoting convolutional operations and 'concat' the concatenation operation. Finally, scores from different channels are combined together to judge the validity of triplets. The whole process is shown in Figure 1.

Figure 1

Process of CNN in IEAKBC





$$f(h, r, t) = \text{concat}(g([\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t] * \Omega)) \cdot \mathbf{w}_2. \quad (4)$$

Loss function is defined as follows, in which

$$l_{(h,r,t)} = \begin{cases} 1, & (h, r, t) \in \mathcal{G} \\ -1, & (h, r, t) \in \mathcal{G}' \end{cases}, \quad \mathcal{G}' \text{ refers to the set of invalid}$$

triplets created by replacing entities of valid triplets in  $\mathcal{G}$  and  $\theta$  denotes the embeddings and parameters acquired by learning processes.

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G} \cup \mathcal{G}'} \log(1 + \exp(l_{(h,r,t)} \cdot f(h, r, t))) + \frac{\lambda}{2} \|\theta\|_2^2. \quad (5)$$

The learning process of IEAKBC is displayed in Algorithm 1; space complexity is  $O(n_e k + n_r k)$ , same as TransE and ConvE while time cost is theoretical within the same order of magnitude of that with TransE.

**Algorithm 1:** Parameter Optimization for IEAKBC

Input: KB  $\mathcal{G}$ , entity set  $\mathcal{E}$ , relation set  $\mathcal{R}$ , embedding dimension  $k$ , kernels  $\Omega$ , batch size  $b$ , weight  $\mathbf{w}_2$ , regularizer  $\lambda$ , pre-trained embeddings produced by TransE

- 1 Initialize  $(\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t)$   
// Using a truncated normal distribution or using [0.1, 0.1, -0.1]
- 2 Initialize  $\omega$
- 3  $\mathbf{w}_2 \leftarrow \text{uniform}(-\frac{\sqrt{6}}{\sqrt{k \times \tau + 1}}, \frac{\sqrt{6}}{\sqrt{k \times \tau + 1}})$
- 4 **For**  $i = 1, 2, \dots, n$ , //  $n$  denotes the upper limit of epochs
- 5   **For**  $j = 1, 2, \dots, \frac{|\mathcal{G}|}{b} + 1$  //  $|\mathcal{G}|$  denotes the number of triplets
- 6     Sample( $\mathcal{G}, b$ )
- 7     InvalidBatch =  $\emptyset$
- 8     **For** each triplet in Sample( $\mathcal{G}, b$ )
- 9       InvalidSample() // sample corrupted triplets denoted as  $(h', r, t')$
- 10       InvalidBatch  $\leftarrow$  InvalidBatch  $\cup (h', r, t')$
- 11       Batch  $\leftarrow$  Sample( $\mathcal{G}, b$ )  $\cup$  InvalidBatch
- 12       **For** each triplet  $\in$  Batch
- 13           $f(h, r, t) = \text{concat}(g([\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t] * \Omega)) \cdot \mathbf{w}_2$
- 14          Compute  $l_{(h,r,t)}$
- 15          Compute gradient  $\nabla \mathcal{L}_{\text{Batch}}$
- 16       Update embeddings,  $\mathbf{w}_2$  and  $\Omega$  with regard to  $\nabla \mathcal{L}_{\text{Batch}}$

To summarize, IEAKBC combines global relationships with relational attributes with a CNN framework, aiming at retaining triplet element correlations as much as possible. Therefore, it could be seen as an extension of TransE, a variant of ConvE with 1D convolution.

## 4. Experiments and Analysis

**Tasks:** Two common KBC tasks including link prediction and triple classification are used to measure model performance.

**Datasets:** Four benchmark datasets are used, FB15k-237 and WN18RR for link prediction, WN11 and FB13 for triple classification. All of the four datasets are specially designed for multi-relational tasks and frequently adopted in related studies.

WN18RR and WN11 are extracted from WordNet, a lexical KG in English in which each entity denotes a synset consisting of several words and corresponds to a distinct word sense. Relationships in WordNet are defined as conceptual-semantic and lexical relations. In WN11, in the case of synsets containing multiple words, only the most frequent one is picked.

FB15k-237 and FB13 derive from Freebase, a large KG of general world facts. Triples in FB13 come from the People domain, containing 13 relations, six of which are removed from the test set owing to high predicting difficulty.

Test leakage<sup>1</sup> in FB15k and WN18 discussed by Toutanova et al. [62] and Dettmer et al. [15] leads to the birth of more robust and challenging datasets, FB15k-237 and WN18RR. On the other hand, WN11 and FB13 do not suffer from such problem. WN18RR is denser than FB15k-237 (the average number of entity pairs per relation is bigger). FB15k-237 and WN18RR contain only positive triples while WN11 and FB13 include negative samples. Details of datasets are shown in Table 1.

**Table 1**

Dataset Statistics

Datasets	#E	#R	Training Set	Validation Set	Test Set
FB15k-237	14541	237	272115	17535	20466
WN18RR	40943	11	86835	3034	3134
WN11	38696	11	112581	2609	10544
FB13	75043	13	316232	5908	23733

<sup>1</sup> Models with simple rules could achieve high scores on specific datasets with high proportions of inverse relations.

#### 4.1. Link Prediction

**Task:** The goal is to infer a missing head / tail entity given a relation and the other entity, denoted as  $(?, r, t)$  or  $(h, r, ?)$ , results of which are made by ranking the scores of candidate triples produced by various score functions.

**Models:** DistMult, ComplEx, ConvE, TransE and ConvKB are used as baselines for comparison with IEAKBC.

**Evaluation Protocol:** The valid test triple and corrupted triples are ranked together and we hope correct answers can be ranked before incorrect ones. **MR** (mean rank), **MRR** (mean reciprocal rank) and **Hits@10** (the proportion of correct entities ranked in the top 10, which is commonly used in downstream tasks such as personalized search) are taken as evaluation metrics. Lower MR, higher MRR and Hits@10 indicate better performance. **Filtered** setting protocol is used, i.e., false-negative triples that already appear either in the training, validation or test set are not taken into account.

**Training Protocol:** Following Bordes et al. [7], entity replacement is performed to create corrupted triplets. The Bernoulli Trick (BT) [71] is employed for all models except ConvE (for ConvE the 1-N strategy as a fea-

ture procedure is kept) to create corrupted triples while reducing the probability of generating false-negative samples. Negative sampling is performed at runtime for each batch. Training is of up to 500 / 2000 epochs (for different models) and early stopping is executed if MRR improvement of last 10 epochs is less than 0.01. Models with highest Hits@10 scores on the validation set are implemented on the test set.

**Hyper-parameters Tuning:** Grid search is employed to find optimal hyper-parameters; the parameter pool is listed as follows, some of which are model specific and with annotations. Optimal performance is acquired with hyper-parameters listed in Table 2.

Embedding Size  $\in \{50, 100, 200, 500, 1000\}$

Loss Function  $\in \{\text{MR (margin ranking), CE (cross entropy)}\}$

Optimizer  $\in \{\text{SGD, AdaGrad [16], Adam [31]}\}$

Batch Size  $\in \{64, 128, 256, 512, 1024\}^2$

Initial Learning Rate (SGD)  $\in \{1e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}, 1e^{-1}\}$

Initial Learning Rate(AdaGrad)  $\in \{0.002, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5\}$

Initializer in PyTorch  $\in \{\text{normal, uniform, xavier\_normal, xavier\_uniform, truncated\_normal, [0.1, 0.1, -0.1]}\}$

**Table 2**

Hyper-parameters for Optimal Performance in Link Prediction Task<sup>3</sup>

Hyper-parameters	FB15k-237		WN18RR	
	TransE	IEAKBC	TransE	IEAKBC
Embedding Size	100	100	50	50
Negative Sampling	BT	BT	BT	BT
Loss Function	MR	MR	MR	MR
Optimizer	SGD	Adam	SGD	Adam
Batch Size	128	256	128	256
Learning Rate	0.0005	0.0005	0.0005	0.0005
Regularizer	L1	L2	L1	L1
Initializer	xavier_normal	[0.1, 0.1, -0.1]	xavier_normal	trun
margin $\gamma$	1	-	5	-
kernel number	-	100	-	500

<sup>2</sup> Due to experimental limitations, the batch size is set to between 64 and 1024 for fear of memory overflow.

