Extending Quaternion Knowledge Graph Embeddings

Ross Alexander
Department of Aeronautics & Astronautics
Stanford University
rbalexan@stanford.edu

Victoria Magdalena Dax
Department of Aeronautics & Astronautics
Stanford University
vmdax@stanford.edu

Abstract

As a result of the recent data revolution, knowledge graphs (KGs) have emerged as powerful representational structures for capturing relational data, which are common in disciplines including biomedicine, academic collaboration, programming, chemistry, product sales, and abstract reasoning. These modern knowledge graphs encode rich semantic knowledge of the underlying domain that, if efficiently decoded, has the potential to significantly impact their respective disciplines. In practice, modern KGs are often large, noisy, and incomplete and we seek to complete the KG, that is, predict missing entities, entity features, relations, relation features, and graph features. In this work, we focus on the link prediction task for a biomedical knowledge graph dataset. We first discuss a state-of-the-art method for learning quaternion knowledge graph embeddings (QUATE), and we then introduce several modifications of QUATE designed to improve performance, including modified negative sampling, de-regularization, ℓ_1 regularization, a distance-based scoring and loss function, and staged training procedures. We train knowledge graph embeddings for these models and compare their performance. We find that the original QUATE method delivers comparable, but slightly weaker performance than the other Open Graph Benchmark (OGB) baselines. Our QUATE-DEREG and QUATE-STAGED deliver performance that is superior to all other baselines on the MRR metric.

1 Introduction

In this section, we provide an overview of our work in (1) implementing a state-of-the-art method for learning quaternion knowledge graph embeddings (QUATE) and (2) designing and testing modifications to QUATE on a link prediction task for a biomedical knowledge graph dataset.

1.1 Dataset

For this project, we selected the **ogbl-biokg**[1] biomedical knowledge graph (KG). This knowledge graph, which consists of 93K+ entities and 5M+ relations, was generated from a large number of biomedical data repositories. The dataset includes 5 entity types – diseases (10,687 entities), drugs (10,533 entities), side effects (9,969 entities), proteins (17,499 entities), and protein functions (45,085 entities). The dataset also features 51 directed relation types, including 39 drug-drug relations and 8 protein-protein relations, along with drug-protein, drug-side effect, drug-protein, and function-function relations. Relations connecting the same entity types (e.g., protein-protein, drug-drug, function-function) are always symmetric.

[1] `https://ogb.stanford.edu/docs/linkprop/#ogbl-biokg`

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

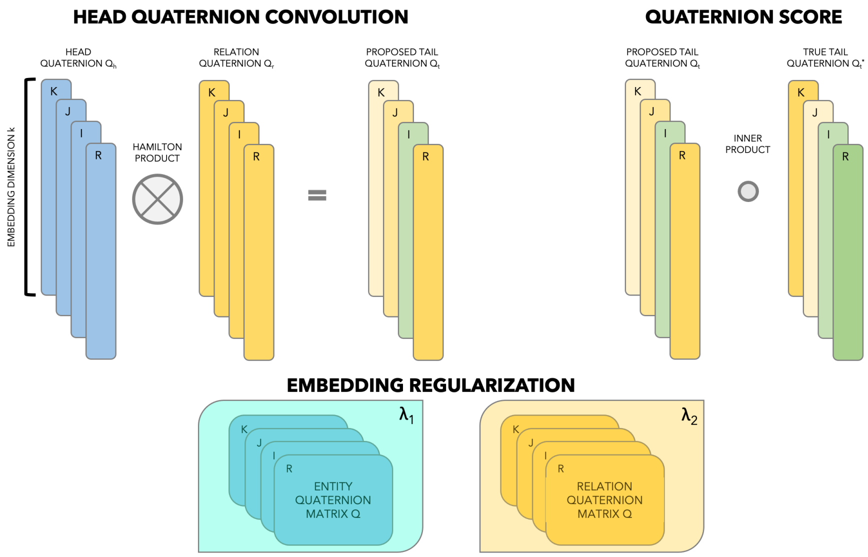

Figure 1: QUATE model architecture. Hamilton product of head quaternion is computed with relation quaternion. Transformed head quaternion (proposed) tail quaternion is scored against true tail quaternion. Embedding matrices are regularized. Quaternion units product: $\{ij = k, ji == k, jk = i, ki = j, kj = -i, ik = -j, i^2 = j^2 = k^2 = ijk = -1\}$

