QuatDE: Dynamic Quaternion Embedding for Knowledge Graph Completion

Haipeng Gao, Kun Yang, Yuxue Yang, Rufai Yusuf Zakari, Jim Wilson Owusu, Ke Qin*
University of Electronic Science and Technology of China
{gaohp, yangkun, yuxueyang}@std.uestc.edu.cn
{rufai6, owilsonjim}@gmail.com
{qinke}@uestc.edu.cn

Abstract

In recent years, knowledge graph completion methods have been extensively studied, in which graph embedding approaches learn low dimensional representations of entities and relations to predict missing facts. Those models usually view the relation vector as a translation (TransE) or rotation (rotatE and QuatE) between entity pairs, enjoying the advantage of simplicity and efficiency. However, QuatE has two main problems: 1) The model to capture the ability of representation and feature interaction between entities and relations are relatively weak because it only relies on the rigorous calculation of three embedding vectors; 2) Although the model can handle various relation patterns including symmetry, anti-symmetry, inversion and composition, but mapping properties of relations are not to be considered, such as one-to-many, many-to-one, and many-to-many. In this paper, we propose a novel model, QuatDE, with a dynamic mapping strategy to explicitly capture a variety of relational patterns, enhancing the feature interaction capability between elements of the triplet. Our model relies on three extra vectors donated as subject transfer vector, object transfer vector and relation transfer vector. The mapping strategy dynamically selects the transition vectors associated with each triplet, used to adjust the point position of the entity embedding vectors in the quaternion space via Hamilton product. Experiment results show QuatDE achieves state-of-the-art performance on three well-established knowledge graph completion benchmarks. In particular, the MR evaluation has relatively increased by 26% on WN18 and 15% on WN18RR, which proves the generalization of QuatDE.

1 Introduction

Billions of facts in the world can be stored in the Knowledge Graphs (KGs) with triples succinctly, and each triple (h, r, t) consist of two nodes and a directed edge between them. KGs such as Freebase(Bollacker et al. 2008), YAGO(Suchanek, Kasneci, and Weikum 2007) and DBpedia(Lehmann et al. 2015) are useful in many AI applications, such as question answering (QA)(Cui et al. 2019), recommended system(Wang et al. 2018), relation extraction(Wang et al. 2020), etc. Even though the KGs have been studied for many years, they still suffer from incompleteness, which makes their downstream assignments more challenging. As a result, researchers have put more focus on knowledge graph completion (KGC) task, which dedicates to predict missing links between nodes. In other words, KGC infers the implicit triples based on the true triplets that exist in the KGs. For example, if the triple (**Bill Clinton**, ***Friendship***, **Seven Spielberg**) is correct, i.e. **Bill Clinton** has a ***friendship*** with **Seven Spielberg**, we can infer that **Seven Spielberg** is a ***friend*** of **Bill Clinton** equally, i.e. (**Seven Spielberg**, ***Friendship***, **Bill Clinton**).

Conventional approaches for KGC have achieved substantial improvement via embedding entities and relations into low-dimensional continuous space, such as TransE(Bordes et al. 2013), TransH(Wang et al. 2014), TransD(Ji et al. 2015), TransR(Lin et al. 2015), etc. Instead of using a real-valued space, ComplEx(Trouillon et al. 2016), RotatE(Sun et al. 2019) project entities and relations into a complex space preforming the strong representation ability. Meanwhile, QuatE(Zhang et al. 2019), Rotate3D(Gao et al. 2020) show the rich feature interaction capacity between triple with Hamilton product in quaternion space, obtaining state-of-the-art (SOTA) link prediction results.

Now, although most models surrounding the KGC task are powerful enough, handling one-to-many, many-to-one, and many-to-many relation patterns are still a major challenge. For the knowledge subgraph about **Bill Clinton** and **Steven Spielberg** from FB15K-237 shown in Figure 1, ***spouse*** is the 1-to-1 relation, ***director*** and ***president*** are 1-to-n relations, ***religion***, ***ethnicity_people*** and ***employment*** are n-to-1 relations, ***friendship*** and ***award*** are n-to-n relations, respectively. TransE(Bordes et al. 2013) represents the relations as a translation from the objects to the subjects. For the triple (**Bill Clinton**, ***Spouse***, **Hillary Rodham Clinton**), TransE minimizes the distance between **Bill Clinton** + ***Spouse*** and **Hillary Rodham Clinton**. However, it's difficult for TransE to reply the query (**Steven Spielberg**, ***director***, **?**), whose answer is diverse, such as, **1941**, **Schindler's List** or others. The problem also appeared on RotatE(Sun et al. 2019) and QuatE(Zhang et al. 2019), both of them define each relation as a rotation from the source entity to the target entity. Intuitively, TransE, RotatE and QuatE only use the three embedding vectors to predict missing facts, which

*Corresponding author.

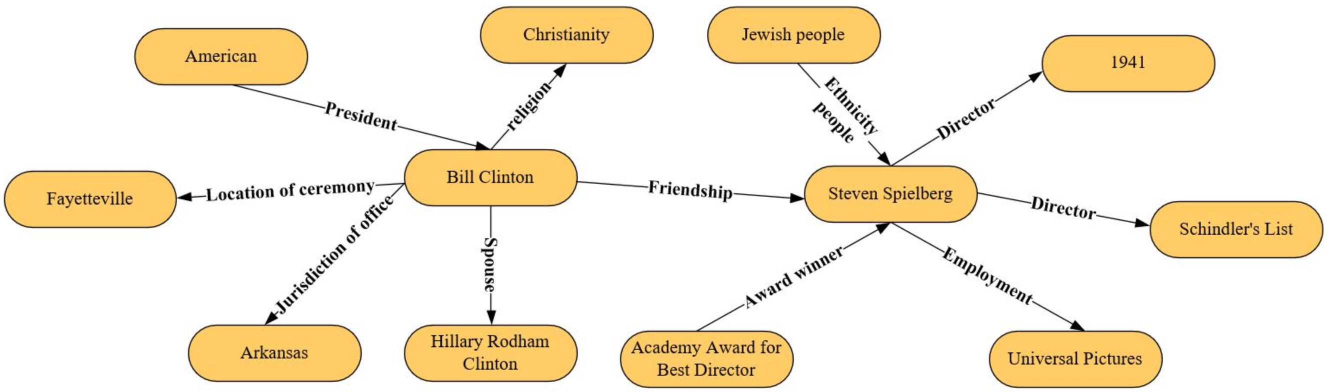

Figure 1: **subgraph of knowledge graph in FB15K-237. Entities are represented as golden blocks and directed links are represented as black arrows**

limits the expressive ability and fails to handle various relational modes.

