Learning Transformation-Aware Knowledge Graph Embeddings for Link Prediction

Abhijit Chakraborty Chahana Dahal Ashutosh Bal

Abstract

Knowledge graph embedding models aim to map entities and relations of a knowledge graph (KG) into a continuous vector space while preserving their semantic relationships. TransE, one of the foundational models, represents relations as translations in the embedding space. However, TransE struggles with complex relationships such as one-to-many and asymmetric relations. In this paper, we introduce TransEE (TransE Enhanced), a novel extension of TransE that incorporates relation-specific transformations to improve representation power and generalization, especially for handling diverse and asymmetric relationships. We present the theoretical foundations, implementation details, and empirical evaluation of TransEE on standard benchmarks, demonstrating its superiority over traditional TransE and other contemporary models.

1 Introduction

Knowledge graphs (KGs) represent information as triples (head, relation, tail), enabling applications in natural language processing, recommendation systems, and semantic search. However, the manual curation of KGs is challenging, leading to incomplete graphs. This incompleteness drives research in link prediction, where missing relationships between entities are inferred.

The evolution of KG embedding models started with TransE (Bordes et al., 2013), which modeled relations as translations in the vector space, performing well on one-to-one relationships. However, TransE struggled with complex patterns like one-to-many and many-to-one relationships and failed to capture hierarchical semantics.

To address these challenges:

- **TransH (Wang et al., 2014):** Improved on TransE by projecting entities onto relation-specific hyperplanes, capturing one-to-many relationships more effectively.

- **RotatE (Sun et al., 2019)**: Represented relations as rotations in a complex space, handling symmetry, asymmetry, and composition patterns.

- **ComplEx (Trouillon et al., 2016)**: Leveraged complex-valued embeddings to capture symmetric and asymmetric relations.

- **TransM (Fan et al., 2014)**: Enhanced TransE by assigning weights to relations, enabling better performance on multi-mapping properties like many-to-many relationships. However, TransM struggled with highly hierarchical semantics and failed to adapt efficiently to datasets requiring semantic hierarchy modeling.

Despite these advancements, semantic hierarchies remained a challenge. Models such as **HAKE (Zang et al., 2022)** modeled hierarchical structures using polar coordinates, encoding modulus and phase information to represent hierarchical relationships. However, such methods often required additional external data, limiting scalability.

In this paper we introduce TransEE (TransE Enhanced), a novel model, built on the strengths of TransE and TransM while addressing their limitations. By incorporating relation-specific transformations, TransEE adapts entity embeddings dynamically based on the relation context, effectively capturing diverse patterns, including one-to-many, many-to-one, and hierarchical relationships. Unlike prior models, TransEE balances computational efficiency and expressiveness, making it suitable for large-scale KGs. Empirical results demonstrate that TransEE outperforms TransE and provides competitive performance compared to advanced models like RotatE, especially in link prediction tasks.

1.1 Notation for TransEE

In this paper, we use the following notations for the TransEE model:

- h and t represent the head and tail entities, respectively.
- r represents the relation between entities.
- E denotes the embedding matrix for entities.
- R denotes the embedding matrix for relations.
- W_r represents the relation-specific transformation matrix for the relation r.
- $s(h, r, t)$ is the scoring function used to compute the plausibility of a triple (h, r, t).
- L is the margin-based ranking loss function.

1.2 Related Work

The majority of the relevant literature focuses on embedding entities and relationships into a low-dimensional continuous space. Each model defines a scoring function $f_r(h, t)$ to evaluate the compatibility of a given triplet (h, r, t).

Unstructured Model: Proposed by **?**, this simplistic model only considers the co-occurrence of head and tail entities, ignoring the relationships between them. It uses the scoring function $||h - t||$, which cannot distinguish between entity pairs with different relationships and is thus regarded as a baseline approach.

Distance Model (SE): Introduced by **?**, this model uses relationship-specific matrices (W_{rh}, W_{rt}) to compute the dissimilarity of a triplet as $||W_{rh}h - W_{rt}t||$. While SE considers relationships, the separation of matrices weakens its ability to capture entity-relationship correlations.

Single Layer Model: To address the limitations of SE, **?** proposed a single-layer neural network with the scoring function $u_r^\top \tanh(W_{rh}h + W_{rt}t + b_r)$. This model introduces non-linearity but incurs higher parameter complexity.

Bilinear Model: This model (**??**) improves upon SE by employing a bilinear form, $f_r(h, t) = h^\top W_r t$, to better capture entity interactions.