This interesting scientific dataset piqued our interest and we ultimately selected it because the associated knowledge graph completion (link prediction) task is quite challenging and methods for KG completion are an area of active research. One of the challenges of this dataset and also of real-world knowledge graphs is how to perform the completion task over a noisy, incomplete, heterogeneous graph. With regard to the **ogbl-biokg** KG dataset, the observations range from objective (drug-drug interactions, protein-protein interactions) to subjective (drug-side effect relations) and are aggregated from sources with a variety of confidence levels, including experimental readouts, human-curated annotations, and automatically extracted metadata.

1.2 Problem Settings

The problem setting is a link prediction task where we want to predict relations in the KG given the training set. We first randomly split the dataset into training, validation, and test sets of relations. We then learn the knowledge graph embeddings using our training and validation sets. On the validation and test sets we use our model to predict whether a triple (head, relation, tail) is a positive relation (it exists in the graph) or a negative relation (it does not exist in the graph). We track various metrics, including mean reciprocal rank (MRR), hits-at-1 (Hits@1) (a proxy for accuracy), precision (P), and recall (R).

1.3 QUATE: Quaternion Knowledge Graph Embeddings

The state-of-the-art method that we chose to implement is called QUATE. While knowledge graph embeddings are usually done in real (Euclidean) space $\mathbb{R}^k$, recent work has been focused on exploiting properties of other spaces, including complex space $\mathbb{C}^k$, hypercomplex space $\mathbb{H}^k$, toroidal space $\mathbb{T}^k$, and hyperbolic space $\mathcal{H}^k$. Building on the success of complex knowledge graph embeddings in methods like COMPLEX [6] and ROTATE [5], the authors of QUATE formulate a straightforward method for learning hypercomplex (or quaternion) knowledge graph embeddings. Hypercomplex numbers (quaternions) are numbers with one real part and three hypercomplex parts **i**, **j**, and **k**. Due to the intrinsic properties of quaternions, many kinds of relations can be naturally encoded in simple quaternion operations, equipping hypercomplex embedding spaces with strong representational power.

Specifically, QUATE proposes to learn quaternion embeddings for each entity and for each relation type. QUATE computes the Hamilton product of the head entity quaternion with the unit relation quaternion (analogous to a quaternion rotation) and then computes the quaternion inner product of the transformed head entity quaternion with the tail entity quaternion - called the "score" of the triple (head, relation, tail). The quaternion inner product is bounded between $[-1, 1]$ so the score can be trained to be a prediction of the label (+1 for positive edges, -1 for negative edges). QUATE uses a logistic loss with ℓ_2 regularization on the entity embeddings and the relation embeddings. The authors of QUATE conduct experiments over several datasets and scoring functions and find superior performance over real and complex knowledge graph embeddings and, in some cases, decreased parameter complexity.

1.4 Modifications to QUATE

We propose several modifications to improve the performance of QUATE, including de-regularization (QUATE-DEREG), ℓ_1 regularization (QUATE-L1REG), a distance-based scoring and loss function (QUATE-ROTATE), and staged training procedures (QUATE-STAGED).[2] For all models (*including the base* QUATE *model*), we utilize a modified negative sampling procedure since the dataset is quite dense, making negative sampling very time-consuming.

1.5 Overview

In Section 2 we review the motivation and methodology of quaternion knowledge graph embeddings as formulated in QUATE. Section 3 details our modifications that attempt to improve the performance of the QUATE model. We provide our experimental setup and discuss our results in Section 4 and conclude with a summary of our work in Section 5.

2 QUATE: Quaternion Knowledge Graph Embeddings

QUATE is not on the ogbl-biokg leaderboard (or any of the ogbl-* leaderboards).