QuatE, a quaternion embedding model with simple architecture, has shown strong adaptability on the standard knowledge graph completion datasets and achieved significant scores. Scoring function of QuatE is performed as $Q_h \otimes W_r^{\triangleleft} \cdot Q_t$, where Q_h is quaternion embedding vector of head, Q_t is quaternion embedding vector of tail, respectively, and $W_r^{\triangleleft}$ is the unit quaternion embedding vector of relation. $\otimes$ is the Hamilton product, modelled as the rotation from the subject to the object. Next, the inner product is used to measure the similarity between the rotated vector and Q_t. In fact, (**Steven Spielberg**, ***director***, **1941**) and (**Steven Spielberg**, ***director***, **Schindler's List**) are golden triples in FB15K-237, therefore the scores of both will have a better result. We assume that the score of former is $S_1 = Q_{\text{Steven Spielberg}} \otimes W_{\text{director}}^{\triangleleft} \cdot Q_{1941}$, the score of latter is $S_2 = Q_{\text{Steven Spielberg}} \otimes W_{\text{director}}^{\triangleleft} \cdot Q_{\text{Schindler's List}}$. So, $S_1 \approx S_2$ should be hold, i.e. $Q_{\text{Steven Spielberg}} \otimes W_{\text{director}}^{\triangleleft} \cdot Q_{1941} \approx Q_{\text{Steven Spielberg}} \otimes W_{\text{director}}^{\triangleleft} \cdot Q_{\text{Schindler's List}}$ that leads to $Q_{1941} \approx Q_{\text{Steven Spielberg}}$. It comes to an unreasonable result that the embeddings of **1941** and **Schindler's List** are closed which is harmful to the entire embedded learning of the KGs triplets. And if $Q_{1941} \approx Q_{\text{Steven Spielberg}}$, the answers of query (**?**, ***actor/performance***, **1941**) and (**?**, ***actor/performance***, **Schindler's List**) are the same according to the above score function, obviously, it's wrong. In short, QuatE ignores to dig out the deep-level feature of relations similar to ***director*** (1-to-n) and ***actor/performance*** (n-to-n).

In this paper, a novel quaternion embedding model QuatDE for knowledge graph completion task is proposed to alleviate the 1-to-n, n-to-1, and n-to-n issues. In our opinion, the same entities in different triples should be represented by distinct vector representations, closely related to relations and positions (head or tail). So, based on QuatE, we introduce three additional quaternion vectors (defined as subject transfer vector, object transfer vector and relation transfer vector), used to construct dynamic mapping strategy function $S_r(Q_e)$, where e is any entity. To be precise, after obtaining the head embedding vector Q_h (tail embedding vector Q_t) for a triple (h, r, t), interact with subject transfer vector P_h (object transfer vector P_t) and then relation transfer vector V_r with Hamilton product respectively, i.e. $S_r(Q_h) = Q_h \otimes P_h^{\triangleleft} \otimes V_r^{\triangleleft}$, $S_r(Q_t) = Q_t \otimes P_t^{\triangleleft} \otimes V_r^{\triangleleft}$. The generated two representation vectors $S_r(Q_h)$ and $S_r(Q_t)$ rich in cross information, filled into QuatE and predict the triplet score, i.e. $S_r(Q_h) \otimes W_r^{\triangleleft} \cdot S_r(Q_t)$. In this way, the embedding Q_{1941} of **1941** existing in (**Steven Spielberg**, ***director***, **1941**) is different from that of $Q_{Schindler's List}$ of **Schindler's List** existing in (**Steven Spielberg**, ***director***, **Schindler's List**), meanwhile, the model gains a higher degree of freedom and full expressive ability.

Our Contributions In summary, our proposed model has the following contributions:

- We propose a novel knowledge graph embedding method in the quaternion space, using a dynamic mapping method (strongly related to entities and relations) to explicitly enhance the interaction among triplets, modeling the diversity of relations and the correlations of the entities.
- Our model has multiple-level improvements in embedding dimensions and has strong generalization capabilities (MR). To be precise, MR and Hit@10 of our model are still better than the QuatE model when the embedding dimension is 50 in WN18 (one-sixth compared with QuatE).
- QuatDE is an excellent result of combining QuatE and quaternion-valued neural network and we expound on the superiority of QuatDE from the perspective of QNN.
- Our method is extended to standard benchmark datasets: FB15K-237, WN18 and WN18RR. Experiment results prove that our method is superior to the previous methods, and our code can be available on GitHub: https://github.com/hopkin-ghp/QuatDE.

2 State of the art

In this section, we will roughly describe the related work in two parts: translational distance models and semantic matching models. Note that semantic matching models exploit similarity-based scoring functions including bilinear models and neural network based models. Then we will discuss the connection between our approach and others.

translational distance models

The translational distance models utilize distance-based cost function and are efficient with low computational cost. We will describe this type of models with a unified formula, the scoring function is designed as follows:

$$f_r(h,t) = ||S_r(\boldsymbol{h}) + \boldsymbol{r} - S_r(\boldsymbol{t})||_{l1/l2} \tag{1}$$

in which function $f_r(h,t)$ represents the score of triple (h,r,t), and $S_r(\cdot)$ donates as the linear function, projecting entity embedding vectors in the relation-specific vector space. Generally, such models measure distance from source entities to target entities with $l1$-norm or $l2$-norm.