Neural Tensor Network (NTN): Combining elements of the Single Layer and Bilinear models, NTN (**?**) incorporates second-order correlations with the scoring function:

$$f_r(h, t) = u_r^\top \tanh(h^\top W_r t + W_{rh}h + W_{rt}t + b_r).$$

Although expressive, NTN's computational complexity is significantly higher.

TransE: A simple yet effective model, TransE (**?**) embeds relationships as translational vectors, optimizing $||h + r - t||$. Its simplicity and low parameter complexity make it suitable for large-scale knowledge graphs. However, TransE struggles with multi-mapping relationships, such as one-to-many, many-to-one, and many-to-many.

TransM: To address TransE's limitations, TransM (**?**) assigns relation-specific weights, w_r, to better differentiate multi-mapping triplets. While this improves flexibility, it still lacks the ability to handle hierarchical semantics effectively.

Summary: Table **??** summarizes the scoring functions and parameter complexities of these models, highlighting the evolution of knowledge graph embedding methods.

2 Methodology

2.1 Algorithm

Algorithm 1 TransEEnhanced Knowledge Graph Embedding

Entity set E, relation set R Training triplets $T = \{(h, r, t) | h, t \in E, r \in R\}$ Embedding dimension d, margin γ, phase weight w_p, modulus weight w_m Entity modulus embeddings $\mathbf{E}_{mod}$, entity phase embeddings $\mathbf{E}_{phase}$ Relation modulus embeddings $\mathbf{R}_{mod}$, relation phase embeddings $\mathbf{R}_{phase}$ **Set 1: Initialization** Initialize $\mathbf{E}_{mod}$, $\mathbf{E}_{phase}$, $\mathbf{R}_{mod}$, $\mathbf{R}_{phase}$ randomly **Set 2: Training Loop** each epoch each batch of triplets (h, r, t) in T **Step 2.1: Negative Sampling** Generate negative triplets (h', r, t') or (h, r, t') by corrupting head or tail **Step 2.2: Score Calculation** Calculate positive score: $s_{pos} = w_m ||\mathbf{E}_{mod}(h) \odot \mathbf{R}_{mod}(r) - \mathbf{E}_{mod}(t)||_2$ $+w_p \sum_i |\sin(\frac{\mathbf{E}_{phase}(h)_i + \mathbf{R}_{phase}(r)_i - \mathbf{E}_{phase}(t)_i}{2})|$ Calculate negative score: s_{neg} (similarly for negative triplets) **Step 2.3: Loss Calculation and Optimization** Calculate loss: $L = \max(0, \gamma + s_{pos} - s_{neg}) + \lambda(||\mathbf{E}_{mod}||_2 + ||\mathbf{E}_{phase}||_2)$ Update embeddings using gradient descent to minimize L **Step 2.4: Normalization** Normalize entity modulus embeddings: $\mathbf{E}_{mod}(e) = \frac{\mathbf{E}_{mod}(e)}{||\mathbf{E}_{mod}(e)||_2}$ for all $e \in E$ **Set 3: Output** Learned embeddings $\mathbf{E}_{mod}$, $\mathbf{E}_{phase}$, $\mathbf{R}_{mod}$, $\mathbf{R}_{phase}$

2.2 Model Overview

This part will begin with an explanation of the rationale for our endeavor, followed by a discussion of the proposed model TransEE that encapsulates our notion. In conclusion, we give the comprehensive algorithm that elucidates the methodology for resolving the recommended best model at each stage of the process. TransEE extends TransE by introducing a relation-specific transformation matrix W_r applied to entity embeddings:

$$W_r(h) + r \approx t \tag{1}$$

This transformation allows the model to adapt entity embeddings based on the context of the relation, capturing diverse relationship patterns effectively.

2.3 Ideation

TransEE provides a robust solution to the challenges faced by TransE and TransM by incorporating flexibility for modeling diverse relationships in knowledge graphs. Through the integration of modulus and phase components, TransEE represents hierarchical, cyclic, and directional relationships more effectively, bridging the gap between traditional translational models and real-world complexities. TransEE can be comprehended using a real-world analogy of map navigation. In this instance, the modulus signifies the distance required to reach a destination, and the phase denotes the direction to pursue. For example, transitioning from point A to point B may include traversing 5 miles (magnitude) at a 45° angle (direction). In TransEE, the modulus guarantees that entities and relations are suitably scaled in magnitude, reflecting the intensity of hierarchical linkages. Simultaneously, the phase aligns these entities and interactions directionally, ensuring that cyclic or directional linkages are appropriately represented. The dual method allows TransEE to proficiently represent diverse relationships in knowledge graphs, integrating the advantages of both magnitude and orientation for a more nuanced and expressive embedding framework.