2.1 Motivation & Prior Work

The authors of QUATE [9] note that many modern semantic applications (e.g. question answering, natural language processing, and search) utilize knowledge graphs, which have enabled powerful relational reasoning. These knowledge graphs, which consist of entities and relations, are structured as directed graphs. In these graphs, directed edges are represented as triples - consisting of a head entity, a relation, and a tail entity. While this semantic framework induces significant representational power, many real-world knowledge graphs are incomplete. As a result, knowledge graph completion has become an essential task that is often framed as link prediction task. Much of the successful prior work in knowledge graph link prediction has focused on techniques for embedding both the entities and relations in a low-dimensional embedding space and then identifying potential triples. A variety of techniques have been developed that largely utilize a particular relational operator, e.g. translation [1, 7, 4] or rotation [5, 6] along with various normalization schemes to determine potential candidate triples. Specifically, most schemes take a proposed head entity embedding, perform the relational operation with a proposed relation embedding, and score the resulting transformed head entity embedding against all observed entity embeddings (as candidate tail entities). Typical scoring functions are inner products, where higher scores indicate likelier triples.

Prior work has thoroughly explored translation and rotation relational operators. Recently, significant attention has been directed to the particular kind of embedding space. Early researchers utilized mostly real spaces ($\mathbb{R}^k$) [1, 7, 4] while recent researchers have explored toroidal spaces ($\mathbb{T}^k$) [3] and complex spaces ($\mathbb{C}^k$) [5, 6, 9]. Of these methods, complex embedding spaces have been the most successful, in part due to the fact that complex space induces a natural anti-symmetry relation (a useful inductive bias), which ultimately leads to increased expressive power and parameter efficiency. The authors of QUATE are motivated by the success of complex embedding space methods and take

[2] We attempted so many methods since we were having trouble finding a method that improves over the baseline model.

the next step to generalize these methods to hypercomplex embedding spaces ($\mathbb{H}^k$), which are spaces with one real part and three hypercomplex parts (**i**, **j**, and **k**).

2.2 Method

The method presented in [9] chooses to learn effective representations of the semantic relations in knowledge graphs by learning quaternion embeddings for entities and relations. This approach leverages the expressive rotational capability of quaternions, which are more efficient and numerically stable than rotational matrices. It can be summarized into two steps:

1. rotate the head quaternion using the unit relation quaternion;
2. take the quaternion inner product of the rotated head and tail quaternions to score each (head, relation, tail) triplet.

More specifically, in a graph G with N entities and M relations, the aim lies in learning the node embedding matrix $Q \in \mathbb{H}^{N\times k}$ and the relation embedding matrix $W \in \mathbb{H}^{M\times k}$, where k denotes the dimension of the embeddings. Evidently, the method defines a new W_r for each relation type. To eliminate adverse scaling effects, each relation quaternion embedding is normalized using the ℓ_2-norm.

$$\hat{W}_r = \frac{W_r}{|W_r|} = \frac{a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}}{\sqrt{a^2+b^2+c^2+d^2}}$$

The scoring function is defined as the quaternion inner product of the rotated head quaternion with the tail quaternion, i.e.

$$\phi(h, r, t) = [Q_h \otimes \hat{W}_r] \cdot Q_t.$$

Here, $\otimes$ denotes the Hamilton product which is defined as

$$\begin{aligned} Q_h \otimes \hat{W}_r = &(a_h \circ p - b_h \circ q - c_h \circ u - d_h \circ v) \\ &+ (a_h \circ q + b_h \circ p + c_h \circ v - d_h \circ u)\mathbf{i} \\ &+ (a_h \circ u - b_h \circ v + c_h \circ p + d_h \circ q)\mathbf{j} \\ &+ (a_h \circ v + b_h \circ u - c_h \circ q + d_h \circ p)\mathbf{k} \end{aligned}$$

with $\circ$ denoting an element-wise multiplication.