TransE(Abboud et al. 2020) is the most primitive but prominent model: $S_r(\boldsymbol{h}) = \boldsymbol{h}$, $S_r(\boldsymbol{t}) = \boldsymbol{t}$. However, TransE does not do well in dealing with 1-to-N, N-to-1, N-to-N relations. To solve this problem, the variants of above model, TransH, TransD, TransR, etc, consider unequal mapping strategy to project relation embedding in vector or matrix space, or capturing relational interactions(Ji et al. 2020). TransH(Wang et al. 2014) interprets a relation as a hyperplane with a translation operation, thus, original entity embedding vectors are projected into corresponding relation hyperplane. TransH donates the normal vector in the hyperplane as w_r: $S_r(\boldsymbol{e}) = \boldsymbol{e} - \boldsymbol{w_r}^\top \boldsymbol{e}\boldsymbol{w_r}$. TransR(Lin et al. 2015) proposes to use a relation-specific projection matrix M_r rather than hyperplane to project the entities embedding vectors into the space: $S_r(\boldsymbol{e}) = M_r\boldsymbol{e}$. It can be seen that large-scale parameters will be designed, so training the model is demanding which requires a lot of storage space. TransD(Ji et al. 2015) models the relational mapping matrix M_{re} in a more flexible way: $S_r(\boldsymbol{e}) = M_{re}\boldsymbol{e}$, where $M_{re} = \boldsymbol{w_r}\boldsymbol{w_e}^\top + \boldsymbol{I}$. TranSparse(Ji et al. 2016) leverages a numerical space to deal with the heterogeneity and imbalance issues of KGs. TransM(Fan et al. 2014) focus on the structure of the knowledge graph via pre-calculating the distinct weight for each training triplet according to its relational mapping property: $f_r(h,t) = -\theta_r\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{l_1/l_2}$. TransAP(Zhang, Sun, and Zhang 2020) notes that scoring function based on translation can't deal with the circle structure and hierarchical structure, so it introduces position-aware entity embeddings and attention mechanism to capture different semantic of triples.

semantic matching models

To be precise, translation models only obtain shallow linear characteristics through simple subtraction or multiplication operations. The scoring functions of semantic matching models reflect the confidence of the semantic information of the triples. RESCAL(Nickel, Tresp, and Kriegel 2011) represents each relation as a full rank matrix, optimizing a scoring function that computes a bilinear product between head and tail entity embedding vectors and relation matrix. Due to the large number of parameters, the model has overfitting problem. To alleviate the issue, DistMult(Yang et al. 2014) uses a diagonal matrix for each relation which reduces parameters to a certain extent. Subsequently, ComplEx(Trouillon et al. 2016) extends DistMult to the complex-valued space, and use a trick that head and tail entity embeddings of the same entity are complex conjugates. SimplE(Kazemi and Poole 2018) and TuckER(Balažević, Allen, and Hospedales 2019b) build on canonical polyadic (CP) and Tucker decomposition, respectively. TuckER shows that several linear models, RESCAL, DistMult, ComplEx, SimplE, are special cases of TuckER.

To explore deep information, ConvE(Dettmers et al. 2018) propose a simple multi-layer convolutional architecture for link prediction. ConvE splices and reshapes the subject and relation embedding vectors, then performs a 2D convolution operation, vector flattening, fully connected layer, finally, it matches with all candidate object embeddings. To obtain deeper features, the 3-column matrix of the triple embedding vector is used in ConvKB(Nguyen et al. 2017) and CapsE(Vu et al. 2019), ConvKB changes the form of input data of ConvE while CapsE is based on the capsule network. InteractE(Vashishth et al. 2020) is improved in the convolution step of ConvE, which captures entity and relation feature interactions through three ideas: Feature Permutation, Checkered Reshaping, and Circular Convolution. HypER(Balažević, Allen, and Hospedales 2019a) propose a hypernetwork architecture that generates simplified relation-specific convolutional filters.

Recently, the approaches in geometric rotation with complex-valued and quaternion-valued embeddings which link prediction have proposed. RotatE(Sun et al. 2019) introduce relation-based rotation from subject entity to object entity in complex space, which can leverage the advantages of ComplEx(Trouillon et al. 2016) and DistMult(Yang et al. 2014) and infer multiple relation models. QuatE(Zhang et al. 2019) extends this idea to the quaternion space, and prove that by making rotations on two planes rather than on a single plane (RotatE), which has a high degree of freedom. Currently, Rotate3D(Gao et al. 2020) models the non-commutative composition pattern in three-dimensional space with quaternion representation. BoxE(Abboud et al. 2020) embeds entities as points, and relations as a set of hyper-rectangles (or boxes), which spatially characterize basic logical properties. HittER(Chen et al. 2020), a Hierarchical Transformer model consists of two Transformer blocks, joined to learn entity-relation composition and relational contextualization based on information of entity's neighborhood.

3 Architecture design

KGs are usually expressed as form: ($\boldsymbol{E}$, $\boldsymbol{R}$, $\boldsymbol{O}$), in which $\boldsymbol{E}$ is the set of entities, $\boldsymbol{R}$ is the set of relations and $\boldsymbol{O}$ is the set of fact represented as triplets (h,r,t). Link prediction task aims to utilize observed triples to predict hidden triples. We use lowercase letters h, r, t to denote subject entities, relations, and object entities, and the corresponding bold letters $\boldsymbol{h}$, $\boldsymbol{r}$, $\boldsymbol{t}$ denote column embedding vectors. Real-valued space and quaternion space are defined as $\mathbb{R}$ and $\mathbb{H}$, respectively.

Preliminaries

Quaternion algebra(Hamilton 1844) is an expansion of the complex algebra, belongs to the hypercomplex number system. Usually, a quaternion $q = a1 + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathbb{H}$ is composed of one real part and three imaginary parts, where $a, b, c, d \in \mathbb{R}$, and 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are the quaternion unit basis and $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Some basic definitions of quaternion are defined as follows (declaring two quaternions: $q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and $q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$):

- Quaternion ordered pairs:

$$\mathrm{q} = [a, \mathbf{v}] = [a, b\mathbf{i} + c\mathbf{j} + d\mathbf{k}] \quad \mathbf{v} \in \mathbb{R}^3 \tag{2}$$

  i.e. $q_1 = [a_1, \mathbf{v}_1]$, $q_2 = [a_2, \mathbf{v}_2]$. In this representation, we see the similarity between quaternion and complex number.