<sup>3</sup> trun is short for truncated\_normal

Initial Learning Rate (Adam)  $\in \{1e^{-6}, 5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}, 1e^{-2}\}$

Regularizer  $\in \{L1, L2, L3 [32]\}$

Margin  $\gamma \in \{1, 2, 3, 5, 7, 10\}$  (for TransE)

Kernel Number  $\tau \in \{100, 200, 300, 400, 500\}$  (for IEAKBC)

Others: Maximum Epoch=500 (for IEAKBC); Maximum Epoch=2000 (for TransE); Regularizer Coefficient=0.001 (for IEAKBC)

#### 4.1.1. Overall Results

Experimental results are shown in Table 3, where [\*] means that data are taken from related literature, since previous experiments are carried out on the same datasets. Results of DistMult, ComplEx and ConvE are from Dettmers et al. [15]. Results of ConvKB are from Nguyen et al. [42]. We use our own implementation of TransE and embeddings produced by TransE are employed as initialization embeddings for IEAKBC.

In a nutshell, IEAKBC secures the best Hits@10 score on FB15k-237 and No.2 at MRR, while achieving the best results at MRR and Hits@10 on WN18RR, No.2 at MR.

MRR scores of DistMult and ComplEx on WN18RR are rather high, since bilinear models are good at generalizing semantic similarity from denser datasets with multiplicative operations. However, the MRR and Hits@10 scores of these two models on FB15k-237 are

not so good, since the number of entity pairs per relation is relatively small in sparse datasets and these models may not learn sufficient information.<sup>4</sup>

On the contrary to bilinear models, TransE excels at handling sparse datasets, achieving better performance at MRR and Hits@10 on FB15k-237. Nevertheless, TransE suffers an obvious degradation at MRR on WN18RR due to the representation restrictions.

Although the neural network mechanism is still a black box, ConvE obtains solid scores at all metrics except at MR on WN18RR and we consider its advantages over DistMult and ComplEx at MRR and Hits@10 on FB15k-237 are due to the 2D convolution extracting richer relation specific features. However, the 2D kernel size (3×3) along with complicated score function may bring about a relatively high computational cost.

On FB15k-237, there is comprehensive improvement at all metrics for ConvKB compared with TransE, since it takes embeddings produced by TransE as input and further generalizes transitional relationships in a CNN framework. However, when it comes to datasets with high proportion of complex relations, ConvKB also suffers a drop at MRR on WN18RR, since it does not solve the problem of entity representations tending to be close.

On FB15k-237, IEAKBC narrowly loses to ConvE, DistMult and ConvKB at MR but obviously beats TransE with an improvement of 78. At MRR which

**Table 3**

Results of Link Prediction on FB15k-237 and WN18RR<sup>5</sup>

Method	FB15k-237			WN18RR		
	MR	MRR	Hits@10(%)	MR	MRR	Hits@10(%)
DistMult[*]	<u>254</u>	0.241	41.9	5110	0.43	49
ComplEx[*]	339	0.247	42.8	5261	<u>0.44</u>	51
ConvE[*]	<b>246</b>	0.316	49.1	5277	<b>0.46</b>	48
TransE	343	0.297	46.5	3109	0.236	51.3
ConvKB[*]	257	<b>0.396</b>	<u>51.7</u>	<b>2554</b>	0.248	<u>52.5</u>
IEAKBC	265	<u>0.374</u>	<b>57.0</b>	<u>2762</u>	<b>0.460</b>	<b>53.8</b>

<sup>4</sup> We also notice that DistMult performs better than ConvKB and IEAKBC at MR on FB15k-237, so one of our plans is to integrate merits of bilinear models into IEAKBC.

<sup>5</sup> The best performance is marked in bold while No.2 in italic font with underlines.



is a more stable metric, IEAKBC outperforms all the other models except ConvKB with a wide margin implying the model retains transitional constraint and relational integration could extract more features. IEAKBC's best performance is at Hits@10 with about 10% increase compared to ConvKB, let alone ConvE and TransE, indicating the probability of applying IEAKBC to tasks such as search personalization.

On WN18RR, the denser dataset, IEAKBC achieves a similar result to ConvKB at MR, steadily outperforming some other models. At MRR, IEAKBC is on par with ConvE, solving the problem of performance slump confronting TransE and ConvKB who rely solely on transitional characteristics, and also better than DistMult and ComplEx indicating the feasibility

of extracting latent features by fusing relational attributes instead of computing entity similarity. At Hits@10 IEAKBC achieves the highest score again, slightly overpassing ConvKB. To summarize, IEAKBC proves itself a stable and effective model.

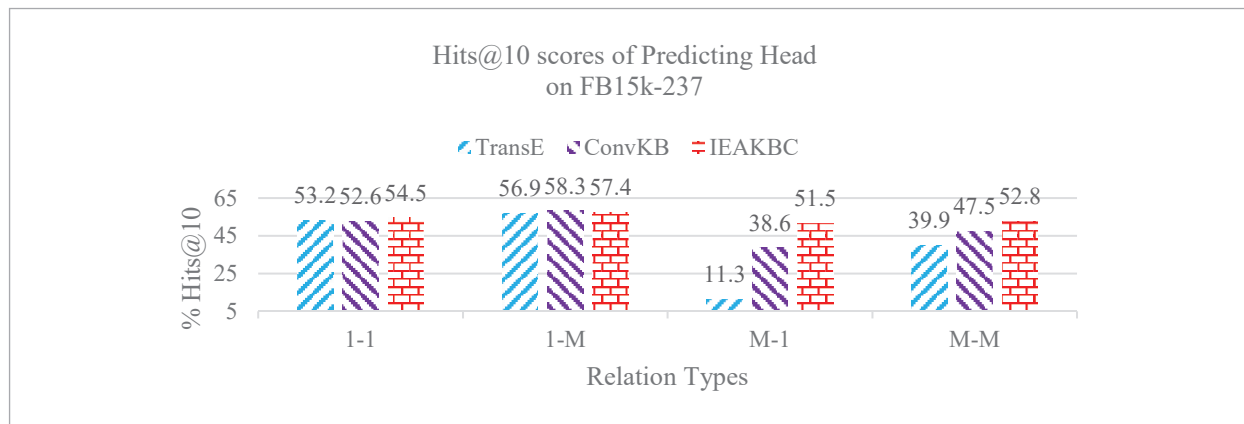
In further experiments afterwards, TransE and ConvKB are selected as baselines.

#### 4.1.2. Results for Predicting Head / Tail on FB15k-237

The core initiative of IEAKBC is to enhance the modeling capability for complex relations, so Hits@10 scores for predicting head and tail entities respectively for each relation category on FB15k-237 are computed and shown in Figures 2-3 so as to see whether

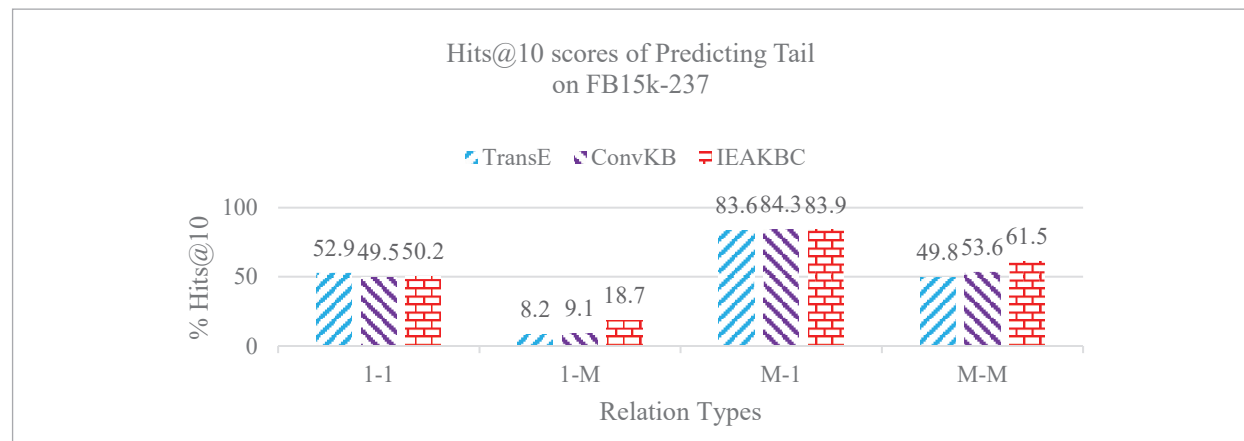
**Figure 2**

Hits@10 on FB15k-237 for Predicting Head



**Figure 3**

Hits@10 on FB15k-237 for Predicting Tail



the improvement of IEAKBC over TransE and ConvKB is related to its performance when handling complex relations. A common definition for relation classification [8, 42] is adopted.