The total loss is then defined as a logistic loss with independent regularization on the entity and relation embeddings, which are regularized by coefficients λ_1 and λ_2, respectively.

$$L(Q, W) = \sum_{\langle h,r,t\rangle \in \Omega, \Omega^-} \log(1 + \exp(-Y_{\langle h,r,t\rangle}\phi(h, r, t)) + \lambda_1 ||Q||_2^2 + \lambda_2 ||W||_2^2.$$

The labels for the relation are given by an indicator function of the triplet's presence in the observed set Ω (positive samples) or in the unobserved set Ω^- (negative samples).

$$Y_{\langle h,r,t\rangle} = \begin{cases} +1 & \text{if } \langle h, r, t\rangle \in \Omega \\ -1 & \text{otherwise} \end{cases}$$

Similar to COMPLEX, [6], QUATE can model symmetry - when imaginary parts of W_r are null - and anti-symmetry relations, as well as inversion patterns, by utilizing the conjugation of quaternions (since conjugation is an involution and its own inverse). Regarding composition patterns, a relation between two nodes z and x is not only determined by relations r_1 and r_2 but also simultaneously influenced by the node embeddings themselves. Compared to TRANSE [1] and ROTATE [5], these composition patterns are not fixed, which increases the expressivity of the quaternion representation.

3 Modifications to QUATE

We propose several modifications to improve the performance of QUATE, including de-regularization (QUATE-DEREG), ℓ_1 regularization (QUATE-L1REG), a distance-based scoring and loss function (QUATE-ROTATE), and staged training procedures (QUATE-STAGED). For all models (*including the base* QUATE *model*), we utilize a modified negative sampling procedure since the dataset is quite dense, making negative sampling very time-consuming.

3.1 Modified Negative Sampling

The negative sampling scheme presented in the original QUATE paper uses "head-or-tail" modification. In this sampling scheme, negative samples are generated by modifying the batch positive samples $\langle h, r, t \rangle \in \Omega$, either by (1) resampling a new tail t' such that $\langle h, r, t' \rangle \in \Omega^-$ or (2) resampling a new head h' such that $\langle h', r, t \rangle \in \Omega^-$. Since the dataset is rather dense, especially for drug-drug relations, the negative sampling scheme was rarely able to generate valid negative samples using this method, leading to long computation times. Our modified negative sampling scheme allows generation of arbitrary triples that are not "head-or-tail" modified negative samples. This enables faster negative sampling, and we can further view our sampling strategy as decoupling samples within a batch.

3.2 Model Modifications

QUATE-DEREG: De-regularization of entity and relation embedding matrices We believe that the regularization terms in the original QUATE loss function weaken the formulation by unnecessarily compressing the quaternion entity and relation embeddings. Since quaternions multiplication naturally represents rotations and dilations, and since QUATE already normalizes the relation quaternion, we believe there is little value in adding regularization over the Q and W entity and relation embedding matrices. As a result, for this method, we set the regularization parameters $\lambda_1 = 0$ and $\lambda_2 = 0$.

QUATE-L1REG: ℓ_1-regularization of entity and relation embedding matrices While ℓ_2-regularization leads to an overall (likely unnecessary) compression of the general magnitude of entity and relation quaternions, we propose to consider an ℓ_1-regularization instead since it leads to sparser representations. Taking a channel capacity viewpoint, we believe that sparsifying quaternions can help force particular entity types, relation types, or triplet types to operate on specific quaternion parts, leaving other unnecessary part set to zero. This frees up the channel for other entity types, relation types, or triplet types to specialize. We postulate that this improves exploitation of hypercomplex space's natural relational properties.

QUATE-STAGED: Staged training procedure One of the major challenges we faced in training the original QUATE method was the slow model training procedure, which takes a long time to complete even a single epoch. To alleviate this, we investigated using a staged training procedure that would allow specific entity types, relation types, or triplet types to be learned first, while others are learned later. The staged training procedure would essentially consist of initializing the entity and relation embedding matrices as usual (over the entire graph), performing an epoch of training over mini-batches formed from a batch consisting only of a single triplet type. After the first epoch, we add all of the training data for an additional type and perform an epoch of training. The procedure continues until the full training set is re-formed. We believe that this training procedure can lead to faster learning, especially in large heterogeneous graph datasets.