- Product of $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$:

$$\begin{array}{ccc} \mathbf{ij} = \mathbf{k} & \mathbf{jk} = \mathbf{i} & \mathbf{ki} = \mathbf{j} \\ \mathbf{ji} = -\mathbf{k} & \mathbf{kj} = -\mathbf{i} & \mathbf{ik} = -\mathbf{j} \end{array} \tag{3}$$

- Quaternion Addition and Subtraction:

$$q_1 \pm q_2 = [a_1 \pm a_2, \mathbf{v_1} \pm \mathbf{v_2}] \tag{4}$$

- Inner Product:

$$q_1 \cdot q_2 = [a_1 \cdot a_2, \mathbf{v_1} \cdot \mathbf{v_2}] = a_1 \cdot a_2 + b_1 \cdot b_2 + c_1 \cdot c_2 + d_1 \cdot d_2 \tag{5}$$

- Conjugate q^* of q:

$$q^* = [a, -\mathbf{v}] = [a, -b\mathbf{i} - c\mathbf{j} - d\mathbf{k}] \tag{6}$$

- Quaternion Normalization $q^{\triangleleft}$ of q:

$$q^{\triangleleft} = \frac{q}{\sqrt{a^2 + b^2 + c^2 + d^2}} \tag{7}$$

- Hamilton Product (Quaternion Multiplication):

$$\begin{aligned} q_1 \otimes q_2 = & (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\ & + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\,\mathbf{i} \\ & + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)\,\mathbf{j} \\ & + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)\,\mathbf{k} \end{aligned} \tag{8}$$

QuatDE

Specifically, we represent the entity embedding matrix $Q \in \mathbb{H}^{|\boldsymbol{E}| \times k}$ and the relation embedding matrix $W \in \mathbb{H}^{|\boldsymbol{R}| \times k}$ in the quaternion space, where $|\cdot|$ represents the number of elements in set, and k represents the embedding dimension. We use the following formula to calculate score of a triple (h, r, t) with our model:

$$f_r(h, t) = S_r(Q_h) \otimes W_r^{\triangleleft} \cdot S_r(Q_t) \tag{9}$$

in the formula, Q_h, Q_t and $W_r^{\triangleleft}$ donated as:

$$\begin{aligned} Q_h &= a_h + b_h\mathbf{i} + c_h\mathbf{j} + d_h\mathbf{k} \\ Q_t &= a_t + b_t\mathbf{i} + c_t\mathbf{j} + d_t\mathbf{k} \\ W_r^{\triangleleft} &= \frac{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}} \end{aligned} \tag{10}$$

Where Q_h, Q_t and unit quaternion $W_r^{\triangleleft} \in \mathbb{H}^k$. Correspondingly, the coefficient a_h, b_h, c_h, d_h, a_t, b_t, c_t, d_t, a_r, b_r, c_r and $d_r \in \mathbb{R}^k$. Here, $S_r(e)$ is a dynamic mapping function driven by entity ontology e and relation r. Symbol $\otimes$ defines the Hamilton product operation and symbol $\cdot$ defines the inner product operation, respectively.

For our model, dynamic mapping function lies on entity transition matrix $P \in \mathbb{H}^{|\boldsymbol{E}| \times k}$ and relation transition matrix $V \in \mathbb{H}^{|\boldsymbol{E}| \times k}$. We link h, t, r to vector $P_h^{\triangleleft}$, $P_t^{\triangleleft}$ and $V_r^{\triangleleft} \in \mathbb{H}^k$, and $S_r(Q_h)$ and $S_r(Q_t)$ are represented as follows:

$$\begin{aligned} S_r(Q_h) &= Q_h \otimes P_h^{\triangleleft} \otimes V_r^{\triangleleft} \\ S_r(Q_t) &= Q_t \otimes P_t^{\triangleleft} \otimes V_r^{\triangleleft} \end{aligned} \tag{11}$$

where $P_h^{\triangleleft} = a_{ph} + b_{ph}\mathbf{i} + c_{ph}\mathbf{j} + d_{ph}\mathbf{k}$, $P_t^{\triangleleft} = a_{pt} + b_{pt}\mathbf{i} + c_{pt}\mathbf{j} + d_{pt}\mathbf{k}$ are normalized entity transfer vectors, and $V_r^{\triangleleft} = a_{vr} + b_{vr}\mathbf{i} + c_{vr}\mathbf{j} + d_{vr}\mathbf{k}$ is normalized relation transfer vectors, in which a_{ph}, b_{ph}, c_{ph}, d_{ph}, a_{pt}, b_{pt}, c_{pt}, d_{pt}, a_{vr}, b_{vr}, c_{vr}, $d_{vr} \in \mathbb{R}^k$. The entity transfer vector P_e can adjust the spatial position of the same entity e when facing different triples, and the relation transfer vector V_r projects same entity to different relation-special representation spaces. The dynamic mapping function $S_r(e)$ combines the embedding vector of the entity, the entity transfer vector and the relation transfer vector via Hamilton product. In this way, QuatDE captures more detailed information, and can fit in all triples in the overall knowledge representation of the knowledge graph.

Formally, the score function of QuatDE can be represented as follows:

$$f_r(h, t) = [Q_h \otimes P_h^{\triangleleft} \otimes V_r^{\triangleleft}] \otimes W_r^{\triangleleft} \cdot [Q_t \otimes P_t^{\triangleleft} \otimes V_r^{\triangleleft}] \tag{12}$$

Loss Function : QuatDE were trained using Adagrad optimizer, by minimizing the negative log-likelihood of the logistic model with L^2 regularization on the parameters w of our model:

$$\begin{aligned} \mathcal{L} = \textstyle\sum_{(h,r,t) \in \{O \cup O^-\}} & \log\left(1 + \exp\left(-l_{(h,r,t)} f_r(h, t)\right)\right) \\ & + \lambda \|w\|^2 \\ \text{in which, } l_{(h,r,t)} = & \begin{cases} 1 & \text{for } (h, r, t) \in O \\ -1 & \text{for } (h, r, t) \in O^- \end{cases} \end{aligned} \tag{13}$$

in our model, the parameters w for L^2 norm include the embedding vectors and transfer vectors with rate λ. O is the set of golden triples, and O^- donates the set of negative triplets. Following opinion of Wang(Wang et al. 2014), adopting a Bernoulli distribution to generate negative triplets.