We can see that on “side 1” prediction all models perform well, since in such task one or more entities point to just one entity, making the latter easy to identify. Gaps get wider on “side M” prediction, in which IEAKBC obtains the laurels on all the four tasks, i.e., M-1 and M-M for predicting head, 1-M and M-M for predicting tail. Since the main difference between IEAKBC and ConvKB lies in fusing relational attributes into entity embeddings, such results indicate that relational integration indeed helps with KBC tasks.

**4.1.3. Results with Regard to Each Relation on WN18RR**

To confirm our judgement with more evidence, link prediction task is pushed further; Hits@10 and MRR scores with regard to all 11 relations on WN18RR are counted and shown in Figures 4-5 to see where the strength of IEAKBC lies, in which the order of relations is organized according to their proportions of

all triplets in ascending order (see the broken line). IEAKBC achieves the highest Hits@10 scores on 7 out of 11 relations (including on par with TransE / ConvKB on 2 relations).

At MRR, it is obvious that IEAKBC maintains previous performance on challenging relations including *has\_part* (1-M), *member\_meronym* (1-M) and *hypernym* (M-1), while TransE and ConvKB suffer a huge degradation on these relations, of which the latter two constitute a substantial fraction of all test triplets, leading to the decrease of whole MRR scores for TransE and ConvKB. Such performance is in accord with results of predicting tail for 1-M relations and predicting head for M-1 relations on FB15k-237. On M-M complex relations such as *similar\_to*, *verb\_group*, *also\_see* and *derivationally\_related\_form*, since symmetric patterns could boost performance to a great extent, all the models achieve similar scores.

Results shown in Figures 4-5 are consistent: IEAKBC works well on prediction for complex relations, evidence that taking into consideration various properties of entities pairing with different relations makes sense.

**Figure 4**  
Hits@10 on WN18RR with Regard to Each Relation

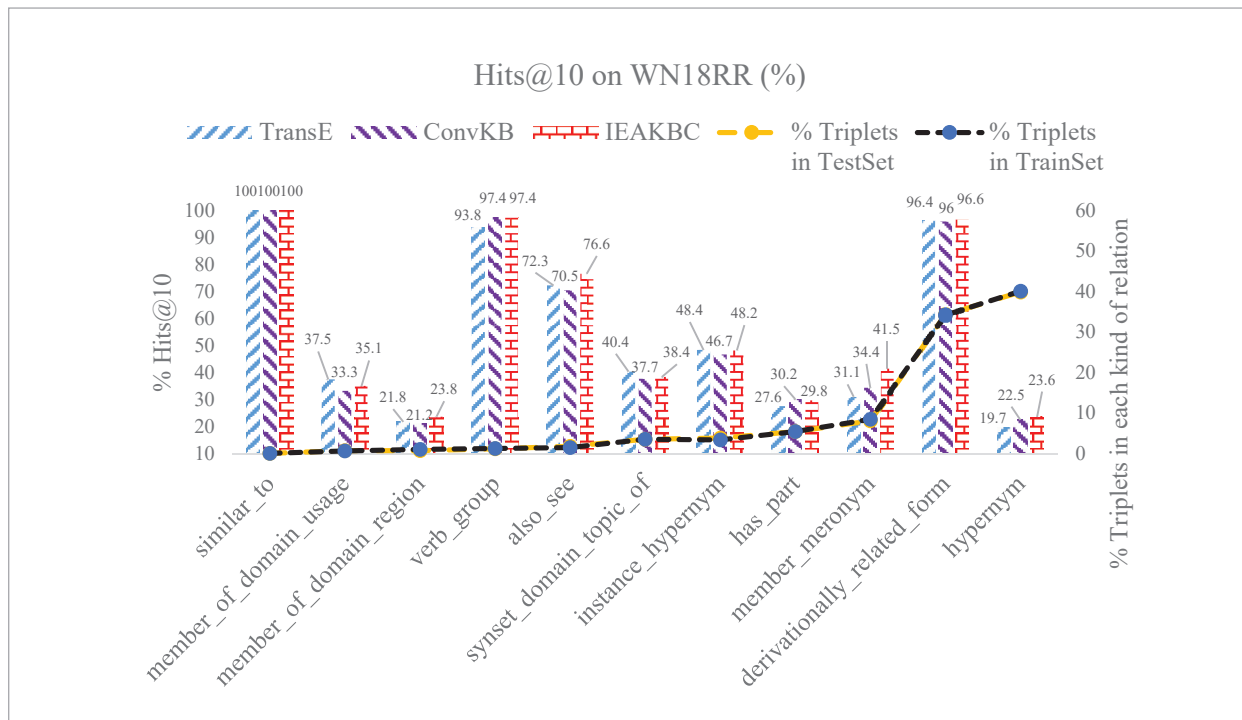
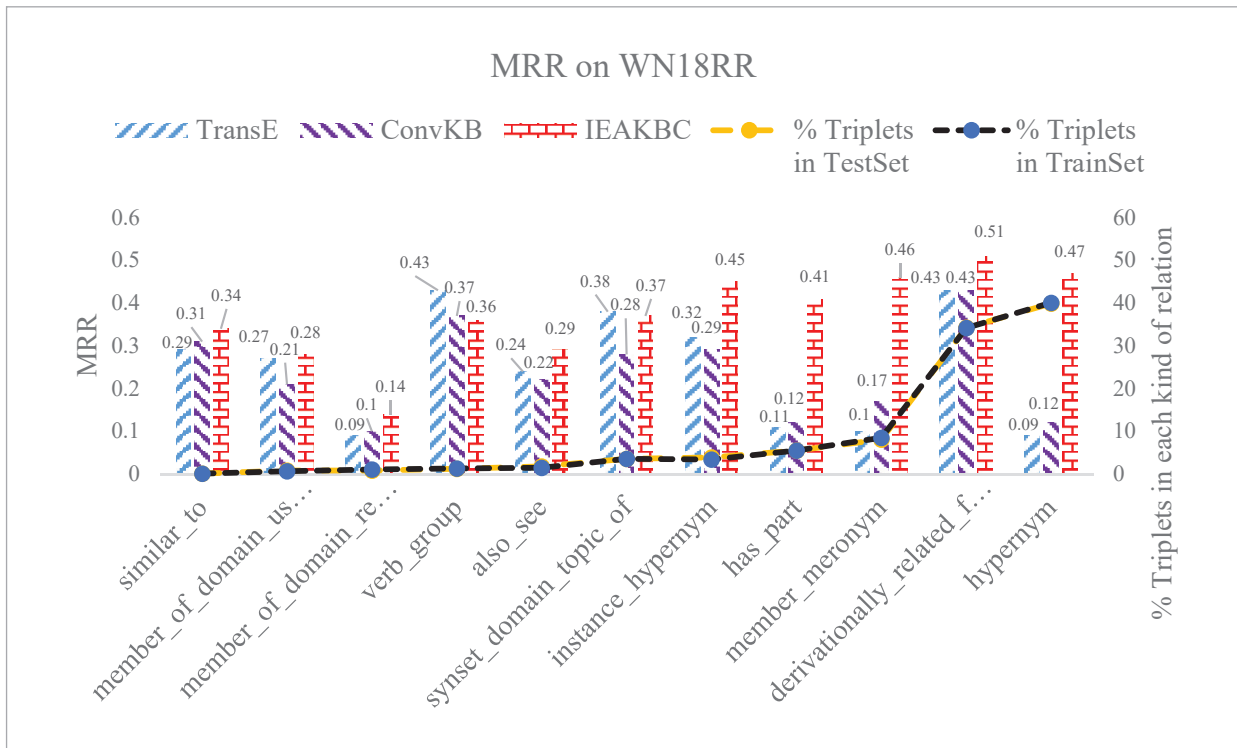


Figure 5

MRR on WN18RR with Regard to Each Relation



## 4.2. Triplet Classification

**Task:** The aim is to predict whether or not a given triplet is valid, i.e., this is a **binary classification task**. A threshold for each relation is introduced to help judge the validity of unseen triplets. The triplet is considered valid only if the triplet dissimilarity score is lower than  $q$ . According to Chen D et al. [12],  $q$  is achieved by maximizing the micro-averaged classification accuracy on the validation set.

