Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 10
ISET 2025

Advancements in Natural Language Processing: A Study of Knowledge Graph Embedding Techniques

Jinsong Liu

School of Computer Science, Wuhan University, Wuhan, Hubei, 430072, China

2022302111316@whu.edu.cn

Abstract. The use of Natural Language Processing (NLP) has made it possible for machines to understand, interpret, and produce human language, making it a cornerstone of artificial intelligence. The ability to represent and reason with structured knowledge is crucial for advancing NLP capabilities. Knowledge Graphs (KGs) offer a powerful way to model entities and their relationships, and learning low-dimensional vector representations of these components is the objective of this technique. This paper delves into the study of NLP technologies, with a specific focus on Knowledge Graph Embedding (KGE) methods. The paper explored the principles and application of translation-based embedding models, particularly TransE, by detailing its training methodology on a standard benchmark dataset (FB15k-237). The research content involves data preprocessing, model initialization with specific embedding dimensions, an iterative training process utilizing margin-based loss, and evaluation through loss convergence and t-SNE visualization of learned entity embeddings. The results demonstrate the model's ability to converge effectively, as evidenced by a significant reduction in training loss over epochs. Furthermore, visualizations reveal distinct clustering of entity types, indicating that the model successfully captures semantic similarities and differences. This study underscores the efficacy of KGE models in learning meaningful representations from structured data, paving the way for enhanced accomplishment in numerous downstream NLP tasks such as linking prediction and entity classification.

Keywords: Natural Language Processing, Knowledge Graph Embedding, TransE, Semantic Similarity, Vector Representation.

1. Introduction

Natural Language Processing (NLP) is a core AI field enabling computers to be capable of processing, comprehending, and creating human language [1]. Its widespread applications, from machine translation to virtual assistants, highlight the increasing need for advanced techniques as digital language data explodes. Complementing this, Knowledge Graphs (KGs) are crucial. These structured representations of facts and relationships provide rich context, significantly enhancing NLP by enabling more accurate language interpretation, disambiguation, and robust knowledge reasoning. The synergy between NLP and KGs is vital for true language understanding and reasoning.

The pursuit of more nuanced language understanding has led to the development of sophisticated models capable of learning rich representations of textual data. Pre-trained language models, such as BERT, have revolutionized the field by learning contextual embeddings from vast amounts of text, leading to state-of-the-art performance on a wide array of NLP tasks, including extractive summarization [2]. Concurrently, the representation of structured knowledge has gained significant attention, with Knowledge Graphs (KGs) emerging as a powerful paradigm. KGs represent entities and their interrelations as a graph structure, and Knowledge Graph Embedding (KGE) techniques seek to embed these components into continuous low-dimensional vector spaces. Early influential KGE models like TransE introduced the concept of modeling relationships as translations in the embedding space, demonstrating remarkable efficiency and effectiveness. Subsequent models, such as TransH and TransR, built upon this by introducing more complex mechanisms like relation-specific hyperplanes and relation-specific spaces to handle more intricate relational patterns [3]. These advancements highlight a continuous effort to refine how semantic relationships are captured and utilized by computational models.

For example, a study by Zheng et al focused on enhancing the efficiency of KGE models for large-scale knowledge graphs. Their research, based on a novel sampling strategy and optimized training objective, achieved a significant improvement in computational efficiency, the results show a 2× ~ 5× speedup over the best competing approaches [4]. This work exemplifies the ongoing drive to develop more scalable and practical KGE techniques.

This paper investigates NLP technologies by focusing on the practical implementation and analysis of the TransE model for knowledge graph embedding. The content explores the model's training dynamics and its ability to generate meaningful entity representations from the FB15k-237 dataset [5]. The purpose is to provide an empirical understanding of TransE's behavior and its effectiveness in capturing semantic structures within a knowledge graph.

2. Methods

This section details the methodology employed to train and evaluate the TransE knowledge graph embedding model. The process encompasses dataset selection and preparation, a description of the TransE model architecture and its core principles, and the design of the experimental setup for training and visualization.