Discussion

Ability to handle complex relations We take relation ***director*** (1-to-N) as an example to describe our solution strategy, and choose the triples (**Steven Spielberg**, ***director***, **1941**) and (**Steven Spielberg**, ***director***, **Schindler's List**), whose labels are true. So according to the score function of QuatDE, $S_{\text{director}}(Q_{1941}) \approx S_{\text{director}}(Q_{\text{Schindler's List}})$, but the similarity between Q_{1941} and $Q_{\text{Schindler's List}}$ are still depend on P_{1941} and $P_{\text{Schindler's List}}$ which combine the feature extracted from the triples which include **1941** or **Schindler's List**. QuatDE makes a good trade-off between model efficiency and parameter complexity. Although introducing additional transfer vectors, QuatDE shows strong capabilities in some indicators by setting a smaller embedding dimension.

Connection to QuatE We applied our motivation to QuatE, and the result proves the feasibility of QuatDE. We believe that through the Hamilton product, the diversity problem of other quaternion models in the knowledge graph completion task can be solved. The transfer vector is removed, QuatDE degenerates to QuatE.

Connection to TransH, TransR, TransD Although our model belongs to semantic matching models, it also incorporates the advantages of translational distance models. TransH, TransR, and TransD alleviate the problem that TransE does not do well in dealing with 1-to-N, N-to-1, N-to-N relations at different perspectives. TransH and TransR dynamically model the connection structure characteristics between different triples facing different relations. However, TransD do not only consider the diversity of relationships, but also pays attention to entities, which is in line with our paper. The biggest motivation of QuatDE comes from the problem of entities and relations diversity in quaternion space, and we extend this idea to quaternion space via Hamilton product.

Connection to quaternion-valued neural network (QNN) Our work is also inspired by the widespread success of Quaternion number across countless fields, such as heterogeneous image processing(Parcollet, Morchid, and Linarès 2019), theme identification of telephone conversations(Parcollet, Morchid, and Linarès 2017), automatic speech recognition(Parcollet et al. 2018). As far as we know, we are the first to use the idea of QNN to connect and explain the knowledge graph embedding models. In this section, we will visualize the model architecture of QuatE (Figure 2) and QuatDE (Figure 3) from the perspective of QNN, which is not mentioned in QuatE.

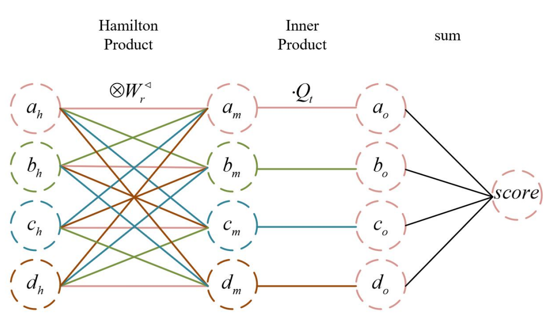

Figure 2: **architecture of QuatE**

As shown in Figure 2, the four components of a quaternion are represented by different colors, the weights are also represented with quaternions and transformation is performed with Hamilton product or inner product. In the QuatE model, the input is the quaternion embedding of head; the weights of the first layer are unit quaternion of relation; we can get the intermediate vector via Hamilton product; the intermediate vector is filled into the second layer, and carries out inner product with tail quaternion embedding; Finally, the resulting output vector can be viewed as real number, so we can calculate the triplet score by addition.

As shown in Figure 3, on the basis of QuatE, we add four quaternion feed-forward layers, two layers are used to construct the dynamic strategy function of the head, and the other two are used for that of the tail. The weights of four new layers are closely related to elements of the triplet, i.e. subject transfer vector $P_h^{\triangleleft}$, object transfer vector $P_t^{\triangleleft}$ and relation transfer vector $V_r^{\triangleleft}$, rather than random parameters like traditional neural networks. Meanwhile, increased layers enable more complicated interactions and is less likely to cause over-fitting.

4 Experiment and Analysis

Datasets

We evaluate the performance of our model on three general data sets: WN18, WN18RR and FB15K-237. Statistics about the data set are shown in the table 1. FB15K(Bordes et al. 2013) is a subset of Freebase, which is a large dataset contains the facts about sports, actors, movies and others. FB15K-237(Toutanova and Chen 2015) was extracted from FB15K and removed inverse relations, which prevent the leakage issue of test triples. WN18(Bordes et al. 2013) is the subset of Wordnet, and it is full of lexical relations between words. WN18 also has many inverse relations, hence, WN18RR(Dettmers et al. 2018) is removed inverse relations.

Dataset	**|E|**	**|R|**	**#training**	**#validation**	**#test**
FB15K-237	14541	237	272115	17535	20466
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134

Table 1: **statistics about the experimental datasets.**

Evaluation protocol

We evaluate related methods on two tasks: link prediction and triplet classification. Link prediction task aims to infer the answer of the query $(h, r, ?)$ or $(?, t, r)$ where $?$ means the missing element. So, for each test triple, we calculate the score of all possible triples which can be obtained by substituting subject and object, and sort all scores in descending order.

Mean Rank (MR) and Hit at n are standard evaluation measures for these datasets which are applied in our models. MR measures the average rank of each triplet to predict the correct answer. MRR is defined as the average value of the reciprocated rank, and Hit@n calculates the probability of including the correct entity in the top n ranks. Note that we use the filtered metrics following bordes(Bordes et al.