**Models:** ConvE, DistMult, TransE and ConvKB are picked as representatives of their kind for comparison and we also refer to data from previous studies for comparison.

**Evaluation Protocol:** Classification accuracy is employed as the metric.

**Training Protocol:** Same as in the link prediction task, negative sampling with Bernoulli Trick (BT) is kept except for ConvE (1-N). Embedding representations initialized by TransE are employed as input to ConvKB and IEAKBC.

**Hyper-parameters Tuning:** We still use grid search to find optimal hyper-parameters, the parameter pool is similar to that in the link prediction task with supplements listed below. Optimal hyper-parameters are listed in Table 4. Due to limited computation resources, we did not combine word embedding approaches [61, 52-53].

Initializer in TensorFlow  $\in$  {normal, uniform, truncated\_normal, [0.1, 0.1, -0.1]}

Embedding Layer Dropout / Feature Map Layer Dropout / Projection Layer Dropout  $\in$  {0.0, 0.1, 0.2, 0.3, 0.4, 0.5} (for ConvE)

Label Smoothing  $\in$  {0.05, 0.1, 0.15, 0.2} (for ConvE)

Kernel Number  $\in$  {100, 200, 300, 400, 500} (for ConvE, ConvKB and IEAKBC)

Maximum Epoch=500 (for ConvE, ConvKB and IEAKBC)

Maximum Epoch=2000 (for simple models including DistMult and TransE)

Regularizer Coefficient=0.001 (for ConvKB and IEAKBC)

**Table 4**Hyper-parameters for Optimal Performance in Triplet Classification Task<sup>6</sup>

	WN11					FB13				
	TransE	ConvE	DistMult	ConvKB	IEAKBC	TransE	ConvE	DistMult	ConvKB	IEAKBC
Embedding Size	50	200	100	50	50	100	200	100	100	100
Negative Sampling	BT	1-N	BT	BT	BT	BT	1-N	BT	BT	BT
Loss Function	MR	CE	CE	MR	MR	MR	CE	CE	MR	MR
Optimizer	SGD	AdaGrad	SGD	Adam	Adam	SGD	AdaGrad	SGD	Adam	Adam
Batch Size	128	128	512	256	256	128	128	512	256	256
Learning Rate	0.001	0.001	0.001	0.0005	0.0005	0.001	0.001	0.001	0.00005	0.0005
Regularizer	L1	L2	L2	L2	L1	L2	L2	L2	L2	L1
Initializer	normal	xavier_normal	uniform	trun	trun	normal	xavier_normal	uniform	trun	trun
margin $\gamma$	7	-	-	-	-	1	-	-	-	-
kernel number	-	100	-	200	200	-	100	-	200	200

**Table 5**

Accuracy for Triplet Classification (%)

Models	WN11	FB13	Avg.
NTN[*]	70.6	87.2	78.9
ConvE	83.5	86.2	84.9
DistMult	84.7	<b>89.8</b>	86.8
TransE	86.2	87.1	86.7
TransD[*]	86.4	89.1	<u>87.8</u>
TranSparse-S[*]	86.4	88.2	87.3
TranSparse-US[*]	<u>86.8</u>	87.5	87.2
ConvKB[*]	<b>87.6</b>	88.8	<b>88.2</b>
ConvKB[our results]	86.3	88.8	87.6
IEAKBC	85.9	<u>89.4</u>	87.7

<sup>6</sup> For ConvE, other optimal hyper-parameters include embedding dropout 0.2, feature map dropout 0.2, projection layer dropout 0.4, label smoothing 0.1.

Results are shown in Table 5, where [\*] means that data are taken from related publications. Results of NTN, TranSparse-S and ConvKB are from Nguyen et al. [42], results of TransD are from Ji et al. [25], and results of TranSparse-US are from Chang et al. [11] (For TranSparse, 'S' and 'US' mean structured and unstructured patterns, respectively). Implementations of DistMult and ConvE are from Dettmers et al. [15]. The implementation of ConvKB is from Nguyen et al. [42].

On the denser dataset FB13, DistMult and IEAKBC beat other competitors, indicating higher generalization capability. IEAKBC achieves higher accuracy than TransE and ConvKB (2.3% and 0.6%, respectively), which is in accord with the results of the link prediction task on WN18RR.

On WN11, IEAKBC loses to several models with a narrow margin. Our hypothesis is while simultaneously considering variant relations and transitional characteristics brings benefits, such combination of multiple criteria may lead to confusion in specific scenarios. The performance degradation of ConvE may come from similar cause. Improvement is still under consideration.

On average, ConvKB (the original results) and TransD outperform other models. IEAKBC loses to TransD with a gap of 0.1%, yielding better performance than multiple classic models (In our experiments, IEAKBC performs marginally better than ConvKB).

We notice that ConvE with 2D convolution seems to lag behind on this task, while the average score of DistMult is relatively low. On the other hand, multiple successors to TransE perform well on this task, implying the significance of retaining transitional characteristics.

Results of TransE and ConvKB on FB13 are again used as baselines to discuss the impact of relational integration on model performance for different kinds of relations, as is shown in Figure 6. Proportions of all relations in the test set are shown as a broken line. All relations could be classified as M-1 type except *institution* and *profession* belong to M-M type. IEAKBC advances ConvKB on both M-M relations and only loses on 2 out of 5 M-1 relations, *religion* and *nationality*, outperforming TransE on all the seven relations, consistent with results in the link prediction task.

## 5. Application in Personalized Search

**Task:** In this part, we hope to apply IEAKBC to practical scenarios, so the search personalization problem is consulted, since personalized search has become in the spotlight of search engine business in recent years [10, 14, 39, 68, 79]. Related studies exploit user behavior data [73], building user profiles so as to tailor search results accordingly with the idea of relevant items coming first. In such applications, query, relevant user and document are usually put together and constitute triple-like structure (query, user, document) [66], representing lots of M-M relationships and implying the user's interest on the document under one specific query. To such structure knowledge representation approaches could be applied and evaluated. To be specific, the idea is to **re-rank** the returned documents with these approaches and expectations are more relevant documents should be ranked higher.

**Dataset and Pre-processing:** The dataset is a subset of a query log repository from a large commercial search engine, which originally includes 1166 ran-

domly chosen users and 489384 triplets, with the time span from July 1st to July 28th, 2012. The same source data has been employed by Nguyen, et al. [42] and Vu, et al. [67, 69]

One search entry includes a masked user ID, a query, top-10 returned URLs (documents) with ranking orders from the search engine, clicking records, dwell time and the relevant-or-not label to documents judged by SAT criteria (either a click with a dwell time of at least 30 seconds or the last result click in a search session is seen as a SAT click. Related document is labeled relevant and all the other documents in the same query are labeled as irrelevant). The rank order of relevant labeled documents is deemed as the ground truth to evaluate the search performance before and after re-ranking.