2.1. Dataset

The primary dataset utilized for the experiments is FB15k-237. This dataset is a well-established benchmark in knowledge graph completion research and is a subset of Freebase, a large collaborative knowledge base [5]. FB15k-237 is specifically curated to remove inverse relations present in its predecessor, FB15k, to provide a more challenging evaluation scenario. It consists of a collection of triples, each structured as (head entity, relation, tail entity), denoted as (h,r,t). These triples represent factual statements where h and t are entities and r is the relationship connecting them.

The dataset for KGE model training and evaluation is typically partitioned into three distinct sets. The training set is utilized to train the KGE model, where entity and relation embeddings are learned through the optimization of a scoring function over known facts. The validation set serves the purpose of tuning the model's hyperparameters and implementing early stopping during the training process, a crucial step in preventing overfitting. Finally, the test set is employed for the ultimate evaluation of the trained model's performance on unseen data. This evaluation is typically carried out on tasks such as link prediction, providing an unbiased assessment of generalization capability.

In addition to the triples, the dataset often includes entity type information, which maps entities to their semantic types (e.g., 'person', 'location', 'organization') [5]. This type of information can be valuable for analyzing the learned embeddings, for instance, by visualizing whether entities of the same type cluster together in the embedding space. The provided experimental setup involves reading these triples and entity types, mapping them to unique numerical IDs for processing by the model. For the experiments described in the provided documents, the FB15k-237 dataset comprised 14,541 unique entities and 237 unique relations.

2.2. Model

The TransE model is a foundational and widely recognized KGE technique that represents entities and relations as vectors in the same low-dimensional continuous space. The core idea behind TransE is that for a given triple (h,r,t), the embedding of the tail entity t should be in line with the embedding of the head entity h and the relation r. This can be expressed as:

$$h + r \approx t \tag{1}$$

where $h, r, t \in R^t$ are the k-dimensional vector embeddings for the head entity, relation, and tail entity, respectively. The dimension k is a hyperparameter, often referred to as emb-size [6].

Initialization of entity and relation embeddings typically occurs randomly prior to training. In the provided scripts, the relevant class facilitates this process through the use of a specialized neural network layer designed for handling discrete inputs.

To measure the plausibility of a triple (h,r,t), TransE uses a distance-based scoring function [7]. A common choice is the L_1 or L_2 norm of the difference (h+r−t). A lower score indicates a more plausible triple.

$$f(h,r,t) = -\|h + r - t\|_{1/2} \tag{2}$$

The negative sign is often used so that higher scores correspond to more plausible triples if optimization aims to maximize the score.

TransE is typically trained using a ranking loss function based on margins. The objective is to ensure that the scores of valid (positive) triples are higher (or their distances lower) than the scores of corrupted (negative) triples. To generate a negative triple, a random entity from the KG is substituted for either the head or tail entity of a positive triple. The loss function can be formulated as:

$$\tau = \sum_{(h,r,t)\in S}\sum_{(h',r,t')\in S'}[\gamma + d(h + r - t) - d(h' + r - t')]_+ \tag{3}$$

S represents the set of negative triples, d is the dissimilarity measure (e.g., L2 norm), the margin hyperparameter is measured by γ>0, while the hinge loss is measured by $[x]_+ = \max(0, x)$.

2.3. Experimental Design

The experimental design for this study focuses on training the TransE model, evaluating its learning progress, and visualizing the learned embeddings.

The training process begins with data loading and preprocessing. The Corpus class handles loading the train, valid, and test triples, along with entity type information. This procedure involves mapping entities and relations to unique integer identifiers and preparing the data for batch processing. If a preprocessed data file, such as a previously saved pickle file, is available, it is loaded to save time. Following this, the TransE model is initialized using the total number of unique entities and relations from the corpus, alongside a specified embedding dimension, for example, 100. Model training proceeds for a predefined number of epochs, typically 200. Within each epoch, the training data is iterated through in batches. For each batch, positive triples are sampled from the training set. Corresponding negative triples are then generated by corrupting either the head or tail entity of these positive triples. The model calculates scores for both positive and negative triples within the current batch. The loss, such as margin ranking loss, is computed based on these scores. Subsequently, gradients are backpropagated, and an optimizer, like Adam with a learning rate of 0.001, updates the model parameters, which include both entity and relation embeddings. Finally, the model's state dictionary, containing the learned embeddings, is saved periodically (e.g., every 50 epochs) and upon the completion of training. This practice facilitates subsequent evaluation or the resumption of training.