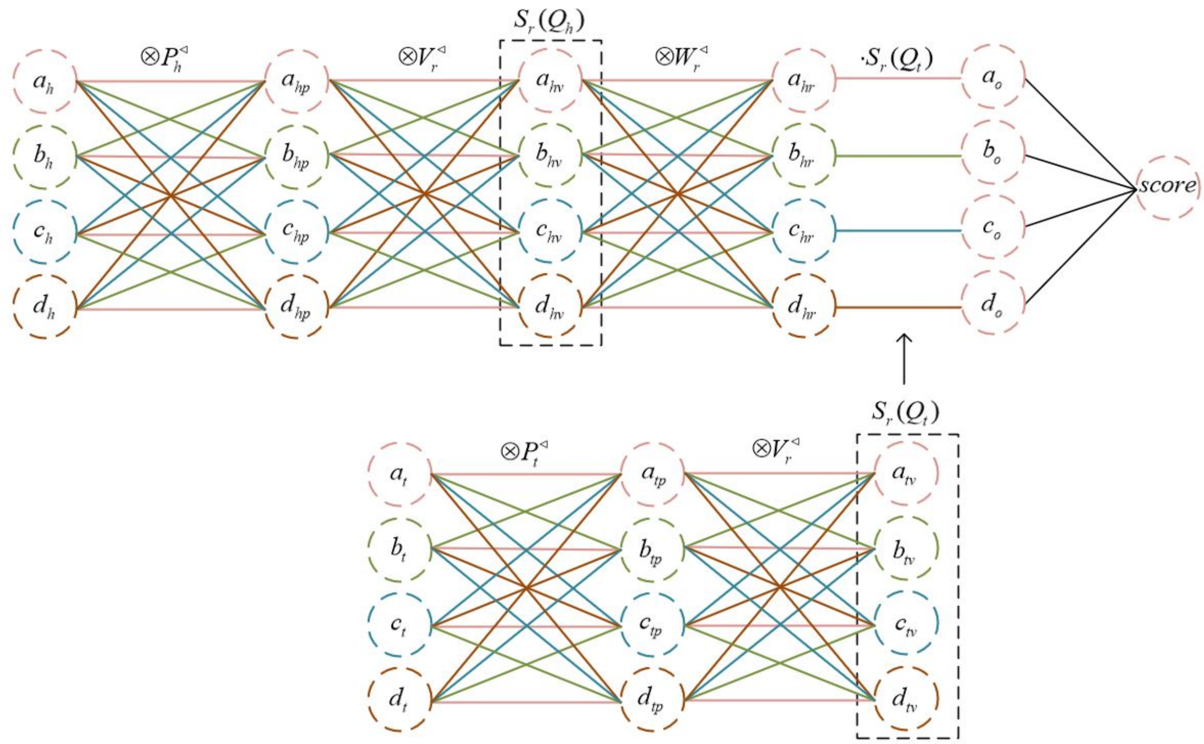

Figure 3: **architecture of QuatDE**

2013). The metrics remove all golden triples that appear in either training, validation or test set from the ranking.

Training details

Our code is based on the OpenKE framework and QuatE, and was implemented with PyTorch. We set 100 batches for all datasets. We train our model for 3000 epochs and valid the performance every 300 epochs on three datasets. The dimensionality of embeddings $k \in \{50, 100, 200, 300, 400\}$, the number of negative triples for each triple $\in \{1, 5, 10\}$, learning rate is selected in $\{0.05, 0.1\}$, and L^2 regularization parameters $\lambda \in \{0.05, 0.1\}$.

Experimental results

Link prediction Table 2 shows link prediction performance of various models: including translational distance models: TransE(Bordes et al. 2013), semantic matching models: ComplEx(Trouillon et al. 2016), ConvE(Dettmers et al. 2018), rotatE(Sun et al. 2019), QuatE(Zhang et al. 2019), and recent well-performing models: InteractE(Vashishth et al. 2020), ATTH(Chami et al. 2020), CompGCN(Vashishth et al. 2019), Rotate3D(Gao et al. 2020). Most of the experimental data are quoted from the original papers. Table 2 shows that QuatDE achieved competitive scores than others.

More deeply, we can observe that: 1) QuatDE obtains best scores for MR, MRR, Hit@10 and Hit@3 on WN18RR, and MRR, Hit@10, Hit@3 and Hit@1 on FB15K-237. 2) On WN18, Rotate3D, QuatE and QuatDE are modeled in quaternion space and achieve comparable scores, but our model performs best on MR than the other two. 3) QuatDE is fruitful on FB15K-237, and gains a 0.017 higher MRR, 1.3% higher Hit@10, 1.8% higher Hit@3 and 2.0% higher Hit@1 than baseline QuatE. 4) The progressive of MR is most obvious. Notably, QuatDE obtains significant improvement of $2314 - 1977 = 337$ in MR (which is about 15% relative improvement) on WN18RR, and $162 - 120 = 42$ (which is about 26% relative improvement) on WN18.

Triplets Classification This task aims to judge whether a given triple (h, r, t) is correct or not. Table 3 presents triple classification accuracy of different models on WN18RR, FB15K-237 and WN18. We reproduced the code of QuatE and recorded the result of the triple classification task in a table, while others((Bordes et al. 2013), (Wang et al. 2014), (Nickel, Rosasco, and Poggio 2016), (Dettmers et al. 2018), (Nguyen et al. 2017), (Jia, Cheng, and Su 2020)) are taken from PConvKB(Jia, Cheng, and Su 2020). Overall, our model QuatDE obtained the best results on three data sets. Especially on FB15K-237 where QuatDE gains considerable improvements of $83.0 - 81.8 = 1.2$ compared with QuatE, and $83.0 - 82.1 = 0.9$ improvement with PConvKB.

Multi-relation analysis We analyzed the experimental results of complex relations on FB15K-237. There are 224 relation types in the FB15K-237 test triplets, and QuatDE has achieved an equal or higher score than QuatE on 186 relations (which accounts for 84%) when taking Hit@10 as

Model	WN18RR					FB15K-237					WN18				
	MR	MRR	Hit@10	Hit@3	Hit@1	MR	MRR	Hit@10	Hit@3	Hit@1	MR	MRR	Hit@10	Hit@3	Hit@1
TransE	3384	0.226	50.1	-	-	357	0.294	46.5	-	-	-	0.496	94.3	88.8	11.3
ComplEx	5261	0.44	51.0	46.0	41.0	339	0.247	42.8	27.5	15.8	-	0.941	94.7	94.5	93.6
ConvE	4187	0.430	52.0	44.0	40.0	244	0.325	50.1	35.6	23.7	374	0.943	95.6	94.6	93.5
InteractE	5202	0.463	52.8	-	43.0	172	0.354	53.5	-	26.3	-	-	-	-	-
rotatE	3340	0.476	57.1	49.2	42.8	177	0.338	53.3	37.5	24.1	309	0.949	95.9	95.2	94.4
ATTH	-	0.486	57.3	49.9	**44.3**	-	0.348	54.0	38.4	25.2	-	-	-	-	-
CompGCN	3533	0.479	54.6	49.3	**44.3**	197	0.355	53.5	39.0	25.4	-	-	-	-	-
Rotate3D	3328	**0.489**	57.9	50.5	44.2	165	0.347	54.3	38.5	25.0	214	**0.951**	**96.1**	95.3	**94.5**
QuatE	2314	0.488	58.2	50.8	43.8	**87**	0.348	55.0	38.2	24.8	162	0.950	95.9	**95.4**	**94.5**
QuatDE	**1977**	**0.489**	**58.6**	**50.9**	43.8	90	**0.365**	**56.3**	**40.0**	**26.8**	**120**	0.950	**96.1**	**95.4**	94.4