We try to keep the same setting as Vu et al. [66] as much as possible for comparable results, so we employ the same short-term user profile protocol [3], remove the queries with empty relevant label set, discard domain-related queries such as Twitter or Facebook and construct the validation set and test set with latest log entries while the training set with remaining entries. Statistical information of the dataset is shown in Table 6. All 7796 triplets are valid.

**Embedding Initialization:** The same process used by Vu et al. [66] is followed.

**Evaluation Protocol:** Triplet score is computed and organized in the ascending order to re-rank the top 10 documents returned by the search engine. **MRR**, **Hits@1** and **Hits@3** are adopted as performance metrics with higher values indicating better performance. (Hits@10 does not work here for we only use top 10 returned documents).

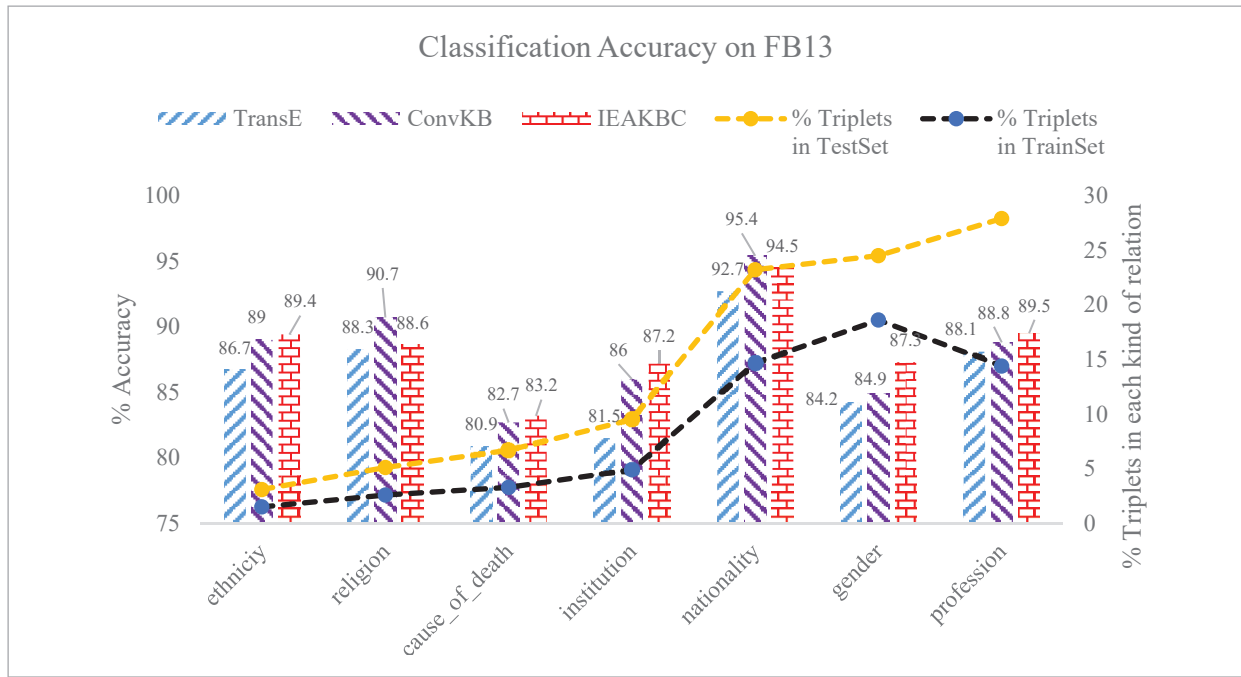
**Models and Hyper-parameters Tuning:** TransE and ConvKB are employed as baselines. The implementation of ConvKB is from Nguyen et al. [42] and we keep the recommended parameters **unchanged**. For IEAKBC, we follow the same **training protocol** and **hyper-parameters tuning** in the link prediction task. Optimal hyper-parameters are listed in Table 7.

In addition, **results from related literature [42] are introduced for comparison marked with [\*], including:** the original rank returned by the search engine (denoted as SE), CI [60] (a personalized method using clicking records), SP [69] (another personalized method employing session-based user profiles).



**Figure 6**

Classification Accuracy on FB13



**Table 6**

Dataset Statistics for Personalized Search

users	distinct queries	SAT clicks	distinct documents	Triples	Training Set	Validation Set	Test Set
106	5741	7796	31282	7796	5475	1141	1180

**Table 7**

Optimal Hyper-parameters for Personalized Search

	TransE	ConvKB	IEAKBC
Embedding Size	100	100	100
Negative Sampling	BT	BT	BT
Loss	MR	MR	CE
Optimizer	SGD	Adam	Adam
Batch Size	128	256	256
Learning Rate	0.005	0.0005	0.0001
Regularizer	L1	L2	L1
Initializer	normal	trun	trun
margin $\gamma$	5	-	-
kernel number	-	500	200

**Table 8**

Performance Comparison for Personalized Search

Model	MRR	Hits@1(%)	Hits@3(%)
SE[*]	0.559	38.5	=
CI[*]	0.597	41.6	=
SP[*]	0.631	45.2	
TransE	0.648	50.2	82.7
ConvKB	<u>0.733</u>	<b>58.6</b>	<u>85.5</u>
IEAKBC	<b>0.749</b>	<u>55.4</u>	<b>88.3</b>

Results are shown in Table 8. All embedding models achieve better performance than traditional approaches CI and SP, implying applications of such models to personalized search tasks. Both ConvKB and IEAKBC perform better than TransE, indicating

better feature extraction capabilities. Compared with a similar model ConvKB, IEAKBC loses at Hits@1, but outperforms ConvKB at MRR and Hits@3. Similar to the accuracy score on WN11, we attribute the Hits@1 score to the relatively complicated architecture of IEAKBC over ConvKB while comprehensive results show that in a wider range relational integration brings more benefits.

## 6. Conclusion and Future Work

Our aim is to remove the representation limitations haunting TransE by integrating relational attributes into entity representations, extracting richer semantics with the help of highly expressive and efficient CNN framework while retaining the transitional

characteristics. We also hope with 1D convolution the ConvE architecture could be simplified without performance degradation. Results of multiple tasks on benchmark datasets validate our model, verifying improvement especially on complex relations. Nevertheless, still space for improvement, and further work may include: combining the strength from bilinear models; drawing support from logic-rules-based reasoning; integrating information from relation path or temporal sequence; initializing embeddings with word vectors; testing model scalability and performance on larger, sparser real world datasets; adapting IEAKBC for other applications like sentiment recognition [29], etc.

## Acknowledgement

This paper is sponsored by Fujian Provincial Educational and Scientific Project(FJJKCG20-402)

## References

1. Baier, S., Ma, Y., Tresp, V. Improving Visual Relationship Detection Using Semantic Modeling of Scene Descriptions. In: d'Amato C. et al. (Eds.), *The Semantic Web - ISWC 2017, Lecture Notes in Computer Science*, vol 10587, Springer, Cham, 2017, 53-68. [https://doi.org/10.1007/978-3-319-68288-4\\_4](https://doi.org/10.1007/978-3-319-68288-4_4)
2. Balažević, I., Allen, C., Hospedales, T. M. Hypernetwork Knowledge Graph Embeddings. In: Tetko, I., Kůrková, V., Karpov, P., Theis, F. (Eds.), *Artificial Neural Networks and Machine Learning - ICANN 2019: Workshop and Special Sessions, Lecture Notes in Computer Science*, vol 11731, Springer, Cham, 2019, 553-565. [https://doi.org/10.1007/978-3-030-30493-5\\_52](https://doi.org/10.1007/978-3-030-30493-5_52)
3. Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F., Cui, X. Modeling the Impact of Short- and Long-term Behavior on Search Personalization. *Proceedings of the 35th International ACM. Sigir Conference on Research and Development in Information Retrieval, (SIGIR 2012), Portland, Oregon, USA, August 2012*, 185-194. <https://doi.org/10.1145/2348283.2348312>
4. Blei, D. M., Ng, A. Y., Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003, 3(3), 993-1022.
5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, (SIGMOD 2008), Vancouver, Canada, June 2008*, 1247-1250. <https://doi.org/10.1145/1376616.1376746>
6. Bordes, A., Glorot, X., Weston, J., Bengio, Y. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 2014, 94(2), 233-259. <https://doi.org/10.1007/s10994-013-5363-6>
7. Bordes, A., Usunier, N., Chopra, S., Weston, J. Large-scale Simple Question Answering with Memory Networks. *arXiv preprint arXiv:1506.02075*, 2015.
8. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, (NIPS 2013), Lake Tahoe, Nevada, December 2013*, 2787-2795. <https://doi.org/10.5555/2999792.2999923>.
9. Bordes, A., Weston, J., Collobert, R., Bengio, Y. Learning Structured Embeddings of Knowledge Bases. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, (AAAI 2011), San Francisco, California, August 2011*, 301-306. <https://doi.org/10.5555/2900423.2900470>.
10. Cai, F., Wang, S., de Rijke, M. Behavior-based Personalization in Web Search. *Journal of the Association for Information Science and Technology*, 2017, 68(4), 855-868. <https://doi.org/10.1002/asi.23735>