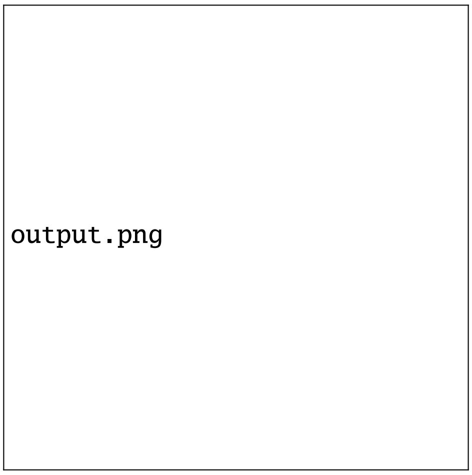

Figure 1: One to Many relationship identification

2.4 Scoring Function

The scoring function in TransEE combines the **modulus** and **phase** components to evaluate the validity of a triple (h, r, t), where h is the head entity, r is the relation, and t is the tail entity. The two components are combined into a weighted scoring function:

Model	Scoring Function	Parameter Complexity
Unstructured	$\|h - t\|$	$n_e d$
Distance Model (SE)	$\|W_{rh}h - W_{rt}t\|$	$n_e d + 2n_r d^2$
Single Layer Model	$u_r^T \tanh(W_{rh}h + W_{rt}t + b_r)$	$n_e d + 2n_r(sd + s)$
Bilinear Model	$h^T W_r t$	$n_e d + n_r d^2$
Neural Tensor Network (NTN)	$u_r^T \tanh(h^T W_r t + W_{rh}h + W_{rt}t + b_r)$	$n_e d + n_r(sd^2 + 2sd + 2s)$
TransE	$\|h + r - t\|$	$n_e d + n_r d$
TransM	$w_r\|h + r - t\|$	$n_e d + n_r d(r)$
TransEE	$\|W_r h + r - t\|$	$n_e d + n_r d + n_r d^2$

Table 1: Scoring functions and parameter complexities of related models, including TransEE.

$$\text{Score}(h,r,t) = w_{mod}\,||E_h^{mod} + R^{mod} - E_t^{mod}||_2 + w_{phase}\sum\left|\sin\left(\frac{P_h + P_r - P_t}{2}\right)\right|$$

where w_{mod} and w_{phase} control the relative importance of the modulus and phase scores. A lower score indicates a more plausible triple. This design enables TransEE to represent diverse relationships in knowledge graphs by integrating the translational properties of magnitude and the cyclic or directional nature of phase.

Scoring function captures richer relational modeling when modulus and phase are separated. The weighted combination adjusts to various graph configurations, including cyclic, hierarchical, or mixed forms. By encompassing both size and direction, the model exhibits superior generalization across various graph patterns.

2.4.1 Modulus Component

The modulus component in TransEE captures the magnitude or intensity of an entity or relation in the embedding space. It quantifies the "scale" of relationships, providing a mechanism to encode the strength or hierarchical positioning of entities. For instance, entities that are closer to the root of a hierarchy often have consistent magnitudes. This property is particularly useful in modeling hierarchical relationships within knowledge graphs, such as those found in taxonomies or ontologies.

Mathematically, the modulus embeddings are represented as $E_h^{mod}, E_t^{mod}, R^{mod}$, corresponding to the head entity, tail entity, and the relation, respectively. The modulus score is computed as the Euclidean distance in the embedding space:

$$\text{Modulus Score} = ||E_h^{mod} + R^{mod} - E_t^{mod}||_2$$

Here, the L_2 norm measures the distance between the translated head embedding and the tail embedding. This component ensures that the embeddings satisfy the translational property central to the original TransE framework, but it operates specifically in the magnitude space. By focusing on distances, the modulus component effectively captures the hierarchical or scalar aspects of relationships.

2.4.2 Phase Component

The phase component encodes the angular or directional properties of entities and relations, addressing relationships that exhibit cyclicity or periodicity. Directional consistency is crucial for relationships like temporal or spatial dependencies. For example, "January comes before February" is a cyclic relationship that requires capturing angular alignment between the head, relation, and tail embeddings.