For visualization, after training, the learned entity embeddings are extracted for a selected group of entity types. Given that embeddings are typically high-dimensional (e.g., 100D), t-Distributed Stochastic Neighbor Embedding (t-SNE) is employed to reduce their dimensionality to two dimensions for visual inspection. t-SNE is a technique particularly well-suited for visualizing high-dimensional datasets by assigning each data point a location in a two or three-dimensional map [8]. The display function utilizes t-SNE with parameters such as a perplexity of 30, two components, PCA initialization, and 5000 iterations. The two-dimensional entity embeddings are then presented as a scatter plot, where different colors represent distinct entity types. This visualization aids in qualitatively assessing whether the model has successfully grouped semantically similar entities. An important preprocessing step involves converting the list of entity responses to a NumPy array before its input into t-SNE.

3. Resualts

The training process of the TransE model on the FB15k-237 dataset was meticulously monitored by tracking the loss per epoch, which provides crucial insight into the model's convergence during training. As illustrated in Figure 1, the training loss curve depicts the progression of the model's error over 200 training epochs. The x-axis, labeled "Epoch," represents the number of training iterations completed, while the y-axis, labeled "Loss," indicates the magnitude of the model's prediction error. Initially, a rapid decrease in loss is observed, signifying the model's quick learning from the training data. This steep decline flattens out significantly after approximately 20-30 epochs, indicating that the model has largely converged and is further refining its embeddings at a slower rate. The consistent downward trend throughout the 200 epochs demonstrates effective learning and the absence of clear overfitting signs based on this metric alone [9].

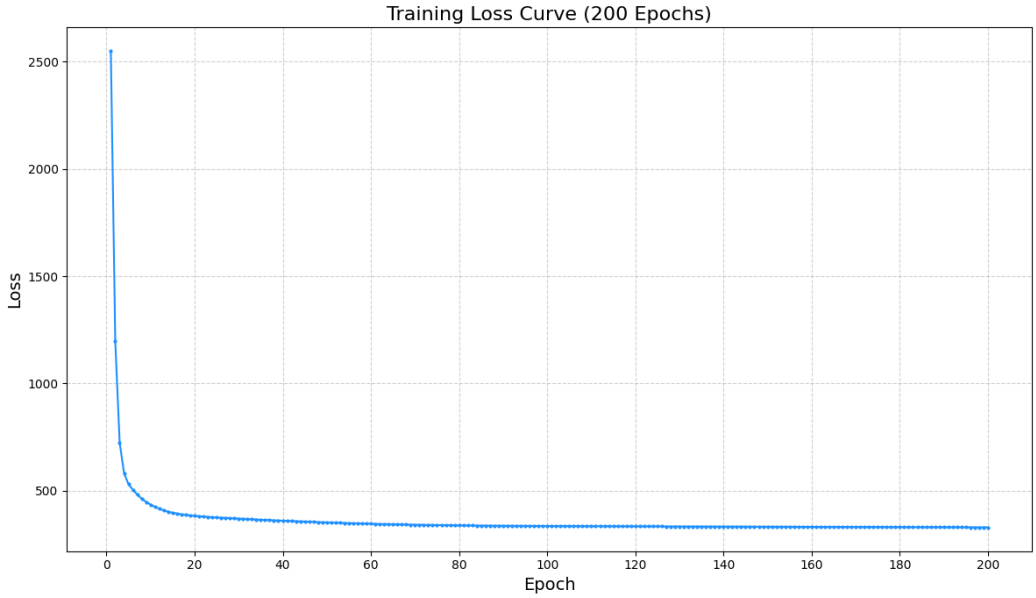

Figure 1. Training Loss Curve of the TransE Model on the FB15k-237 Dataset (Photo/Picture credit: Original).