11. Chang, L., Zhu, M., Gu, T., Bin, C., Qian, J., Zhang, J. Knowledge Graph Embedding by Dynamic Translation. *IEEE Access*, 2017, 5, 20898-20907. <https://doi.org/10.1109/ACCESS.2017.2759139>
12. Chen, D., Socher, R., Manning, C. D., Ng, A. Y. Learning New Facts from Knowledge Bases with Neural Tensor Networks and Semantic Word Vectors. *arXiv preprint arXiv:1301.3618*, 2013.
13. Chen, X., Jia, S., Xiang, Y. A Review: Knowledge Reasoning over Knowledge Graph. *Expert Systems with Applications*, 2020, 141(3), 429-449. <https://doi.org/10.1016/j.eswa.2019.112948>
14. Cheng, Z., Jialie, S., Hoi, S. C. H. On Effective Personalized Music Retrieval by Exploring Online User Behaviors. *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (SIGIR 2016), Pisa, Italy, July 2016, 125-134. <https://doi.org/10.1145/2911451.2911491>
15. Dettmers, T., Minervini, P., Stenatorp P., Riedel, S. Convolutional 2D Knowledge Graph Embeddings. *arXiv preprint arXiv:1707.01476*, 2017.
16. Duchi, J. C., Hazan, E., Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 2011, 12(7), 2121-2159. <https://doi.org/10.1109/TNN.2011.2146788>
17. Ebisu, T., Ichise, R., Torus, E. Knowledge Graph Embedding on a Lie Group. *arXiv preprint arXiv:1711.05435*, 2018.
18. Fan, M., Zhou, Q., Chan, E., Zheng, T. F. Transition-based Knowledge Graph Embedding with Relational Mapping Properties. *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, (PACLIC 2014), Phuket, Thailand, December 2014, 328-337.
19. Fellbaum, C., Miller, G. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. <https://doi.org/10.7551/mitpress/7287.001.0001>
20. Feng, J., Huang, M., Wang, M., Zhou, M., Hao, Y., Zhu, X. Knowledge Graph Embedding by Flexible Translation. *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning*, (KR 2016), Cape Town, South Africa, April 2016, 557-560. <https://doi.org/10.5555/3032027.3032102>
21. Garciaduran, A., Bordes, A., Usunier, N. Effective Blending of Two and Three-way Interactions for Modeling Multi-relational Data. In: *Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (Eds.), Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2014, Lecture Notes in Computer Science*, vol 8724, Springer, Berlin, Heidelberg, 2014, 434-449. [https://doi.org/10.1007/978-3-662-44848-9\\_28](https://doi.org/10.1007/978-3-662-44848-9_28)
22. Han, X., Huang, M., Yu, H., Zhu, X. TransG: A Generative Mixture Model for Knowledge Graph Embedding. *Computer Science*, 2015, 8(12), 2316-2325. <https://doi.org/10.18653/v1/P16-1219>
23. Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H., Zhao, J. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, (ACL 2017), Vancouver, Canada, July 2017, 221-231. <https://doi.org/10.18653/v1/P17-1021>
24. He, S., Liu, K., Ji, G., Zhao, J. Learning to Represent Knowledge Graphs with Gaussian Embedding. *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, (CIKM 2015), Melbourne, Australia, October 2015, 623-632. <https://doi.org/10.1145/2806416.2806502>
25. Ji, G., He, S., Xu, L., Liu, K., Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, (ACL | IJCNLP 2015), Beijing, China, July 2015, 687-696. <https://doi.org/10.3115/v1/P15-1067>
26. Ji, G., Liu, K., He, S., Zhao, J. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, (AAAI 2016), Phoenix, Arizona, February 2016, 985-991. <https://doi.org/10.5555/3015812.3015959>
27. Jiang, T., Liu, T., Ge, T., Sha, L., Li, S., Chang, B., Sui, Z. Encoding Temporal Information for Time-Aware Link Prediction. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (EMNLP 2016), Austin, Texas, November 2016, 2350-2354. <https://doi.org/10.18653/v1/D16-1260> <https://doi.org/10.18653/v1/D16-1260>
28. Kazemi, S. M., Poole, D. Simple Embedding for Link Prediction in Knowledge Graphs. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, (NIPS 2018), Montreal, Canada, December 2018, 4289-4300. <https://doi.org/10.5555/3327144.3327341>

29. Kilimci, Z. H., Omurca, S. I. Extended Feature Spaces Based Classifier Ensembles for Sentiment Analysis of Short Texts. *Information Technology and Control*, 2018, 47(3), 457-470. <https://doi.org/10.5755/joitc.47.3.20935>
30. Kim, Y. Convolutional Neural Networks for Sentence Classification. arXiv:1408.5882, 2014. <https://doi.org/10.3115/v1/D14-1181>
31. Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980, 2014.
32. Lacroix, T., Usunier, N., Obozinski, G. Canonical Tensor Decomposition for Knowledge Base Completion. arXiv preprint arXiv:1806.07297, 2018.
33. Lao, N., Cohen, W. W. Relational Retrieval Using a Combination of Path-Constrained Random Walks. *Machine Learning*, 2010, 81(1), 53-67. <https://doi.org/10.1007/s10994-010-5205-8>
34. Lao, N., Mitchell, T., Cohen, W. W. Random Walk Inference and Learning in a Large Scale Knowledge Base. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP 2011)*, Edinburgh, United Kingdom, July 2011, 529-539. <https://doi.org/10.5555/2145432.2145494>.
35. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Mendes, P. N., Hellman, S., Morsey, M., Kleef, P. V., Auer, S., Bizer, C. DBpedia-A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 2015, 6(2), 167-195. <https://doi.org/10.3233/SW-140134>
36. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S. Modeling Relation Paths for Representation Learning of Knowledge Bases. arXiv:1506.00379, 2015. <https://doi.org/10.18653/v1/D15-1082>
37. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, Z. Learning Entity and Relation Embeddings for Knowledge Graph Completion. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI 2015)*, Austin, Texas, January 2015, 2181-2187. <https://doi.org/10.5555/2886521.2886624>.
38. Liu, H., Wu, Y., Yang, Y. Analogical Inference for Multi-Relational Embeddings. *Proceedings of the 34th International Conference on Machine Learning, (ICML 2017)*, Sydney, NSW, Australia, August 2017, 2168-2178. <https://doi.org/10.5555/3305890.3305905>.
39. Liu, X. Modeling Users' Dynamic Preference for Personalized Recommendation. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, (IJCAI 2015)*, Palo Alto, California USA, July 2015, 1785-1791.
40. Luo, Y., Wang, Q., Wang, B., Guo, L. Context-Dependent Knowledge Graph Embedding. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2015)*, Lisbon, Portugal, September 2015, 1656-1661. <https://doi.org/10.18653/v1/D15-1191>
41. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohhammad, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saproov, A., Greaves, M., Welling, J. Never-ending Learning. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI 2015)*, Austin, Texas, January 2015, 2302-2310. <https://doi.org/10.5555/2886521.2886641>.
42. Nguyen, D. Q., Nguyen, D. Q., Nguyen, T. D., Phung, D. A Convolutional Neural Network-Based Model for Knowledge Base Completion and Its Application to Search Personalization. *Semantic Web*, 2019, 10(5), 947-960. <https://doi.org/10.3233/SW-180318>
43. Nguyen, D. Q., Sirts, K., Qu, L., Johnson, M. Stranse: A Novel Embedding Model of Entities and Relationships in Knowledge Bases. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL 2016)*, San Diego, California, June 2016, 460-466. <https://doi.org/10.18653/v1/N16-1054>
44. Nguyen, D. Q., Vu, T., Nguyen, T. D., Nguyen, D. Q., Phung, D. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, (NAACL 2019)*, Minneapolis, Minnesota, June 2019, 2180-2189. <https://doi.org/10.18653/v1/N19-1226>
45. Nguyen, D. Q. An Overview of Embedding Models of Entities and Relationships for Knowledge Base Completion. arXiv preprint arXiv:1703.08098, 2017.
46. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 2016, 104(1), 11-33. <https://doi.org/10.1109/JPROC.2015.2483592>. <https://doi.org/10.1109/JPROC.2015.2483592>
47. Nickel, M., Rosasco, L., Poggio, T. Holographic Embeddings of Knowledge Graphs. *Artificial Intelligence*, 2016, 2(19), 1955-1961. <https://doi.org/10.5555/3016100.3016172>.
48. Nickel, M., Tresp, V., Kriegel, H. A Three-Way Model for Collective Learning on Multi-Relational Data. *Proceedings of the 28th International Conference on In-*