Table 2: **Link prediction result on WN18RR, FB15K-237 and WN18. The best score is in bold, while the second best score is in underline.**

Model	WN18RR	FB15K-237	WN18
TransE	74.0	75.6	87.6
TransH	77.0	77.0	96.5
HoLE	71.4	70.3	88.1
ConvE	78.3	78.2	95.4
ConvKB	79.1	80.1	96.4
PConvKB	80.3	82.1	97.6
QuatE	86.7	81.8	97.9
QuatDE	**87.6**	**83.0**	**98.0**

Table 3: **Triplets classification result.**

Relation examples		QuatE/QuatDE	
		Hit@10	MR
1-to-1	/film/film/prequel	0.75/0.917	9.53/3.69
	/education/educational_institution/campuses	0.69/1.0	25.1/1.0
	/location/hud_county_place/place	0.81/0.91	38.6/11.8
	/education/educational_institution_campus /educational_institution	0.61/1.0	14.2/1.15
1-to-n	/sports/sports_league/teams./sports /sports_league_participation/team	0.81/0.91	10.6/5.8
	/education/field_of_study/students_majoring. /education/education/student	0.31/0.37	52.1/41.0
	/organization/organization/child./organization /organization_relationship/child	0.38/0.5	28.1/26.4
n-to-1	/film/film/release_date_s./film/film_regional_release_date /film_release_distribution_medium	0.72/0.78	9.3/9.0
	/location/location/time_zones	0.72/0.78	43.8/24.8
	/film/film/produced_by	0.42/0.54	96.5/90.3
	/people/person/nationality	0.55/0.59	130.7/118.3
n-to-n	/location/location/contains	0.47/0.52	155.4/117.2
	/organization/organization_member/member_of. /organization/organization	0.79/0.88	11.9/6.7
	/film/film/country	0.51/0.56	109.4/91.5
	/music/genre/parent_genre	0.47/0.58	27.7/20.6

Table 4: **Hit@10 and MR of QuatE and QuatDE on some one-to-one, one-to-many, many-to-one, many-to-many relations in FB15K-237.**

a measure, proving the ability of QuatDE to mitigate the multi-relations. As shown in Table 4, we extract a few examples in each relational pattern (1-to-1, 1-to-N, N-to-1, N-to-N). It's obviously observed that QuatDE can also increase the prediction accuracy of the 1-to-1 relations, for example, the prediction accuracy of relation ***campuses*** and ***educational_institution*** is 100% in Hit@10, the reason is attributed to intimate feature interaction between the elements of triplet via Hamilton product. And we confirm that QuatDE obtains better MR and Hit@10 than QuatE.

Dimension analysis We make dimensional analysis on WN18 and FB15K-237, and compared each result when different embedding dimension size K was selected, which is shown in Figure 4 and Figure 5.

Figure 4 shows the impact of dimension on the QuatDE performance with the changing of dimension size K. We can observe that when the dimension is set to 10, Hit@10 has achieved notable scores higher than 0.94 and MRR is close to 0.90. It proves that QuatDE can generate fewer parameters to speed up the model, which allows QuatDE to be extended to a large knowledge graph. Hit@10 is used as the criterion for selecting the best model, and we fix the embedding dimension as 50, the results of the QuatDE experiment are better than the optimal score (0.959) of QuatE whose dimension is selected to 300 (six times than QuatDE), and QuatDE achieves the optimal model when the dimension is 100. When the dimension size is selected large than 100, the curve starts to fluctuate up and down gradually due to the introduction of excessive parameters.

Figure 5 depicts the dimensional-related experimental results of FB15K-237. On the whole, the resulting curve of FB15K-237 is smoother and more stable than that of WN18, which once again verifies the negative influence of the reverse relations on the experimental results, in detail, WN18 contains a large proportion of reverse relationships, while FB15K-237 excludes that relations. In addition, Hit@10 and MRR are both improved and slowly saturated around $k = 300$.

5 Conclusion

In this paper, we propose a novel embedding model QuatDE for link prediction task and triplet classification task in quaternion space. QuatDE utilizes a dynamic mapping method to enhance the character interaction with Hamiltonian product between the triple explicitly, and the model has a higher degree of freedom when training and fitting. The experimental results show that QuatDE outperforms other state-of-the-art models on three benchmark datasets WN18, WN18RR, and FB15K-237. Recently, QDN(quaternion deep network) has been proposed, but the network design of architecture isn't mature and lacking the

Relation Name	QuatE	QuatDE
hypernym	0.173	**0.177**
derivationally_related_form	**0.953**	0.946
instance_hypernym	**0.364**	0.363
also_see	0.629	**0.640**
member_meronym	0.232	**0.248**
synset_domain_topic_of	0.468	**0.493**
has_part	0.233	**0.239**
member_of_domain_usage	**0.441**	0.406
member_of_domain_region	0.193	**0.294**
verb_group	**0.924**	0.868
similar_to	**1.000**	**1.000**

Table 5: **MRR of QuatE vs QuatDE**

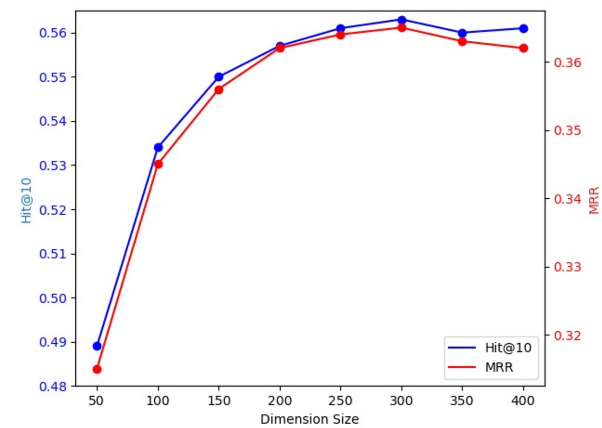

Figure 5: **MRR and Hit@10 with different dimensions of FB15K-237. The red line represents the MRR, and blue line represents the Hit@10**