Following the training of the TransE model for 200 epochs, selected entity embeddings were visualized using t-SNE to reduce their dimensionality to 2D. The entities were colored according to their semantic categories, including individuals, cinematic works, literary subjects, award recipients, and organizational bodies. The resulting t-SNE plot, which is depicted in the supplementary materials, revealed several key characteristics. Entities belonging to the same semantic category formed noticeable clusters in the 2D space. For instance, a green cluster, representing one entity type (individuals), was clearly separated from others, indicating these entities possessed distinct semantic features that the model successfully captured. Furthermore, different clusters, representing distinct semantic categories, generally showed separation, indicating that the model learned to distinguish between these types. However, some clusters exhibited partial overlap. For example, red and purple clusters, representing categories such as cinematic works and organizational bodies, were observed to be clearly aggregated but with some intermingling at their boundaries. This indicates semantic relationships or shared characteristics between these entity types that the model reflected in the embedding space. Additionally, varied cluster cohesion was observed. Some clusters appeared more tightly packed and distinct, while others, such as blue and yellow clusters in one description (literary subjects and award recipients), were more loosely distributed and showed more mixing. This implies that the model found it more challenging to create highly discriminative embeddings for these particular types, or that these types are inherently more diverse or have more ambiguous boundaries in the dataset (Figure 2) [10].

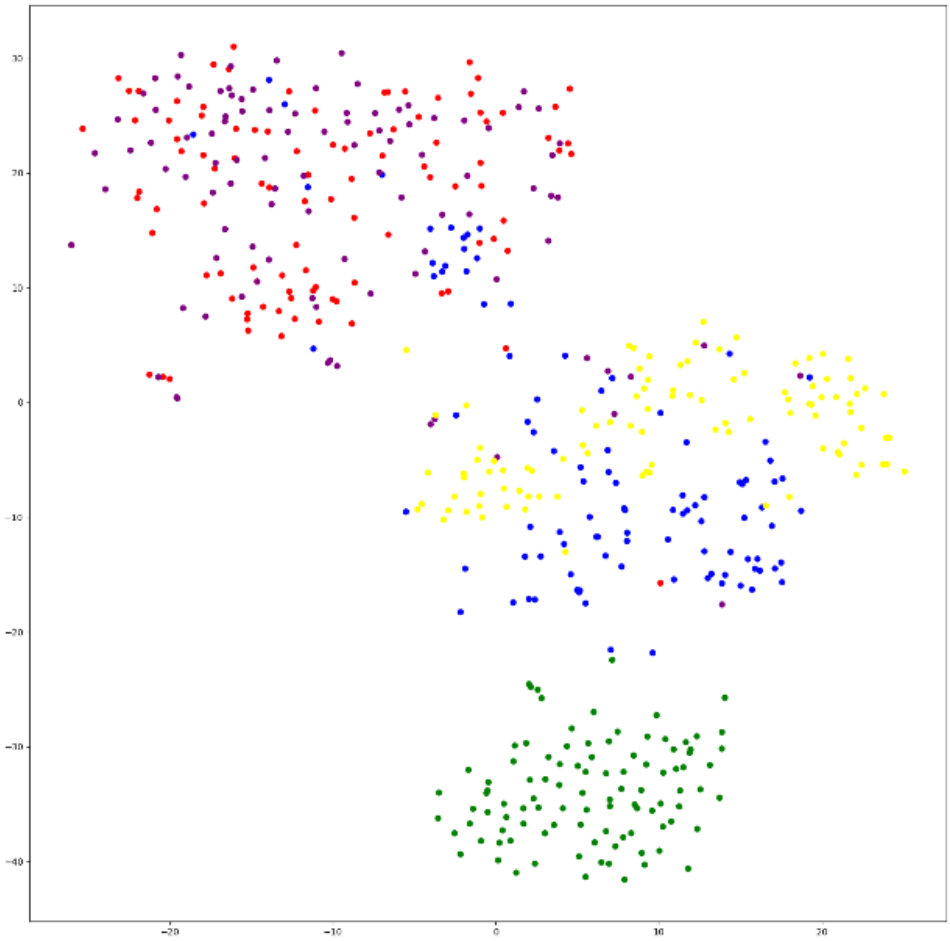

Figure 2. T-SNE Visualization of TransE Model Entity Embeddings (Photo/Picture credit: Original).