- ternational Conference on Machine Learning, (ICML 2011), Bellevue, Washington, USA, June 2011, 809-816. <https://doi.org/10.5555/3104482.3104584>.
49. Nickel, M., Tresp, V., Kriegel, H., et al. Factorizing YAGO: Scalable Machine Learning for Linked Data. Proceedings of the 21st International Conference on World Wide Web, (WWW 2012), Lyon, France, April 2012, 271-280. <https://doi.org/10.1145/2187836.2187874>
  50. Nickel, M., Tresp, V. Logistic Tensor Factorization for Multi-Relational Data. arXiv preprint arXiv:1306.2084, 2013.
  51. Paulheim, H. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. Semantic web, 2017, 8(3), 489-508. <https://doi.org/10.3233/SW-160218>
  52. Pennington, J., Socher, R., Manning, C. D. Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2014), Doha, Qatar, October 2014, 1532-1543. <https://doi.org/10.3115/v1/D14-1162>
  53. Pinter, Y., Eisenstein, J. Predicting Semantic Relations using Global Graph Properties. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2018), Brussels, Belgium, October 2018, 1741-1751. <https://doi.org/10.18653/v1/D18-1201>
  54. Rossi, A., Firmani, D., Matinata, A., Merialdo, P., Barbosa, D. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. arXiv preprint arXiv:2002.00819, 2020.
  55. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI 2019), 33(01), 3060-3067. <https://doi.org/10.1609/aaai.v33i01.33013060>
  56. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. Proceedings of the 23rd International Conference on World Wide Web, (WWW 2014), Seoul, Korea, April 2014, 373-374. <https://doi.org/10.1145/2567948.2577348>
  57. Shi, B., Wenginger, T. ProjE: Embedding Projection for Knowledge Graph Completion. arXiv preprint arXiv:1611.05425, 2016.
  58. Sun, Z., Deng, Z., Nie, J., Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. arXiv preprint arXiv:1902.10197, 2019.
  59. Takahashi, R., Tian, R., Inui, K. Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, (ACL 2018), Melbourne, Australia, July 2018, 2148-2159. <https://doi.org/10.18653/v1/P18-1200>
  60. Teevan, J., Liebling, D. J., Geetha, G. R. Understanding and Predicting Personal Navigation. Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, (WSDM 2011), Hong Kong, China, February 2011, 85-94. <https://doi.org/10.1145/1935826.1935848>
  61. Tomas, M., Ilya, S., Kai, C., Corrado, G., Dean, J. Distributed Representations of Words and Phrases and their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems, (NIPS 2013), Lake Tahoe, Nevada, December 2013, 3111-3119. <https://doi.org/10.5555/2999792.2999959>.
  62. Toutanova, K., Chen, D. Observed Versus Latent Features for Knowledge Base and Text Inference. Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, (CVSC|WS 2015), Beijing, China, July 2015, 57-66. <https://doi.org/10.18653/v1/W15-4007>
  63. Trouillon, T., Dance, C. R., Gaussier, E., Bouchard, G. Knowledge Graph Completion via Complex Tensor Factorization. Journal of Machine Learning Research, 2017, 18(130), 4735-4772.
  64. Trouillon, T., Welbl, J., Riedel, S., et al. Proceedings of the 33rd International Conference on International Conference on Machine Learning, (ICML 2016), New York, NY, USA, June 2016, 2071-2080. <https://doi.org/10.5555/3045390.3045609>.
  65. Uktveris, T., Jusas, V. Application of Convolutional Neural Networks to Four-Class Motor Imagery Classification Problem. Information Technology and Control, 2017, 46(2), 260-273. <https://doi.org/10.5755/j01.itc.46.2.17528>
  66. Vu, T., Nguyen, D. Q., Johnson, M., Song, D., Willis, A. Search Personalization with Embeddings. In: Jose, J. et al. (Eds.), Advances in Information Retrieval- ECIR 2017, Lecture Notes in Computer Science, vol 10193, Springer, Cham, 598-604. [https://doi.org/10.1007/978-3-319-56608-5\\_54](https://doi.org/10.1007/978-3-319-56608-5_54)
  67. Vu, T., Song, D., Willis, A., Tran, S. N., Li, J. Improving Search Personalisation with Dynamic Group Formation. Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, (SIGIR 2014), Gold Coast,



- Queensland, Australia, July 2014: 951-954. <https://doi.org/10.1145/2600428.2609482>
68. Vu, T., Willis, A., Kruschwitz, U., Song, D. Personalised Query Suggestion for Intranet Search with Temporal User Profiling. Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, (CHIIR 2017), Oslo, Norway, March 2017, 265-268. <https://doi.org/10.1145/3020165.3022129>
69. Vu, T., Willis, A., Tran, S. N., Song, D. Temporal Latent Topic User Profiles for Search Personalisation. In: Hanbury, A., Kazai, G., Rauber A., Fuhr, N. (Eds.), Advances in Information Retrieval, ECIR 2015, Lecture Notes in Computer Science, vol 9022, Springer, Cham, 605-616. [https://doi.org/10.1007/978-3-319-16354-3\\_67](https://doi.org/10.1007/978-3-319-16354-3_67)
70. Wang, Q., Mao, Z., Wang, B., Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12), 2724-2743. <https://doi.org/10.1109/TKDE.2017.2754499>
71. Wang, Z., Zhang, J., Feng, J., Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, (AAAI 2014), Quebec City, Quebec, Canada, July 2014, 1112-1119. <https://doi.org/10.5555/2893873.2894046>
72. West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., Lin, D. Knowledge Base Completion via Search-Based Question Answering. Proceedings of the 23rd International Conference on World Wide Web, (WWW 2014), Seoul, Korea, April 2014, 515-526. <https://doi.org/10.1145/2566486.2568032>
73. White, R. W., Chu, W., Hassan, A., He, X., Song, Y., Wang, H. Enhancing Personalized Search by Mining and Modeling Task Behavior. Proceedings of the 22nd International Conference on World Wide Web, (WWW 2013), Rio de Janeiro, Brazil, May 2013, 1411-1420. <https://doi.org/10.1145/2488388.2488511>
74. Xiao, H., Huang, M., Hao, Y., Zhu, X. TransA: An Adaptive Approach for Knowledge Graph Embedding. arXiv preprint arXiv: 1509.05490, 2015.
75. Xiao, H., Huang, M., Zhu, X.. From One Point to A Manifold: Knowledge Graph Embedding for Precise Link Prediction. arXiv preprint arXiv: 1512.04792, 2015.
76. Xie, Q., Ma, X., Dai, Z., Hovy, E. An Interpretable Knowledge Transfer Model for Knowledge Base Completion. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, (ACL 2017), Vancouver, Canada, July 2017, 950-962. <https://doi.org/10.18653/v1/P17-1088>
77. Xiong, C., Power, R., Callan, J. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. Proceedings of the 26th International Conference on World Wide Web, (WWW 2017), Perth, Australia, April 2017, 1271-1279. <https://doi.org/10.1145/3038912.3052558>
78. Yang, B., Yih, W., He, X., Gao, J., Deng, L. Embedding Entities and relations for Learning and Inference in Knowledge Bases. arXiv preprint arXiv:1412.6575, 2014.
79. Yang, L., Guo, Q., Song, Y., Meng, S., Shokouchi, M., McDonald, K., Croft, B. W. Modeling User Interests for Zero-Query Ranking. In: Ferro, N. et al. (Eds.), Advances in Information Retrieval, ECIR 2016, Lecture Notes in Computer Science, vol 9626, Springer, Cham, 171-184. [https://doi.org/10.1007/978-3-319-30671-1\\_13](https://doi.org/10.1007/978-3-319-30671-1_13)
80. Yoon, H., Song, H., Park, S., Park, S.-Y. A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL 2016), San Diego, California, June 2016, 907-916. <https://doi.org/10.18653/v1/N16-1105>
81. Zhang, W., Paudel, B., Zhang, W., Bernstein, A., Chen, H. Interaction Embeddings for Prediction and Explanation in Knowledge Graphs. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, (WSDM 2019), Melbourne VIC, Australia, January 2019, 96-104. <https://doi.org/10.1145/3289600.3291014>