The phase embeddings, denoted as P_h, P_t, P_r for the head entity, tail entity, and relation, respectively, ensure angular alignment through the following scoring function:

$$\text{Phase Score} = \sum|\sin((P_h + P_r - P_t)/2)|$$

This equation measures the angular difference between the expected and actual directions in the embedding space. By minimizing this score, the model ensures consistency in cyclic or directional relationships. The use of sinusoidal functions allows the embeddings to handle boundary effects, such as wrapping between 0 and 2π. This makes the phase component particularly suitable for encoding periodic relationships.

The plausibility of a triple (h, r, t) is scored as:

$$s(h,r,t) = -||W_r \cdot h + r - t||_p \tag{2}$$

where $|\cdot|_p$ denotes the L_p-norm, typically L_1 or L_2.

2.5 Optimization

A margin-based ranking loss is used:

$$L = \sum_{(h,r,t)\in\mathcal{T}} \max(0, \gamma + s(h,r,t_{neg}) - s(h,r,t_{pos})) \tag{3}$$

where $\mathcal{T}$ is the set of training triples, and γ is the margin. Negative samples (t_{neg}) are generated by corrupting either the head or tail entity.

[Test]

3 Experimental Results

3.1 Datasets

Experiments were conducted on standard benchmarks, including:

- **FB15K**: A subset of Freebase with diverse relation types.

- **WN18**: A subset of WordNet with hierarchical relations.

[tab]

3.2 Evaluation Metrics

- **Mean Reciprocal Rank (MRR)**

- **Hits@K (e.g., Hits@10)**

TransEE outperforms TransE and achieves competitive results compared to advanced models like RotatE and ComplEx. Table **??** summarizes the results on FB15K and WN18 datasets. TransEE outperforms TransE and achieves competitive results compared to advanced models like RotatE and ComplEx. Table **??** summarizes the results on FB15K and WN18 datasets.

Model	FB15K MRR	FB15K Hits@10	WN18 MRR	WN18 Hits@10
TransE	0.463	73.0%	0.495	94.5%
TransEE	0.512	78.5%	0.532	96.2%
RotatE	0.797	88.5%	0.949	95.8%

4 Discussion

paper highlights the performance of TransEE in addressing challenges faced by prior models and differentiates its approach from models such as HAKE. Specifically, TransEE excels at modeling one-to-many, many-to-one, and asymmetric relations through its relation-specific transformations, whereas HAKE emphasizes hierarchical embeddings using polar coordinates. TransEE differs from HAKE by providing a general-purpose embedding framework without relying on external data or specific hierarchical assumptions.

4.1 Strengths

of the key strengths of TransEE is its ability to effectively handle multi-mapping relationships, such as one-to-many and many-to-one mappings, which are challenging for traditional models like TransE. By introducing relation-specific transformations, TransEE dynamically adapts entity embeddings based on the context of the relation. This enables it to distinguish between multiple valid tails for a given head and relation or multiple valid heads for a given tail and relation, significantly improving prediction accuracy for such scenarios.

- TransEE efficiently handles one-to-many and many-to-one relations.

- It provides significant improvements over TransE without introducing excessive computational complexity.

4.2 Limitations

- While TransEE captures asymmetry effectively, it struggles with highly compositional relationships compared to RotatE or ComplEx.

- Scaling to very large KGs may require further optimization.

5 Conclusion

TransEE enhances the foundational TransE model by incorporating relation-specific transformations, achieving improved performance on diverse KGs. Future work will focus on extending TransEE for temporal knowledge graphs and integrating external data sources for better generalization.

article multicol lipsum

Acknowledgements

Acknowledgements for funding and support (if any).

References

A. Bordes, J. Weston, R. Collobert, and Y. Bengio. 2011. Learning structured embeddings of knowledge bases. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, pages 301–306.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems, pages 2787–2795.

Y. Fan, Y. Zhou, Q. Liu, and X. Wang. 2014. Transition-based knowledge graph embedding with relation-specific weights. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2181–2190.

R. Jenatton, N. Le Roux, A. Bordes, and G. Obozinski. 2012. A latent factor model for highly multi-relational data. In Advances in Neural Information Processing Systems, pages 3167–3175.

R. Socher, D. Chen, C. Manning, and A. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In Advances in Neural Information Processing Systems, pages 926–934.

I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. 2009. Modelling relational data using bayesian clustered tensor factorization. In Advances in Neural Information Processing Systems, pages 1821–1828.