QUATE-ROTATE: ROTATE-based norm-scoring and margin-based loss function The original QUATE inner product scoring function $\phi(h, r, t)$ and regularized logistic loss function $L(Q, W)$ make sense when framed as logistic regression problem, though we wanted to investigate if a norm-based distance scoring function and a regularized margin-based loss could improve the performance. We propose to consider a modified version of the ROTATE [5] norm-based distance scoring function and simplified noise contrastive estimation (NCE) margin-based loss function with negative sampling:

$$\begin{aligned}\phi(h, r, t) &= ||Q_h \otimes \hat{W}_r - Q_t||_2 \\ L(Q, W) &= - \sum_{\langle h, r, t \rangle \in \Omega, \Omega^-} \log \sigma(Y(\gamma - \phi(h, r, t))) + \lambda_1 ||Q||_2^2 + \lambda_2 ||W||_2^2\end{aligned}$$

where positive samples and negative samples are drawn at a 1-to-1 ratio, σ is the sigmoid function, and $\gamma > 0$ is the margin parameter.

Table 1: Base Model Hyperparameters

Hyperparameter	Description	Value
k	Embedding dimension	500
λ_1	Entity embedding ℓ_2 regularization	0.01
λ_2	Relation embedding ℓ_2 regularization	0.05
α	Learning rate	1×10^{-4}
n	Batch size	1024

4 Experiments

4.1 Experimental Setup

For the experiments, we utilized the **ogbl-biokg** dataset. Using the Open Graph Benchmark toolbox[3], we load in the dataset and randomly split the dataset into training, validation, and test sets. Our training-validation-testing split is 94%/3%/3%, giving us a training set containing 4.7M+ relations, and a validation and test set containing 162K+ relations.

We settled on the base model hyperparameters listed in Table 1. We use an embedding dimension of $k = 500$, which we believe has significant expressivity given the size of our dataset. We ℓ_2-regularize the entity and relation embedding matrices using $\lambda_1 = 0.01$ and $\lambda_2 = 0.05$. We train our model using mini-batches of the training set of size $n = 1024$ using the ADAM (Adaptive Moment Estimation) optimizer with a learning rate $\alpha = 1 \times 10^{-4}$ and tune the model hyperparameters using the validation set. For the QUATE-DEREG method, the regularization parameters are set to 0. For the QUATE-ROTATE method, the margin parameter is set to $\gamma = 12$. Our code is available at `https://github.com/victorialena/easy_quatE`.

4.2 Results and Discussion

Our results are provided in Table 2.[4,5,6]

Table 2: Results

	Test Set				Validation Set			
Model	**MRR**	**Hits@1**	**P**	**R**	**MRR**	**Hits@1**	**P**	**R**
PAIRRE[2]	0.8164	-	-	-	0.8172	-	-	-
COMPLEX[6]	0.8095	-	-	-	0.8105	-	-	-
DISTMULT[8]	0.8043	-	-	-	0.8055	-	-	-
ROTATE[5]	0.7989	-	-	-	0.8055	-	-	-
TRANSE[1]	0.7452	-	-	-	0.7456	-	-	-
QUATE[9]	0.7652	0.5690	0.6126	0.4585	0.7995	0.5990	0.6391	0.5982
QUATE-DEREG	**0.8996**	**0.8626**	0.6663	**0.9959**	**0.9396**	**0.8793**	0.8189	**0.9980**
QUATE-L1REG	-	-	-	-	-	-	-	-
QUATE-STAGED	0.8954	0.8345	**0.6852**	0.9885	0.9382	0.8684	**0.8265**	0.9912
QUATE-ROTATE	-	-	-	-	-	-	-	-

For all of the OGB baselines, which report only the mean reciprocal rank (MRR) metric, the values range from 0.80-0.82. Our QUATE method gives an MRR of 0.765, which is just above the TRANSE baseline value.