4. Discussion

Overall, The TransE model is capable of learning useful semantic representations from knowledge graphs, as demonstrated by the experimental results, leading to effective clustering of entities by type. The initial rapid decrease in loss, followed by gradual convergence, is characteristic of gradient-based optimization methods. The substantial reduction in final stabilized loss compared to the initial state confirms that the learned entity and relation embeddings are significantly better at satisfying the $h + r \approx t$ constraint for positive triples than for negative ones.

However, the TransE model exhibits certain limitations. Its simplicity, relying on a translational assumption, can lead to difficulties in capturing the full nuances of complex knowledge graphs, particularly those involving intricate relational patterns or entities participating in a wide variety of relationships. Specifically, TransE has known limitations in effectively modeling one-to-many, many-to-one, and many-to-many relationships. The observed diffuse and mixed clusters, such as the blue and yellow groupings, further underscore these limitations, suggesting that the model may struggle to create highly discriminative embeddings for inherently diverse semantic categories or those with ambiguous boundaries within the dataset. The exploration of more advanced KGE models, such as TransH or TransR, could be addressed in future research, which introduce more complex mechanisms like relation-specific hyperplanes or distinct relation spaces to better handle these intricate relational patterns and potentially yield more cohesive and separated clusters for all entity types [3].

5. Conclusion

This paper explored Natural Language Processing technologies through an empirical study of the TransE knowledge graph embedding model. The results demonstrated that TransE effectively minimized the training loss over 200 epochs, indicating successful learning of vector representations for entities and relations. The t-SNE visualizations further confirmed this by showing distinct clustering of entities based on their semantic types, although some types exhibited more diffuse or overlapping clusters, hinting at the model's limitations with more complex semantic distinctions. Future research could involve a broader comparative analysis of KGE models, including more recent advancements. Additionally, exploring different hyperparameter settings and their impact on embedding quality, and applying these learned embeddings to specific downstream NLP tasks like question answering or textual entailment, would be valuable extensions.

Knowledge Graph Embeddings are currently applied in areas such as semantic search, recommendation systems, drug discovery, and automatic knowledge base construction. The future of this research domain promises more sophisticated models capable of handling dynamic, multimodal, and text-rich knowledge graphs, further bridging the gap between structured knowledge and natural language understanding.

References

[1] J. O'Connor and I. McDermott, NLP, Thorsons, London, UK, 2001.

[2] M.V. Koroteev, BERT: A review of applications in natural language processing and understanding, arXiv preprint arXiv:2103.11943 (2021).

[3] S.M. Asmara, N.A. Sahabudin, N.S.N. Ismail and I.A.A. Sabri, A review of knowledge graph embedding methods of TransE, TransH and TransR for missing links, in: Proc. 2023 IEEE 8th Int. Conf. Softw. Eng. Comput. Syst. (ICSECS), IEEE, 2023, pp. 470–475.

[4] D. Zheng, X. Song, C. Ma, Z. Tan, Z. Ye, J. Dong, et al., DGL-KE: Training knowledge graph embeddings at scale, in: Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2020, pp. 739–748.

[5] M. Iferroudjene, V. Charpenay and A. Zimmermann, FB15k-CVT: A challenging dataset for knowledge graph embedding models, in: Proc. 17th Int. Workshop Neural-Symbolic Learn. Reason. (NeSy), 2023, pp. 381–394.

[6] X. Chen and Z. Liu, K-nearest neighbor query based on improved Kd-tree construction algorithm, J. Guangdong Univ. Technol. 31 (2014) 119–123.

[7] C.C. Aggarwal, Re-designing distance functions and distance-based applications for high-dimensional data, ACM SIGMOD Rec. 30 (2001) 13–18.

[8] L. van der Maaten and G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 11.

[9] B. Dong, A. Chen and M. Zhang, The role of machine learning in solving the overfitting problem, Psychol. Sci. 44 (2021) 274.

[10] X. Fu, M. Lü, W. Liu and X. Wei, Structured deep discriminative embedding coding network for image clustering, Laser Optoelectron. Prog. 58 (2021) 610016.