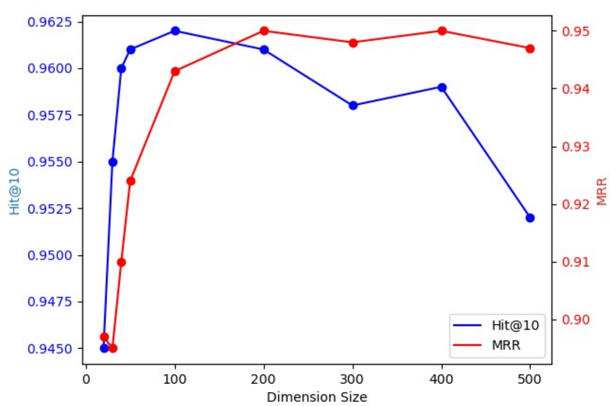

Figure 4: **MRR and Hit@10 with different dimensions of WN18. The red line represents the MRR, and blue line represents the Hit@10**

interpretability, future work will focus on details and theory of QDN, furtherly explore the application of the quaternion deep network to the knowledge graph completion task, and expand the method to open domain knowledge graph tasks.

References

[Abboud et al. 2020] Abboud, R.; Ceylan, I. I.; Lukasiewicz, T.; and Salvatori, T. 2020. Boxe: A box embedding model for knowledge base completion. *arXiv preprint arXiv:2007.06267*.

[Balažević, Allen, and Hospedales 2019a] Balažević, I.; Allen, C.; and Hospedales, T. M. 2019a. Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*, 553–565. Springer.

[Balažević, Allen, and Hospedales 2019b] Balažević, I.; Allen, C.; and Hospedales, T. M. 2019b. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

[Bollacker et al. 2008] Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

[Bordes et al. 2013] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, 1–9.

[Chami et al. 2020] Chami, I.; Wolf, A.; Juan, D.-C.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2005.00545*.

[Chen et al. 2020] Chen, S.; Liu, X.; Gao, J.; Jiao, J.; Zhang, R.; and Ji, Y. 2020. Hitter: Hierarchical transformers for knowledge graph embeddings. *arXiv preprint arXiv:2008.12813*.

[Cui et al. 2019] Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.-w.; and Wang, W. 2019. Kbqa: learning question answering over qa corpora and knowledge bases. *arXiv preprint arXiv:1903.02419*.

[Dettmers et al. 2018] Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Fan et al. 2014] Fan, M.; Zhou, Q.; Chang, E.; and Zheng, F. 2014. Transition-based knowledge graph embedding with relational mapping properties. In *Proceedings of the 28th Pacific Asia conference on language, information and computing*, 328–337.

[Gao et al. 2020] Gao, C.; Sun, C.; Shan, L.; Lin, L.; and Wang, M. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 385–394.

[Hamilton 1844] Hamilton, W. R. 1844. Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 25(169):489–495.

[Ji et al. 2015] Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 687–696.

[Ji et al. 2016] Ji, G.; Liu, K.; He, S.; and Zhao, J. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Ji et al. 2020] Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.

[Jia, Cheng, and Su 2020] Jia, N.; Cheng, X.; and Su, S. 2020. Improving knowledge graph embedding using locally and globally attentive relation paths. In *European Conference on Information Retrieval*, 17–32. Springer.

[Kazemi and Poole 2018] Kazemi, S. M., and Poole, D. 2018. Simple embedding for link prediction in knowledge graphs. *arXiv preprint arXiv:1802.04868*.

[Lehmann et al. 2015] Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6(2):167–195.

[Lin et al. 2015] Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

[Nguyen et al. 2017] Nguyen, D. Q.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2017. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*.

[Nickel, Rosasco, and Poggio 2016] Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Nickel, Tresp, and Kriegel 2011] Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

[Parcollet et al. 2018] Parcollet, T.; Zhang, Y.; Morchid, M.; Trabelsi, C.; Linarès, G.; De Mori, R.; and Bengio, Y. 2018. Quaternion convolutional neural networks for end-to-end automatic speech recognition. *arXiv preprint arXiv:1806.07789*.

[Parcollet, Morchid, and Linarès 2017] Parcollet, T.; Morchid, M.; and Linarès, G. 2017. Quaternion denoising encoder-decoder for theme identification of telephone conversations. In *Interspeech 2017*, 3325–3328. ISCA.

[Parcollet, Morchid, and Linarès 2019] Parcollet, T.; Morchid, M.; and Linarès, G. 2019. Quaternion convolutional neural networks for heterogeneous image processing. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8514–8518. IEEE.

[Suchanek, Kasneci, and Weikum 2007] Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, 697–706.

[Sun et al. 2019] Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

[Toutanova and Chen 2015] Toutanova, K., and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, 57–66.

[Trouillon et al. 2016] Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071–2080. PMLR.

[Vashishth et al. 2019] Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

[Vashishth et al. 2020] Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; and Talukdar, P. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3009–3016.

[Vu et al. 2019] Vu, T.; Nguyen, T. D.; Nguyen, D. Q.; Phung, D.; et al. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2180–2189.

[Wang et al. 2014] Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

[Wang et al. 2018] Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, 1835–1844.

[Wang et al. 2020] Wang, H.; Qin, K.; Lu, G.; Luo, G.; and Liu, G. 2020. Direction-sensitive relation extraction using bi-sdp attention model. *Knowledge-Based Systems* 198:105928.

[Yang et al. 2014] Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

[Zhang et al. 2019] Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q.

2019. Quaternion knowledge graph embeddings. *arXiv preprint arXiv:1904.10281*.

[Zhang, Sun, and Zhang 2020] Zhang, S.; Sun, Z.; and Zhang, W. 2020. Improve the translational distance models for knowledge graph embedding. *Journal of Intelligent Information Systems* 55(3):445–467.