## Appendix A

### Model Comparison<sup>7</sup>

Classification	Model	Score Function	Loss Function
Translation	TransE [8]	$\ v_h + v_r - v_t\ _{L1/L2}$ $v_h, v_r, v_t \in \mathbb{R}^k$ are vector representations from elements of triplet (h, r, t)	$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G}} \sum_{(h',r',t') \in \mathcal{G}'} [\gamma + d(v_h, v_r, v_t) - d(v'_h, v'_r, v'_t)]_+$ $\gamma > 0$ is a margin hyper-parameter $d(v_h, v_r, v_t) = \ v_h + v_r - v_t\ _{L1/L2}$ $[\cdot]_+$ denotes the positive part of ' $\cdot$ ' $\mathcal{G}$ is the set of valid triplets while $\mathcal{G}'$ refers to the invalid triplets
Bilinear	DistMult [78]	$v_h^T W_r v_t$ $W_r$ is a diagonal matrix $\in \mathbb{R}^{k \times k}$	$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G}} \sum_{(h',r',t') \in \mathcal{G}'} \max(0, 1 + s(h', r', t') - s(h, r, t))$ $s(h, r, t) = v_h^T W_r v_t$
Complex Space	Complex [64]	$\text{Re}(c_h^T C_r \hat{c}_t)$ $c_h, c_t \in \mathbb{C}^k$ are entity representations in complex space $C_r$ is a diagonal matrix $\in \mathbb{C}^{k \times k}$ $\hat{c}_t$ is the conjugate of $c_t$ $\text{Re}(c)$ denotes the real part of complex, $c \in \mathbb{C}$	$\mathcal{L} = \sum_{(h,r,t) \in \Omega} \log(1 + \exp(-Y_{rso} \cdot \text{Re}(c_h^T C_r \hat{c}_t))) + \lambda \ \theta\ _2^2$ $\{Y_{rso}\}_{(h,r,t) \in \Omega} \in \{-1, 1\}^{ \Omega }$ $\lambda$ is a L1/L2 regularizer $\theta$ corresponds to the embeddings of entities and relations here $\Omega$ denotes a set of observed triples including both true and false facts and $ \Omega $ denotes the number of all triplets
Neural Network	ConvE [15]	$f(\text{vec}(f(\text{concat}(\overline{v}_h, \overline{v}_r) * \Omega)) W) v_t^T$ $\overline{v}_h, \overline{v}_r$ denote 2D reshaping of $v_h, v_r$ , respectively concat is short for concatenation operation $f$ is a non-linear function $*$ refers to a convolution operation $\Omega$ denotes a set of kernels $\text{vec}(\cdot)$ is a reshaping function transforming tensors into vectors $W$ is a parameter matrix for linear transformation	$\mathcal{L} = -\frac{1}{N} \sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i))$ $p = \text{sigmoid}(f(\text{vec}(f(\text{concat}(\overline{v}_h, \overline{v}_r) * \Omega)) W) v_t^T)$ $t$ is the label vector, elements of which $\in [0, 1]$ $N$ refers to the number of all entities (in ConvE $v_h, v_r$ may be composed of different dimensions)
CNN + Translation	ConvKB [42]	$\text{concat}(\text{ReLU}([v_h, v_r, v_t] * \Omega)) \cdot w$ $w$ is the weight vector	$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G} \cup \mathcal{G}'} \log(1 + \exp(l_{(h,r,t)} \cdot f(h, r, t))) + \frac{\lambda}{2} \ w\ _2^2$ $l_{(h,r,t)} = \begin{cases} 1, (h, r, t) \in \mathcal{G} \\ -1, (h, r, t) \in \mathcal{G}' \end{cases}$ , $f(h, r, t)$ is the score function
	IEAKBC	$\text{concat}(g([v_h, v_r, v_t] * \Omega)) \cdot w$	$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G} \cup \mathcal{G}'} \log(1 + \exp(l_{(h,r,t)} \cdot f(h, r, t))) + \frac{\lambda}{2} \ \theta\ _2^2$ $\theta$ denotes the embeddings and parameters acquired by learning processes

<sup>7</sup> We try to unify the symbols with same meanings here.

## Appendix B

### Implementation of truncated\_normal Distribution in PyTorch

#### Model Comparison:

```
def truncated_normal_(self, tensor, mean=0, std=):
    size = tensor.shape
    tmp = tensor.new_empty(size+(4,)).normal()
    valid = (tmp < 2) & (tmp > -2)
    ind = valid.max(-1, keepdim=True)[1]
    tensor.data.copy_(tmp.gather(-1, ind).
    squeeze(-1))
    tensor.data.mul_(std).add_(mean)
    return tensor
```

## Appendix C

### Embedding Initialization in Personalized Search

The same process used by Vu et al. [66] is followed to initialize embeddings for documents, queries and user profiles. First, a LDA topic model [4] from MALLET toolkit<sup>8</sup> is trained with 200 topics only on the relevant documents. Then the model is employed to

<sup>8</sup> <http://mallet.cs.umass.edu/>

infer the probability distribution over topics for each document and the topic proportion vector of each document is taken as its embedding with the size  $k=200$ . In particular, the  $z$ th element ( $z = 1, 2, \dots, k$ ) of the vector embedding for document  $d$  is denoted as:  $v_{d,z} = P(z|d)$  and  $P(z|d)$  is the probability of the topic  $z$  given the document  $d$ .

Each query is also represented by a probability distribution vector over topics. Let  $\mathcal{D}_q = \{d_1, d_2, \dots, d_n\}$  be the set of top  $n$  ranked documents returned for a query  $q$  ( $n = 10$ ). The  $z$ th element of the vector embedding for query  $q$  is defined as  $v_{q,z} = \sum_{i=1}^n \lambda_i P(z | d_i)$ , where  $\lambda_i = \frac{\delta^{i-1}}{\sum_{j=1}^n \delta^{j-1}}$  is the exponential decay function of  $i$  which is the rank of  $d_i$  in  $\mathcal{D}_q$ , and  $\delta$  is the decay hyper-parameter ( $0 < \delta < 1$ ). In our experiments we found that when  $\delta$  is set to 0.7 model performance is optimal.

After the initialization of query and document embeddings, these representations are fixed during training for TransE, ConvKB and IEAKBC so as to avoid over-fitting. User embeddings are initialized with their clicking records under the temporal weighting scheme proposed by Vu et al. [69] with the idea that most recently clicked documents express more about the users' current interest [3].

For further details like hyper-parameters tuning please consult Vu et al. [66].