We can see that the QUATE-DEREG model formulation performs best on most metrics including MRR, accuracy - Hits@1-, and recall. The precision dropped significantly from the validation to

[3] `https://ogb.stanford.edu`

[4] Results for non-QUATE models are collected from `https://ogb.stanford.edu/docs/leader_linkprop/#ogbl-biokg`.

[5] We were not able to generate confidence intervals for our models due the long model training times.

[6] We were not able to run the QUATE-L1REG and QUATE-ROTATE models due to computational constraints.

the test set suggesting that we overfit our model. The learning curve can be found in 2, showing the validation loss in red and training loss in blue.

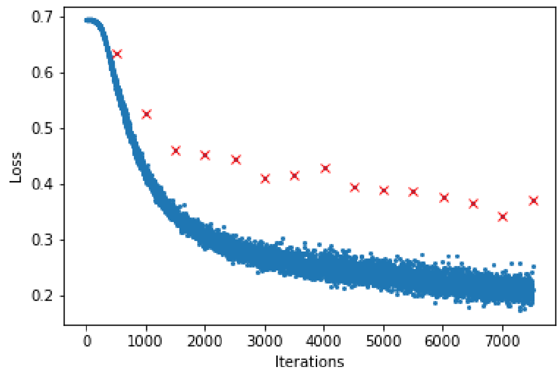

Figure 2: Training loss curve (blue) and validation loss curve (red) over minibatch iterations for the QUAT-DEREG method ($\alpha = 1E - 4$).

We note that the QUATE-STAGED model performs on par with the de-regularized one. However, we would like to point out that the staged model at L1 trained on 5 epoch in 1:22h, while the full model ran 3 epochs in 6:10h. Thus we conclude that the staged model would be a promising architecture to pursue as a learning scheme, progressively building complexity in the embedding space while keeping its expressiveness.

5 Conclusion

In this work, we focused on the link prediction task for a biomedical knowledge graph dataset. We presented the state-of-the-art method for learning quaternion knowledge graph embeddings (QUATE) and proposed several modifications of QUATE designed to improve performance. In particular, we introduced a modified negative sampling scheme and a staged training scheme that improved computational tractability of the model training. We also introduced various modified loss schemes, including de-regularization, ℓ_1 regularization, and a norm-based scoring function and margin-based loss function. We trained knowledge graph embeddings for these models found that QUATE delivers performance that is on-par, but not superior to other methods for the **ogbl-biokg** dataset, and our QUATE-DEREG and QUATE-STAGED deliver performance that exceed all other baseline models on the reported MRR metric.

For future work, we would like to have the chance to generate results for all of the proposed modifications we made in the paper (i.e. completing the QUATE-L1REG and QUATE-ROTATE results, as well as generating confidence intervals for all methods). There are interesting avenues for future work - for quaternion embeddings, it may be useful to learn position-specific embeddings, that is we learn a set of head-specific entity embeddings, relation embeddings, an tail-specific entity embeddings. Since different entities may participate as heads or tails in different methods, different head and tail embeddings for the same entity may be useful in supporting different relational behaviors. We see a natural extension of the QUATE method to octonion systems (systems with one real part and seven hypercomplex parts). It may also be interesting to investigate the formulation of a generalized Hamilton product operator with learnable signs or coefficients. We could see this extending to non-binary powers (that is, not simply real (1), complex, (2), quaternion (4), octonion (8), etc. systems), providing additional flexibility by eliminating mathematical closure requirements.

Contributions

Ross: Wrote report (text, table, figures).
Victoria: Wrote model (code, values, readme).
Both: collaborated on proposed model changes.

References

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[2] Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors, 2021.

[3] Takuma Ebisu and Ryutaro Ichise. Toruse: Knowledge graph embedding on a lie group. *CoRR*, abs/1711.05435, 2017.

[4] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015.

[5] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *CoRR*, abs/1902.10197, 2019.

[6] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2016.

[7] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes, 2014.

[8] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases, 2015.

[9] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embedding. *CoRR*, abs/1904.10281, 2019.