

Knowledge Graph Embedding for Link Prediction Models

Ehtisham Ur Rehman^{1,2}, Aamir Saeed^{1,2}, Nasru Minallah², Abdul Hafeez²

¹Department of Computer Science and IT, University of Engineering and Technology, Peshawar, Pakistan,

²National Center for Big Data and Cloud Computing, University of Engineering and Technology, Peshawar, Pakistan

ehtishamrehman@uetpeshawar.edu.pk, asaheed@uetpeshawar.edu.pk, n.minallah@uetpeshawar.edu.pk,
abdul.hafeez@uetpeshawar.edu.pk

Abstract: For disciplines like biological science, security, and the medical field, link prediction is a popular research area. To demonstrate the link prediction many methods have been proposed. Some of them that have been demonstrated through this review paper are TransE, Complex, DistMult, and DensE models. Each model defines link prediction with different perceptions. We argue that the practical performance potential of these methods, having similar parameter values, using the fine-tuning technique to evaluate their reliability and reproducibility of results. We describe those methods and experiments; provide theoretical proofs and experimental examples, demonstrating how current link prediction methods work in such settings. We use the standard evaluation metrics for testing the model's ability.

Keywords: Knowledge Graphs, Link Prediction, Semantic-Based Models, Translation Based Embedded Models.

I. INTRODUCTION

In recent years, knowledge graphs have gotten great coverage by presenting the large complex type of data into entities and relations. Many data scientists have used different knowledge bases such as Deep-dive [1], DBpedia [2] and each knowledge base has enough information to boost the desired algorithm process. Google's search engine algorithm makes use of the idea of knowledge graphs [3]. Millions of items and relations are represented in the knowledge graph. The WN18 [4] and FB15K [5] knowledge graph datasets, for instance, are made up of millions of information bits of triplets. Entities and relations are contained within each triplet. A large multi-relational data graph like this has a lot of potential for enhancing a variety of activities, from information extraction and recommendation systems to data mining [6].

Previously, modelling relations via rotating operators such as RotatE [7] appeared as a viable strategy in knowledge graph embedding. In translational knowledge-based techniques, this paradigm incorporates relational rotation. Relationships are viewed as rotation operations of entities in a 2D complicated space. Furthermore, the rotation axis was constrained to be orthogonal to the entity embeddings, removing the possibility of interaction between relation and entity embeddings [8].

By looking at the core models Nickel et al. [9], have capitalized much determination in relational learning methods such as Tensor Factorization [10] and Translation-based embedded model by Bordes et.al [11]. These are the two most common techniques of converting data into low-dimensional representations of exploring the relationships. In terms of validating unseen data given the current knowledge base, these graph embedding approaches have demonstrated reasoning and scalability ability.

In this review, we concentrate on the learning of embedding models, where models are trained using neural networks. Existing graph representation embedding models NTN (Socher et al. [12]) and TransE (Bordes et al. [11]) have shown impressive prediction performance when compared to tensor factorization approaches like RESCAL (Nickel et.al. [9]). Through this review paper, we compare the TransE [11], Complex [13], DistMult [14], and DensE [15] algorithms to learn embeddings for link prediction. To do this we evaluate these models with two different metrics such as MRR (raw and filtered) and Hits ratio measured on WN18 [4] and FB15K [5] datasets.

The rest of this review is organized as follows. Section II includes a review of the associated literature as well as the underlying information needed to undertake the study. Section III discusses how to use embeddings. Sections IV and V describe the contribution as well as the datasets utilized to train the models. Section VII examines the assessment and results gained by employing various statistical analysis methodologies based on their performance matrices. Finally, Section VIII ends the study work using the collected results while Section IX exhibits the future research work opportunities in this subject.

II. RELATED WORK

For the past few years, researchers have been focusing on multi-relational data learning. Richardson and Domingo's proposed conventional learning techniques such as Markov-logic networks [15] which suffer from scalability issues. For multi-relational data, it is challenging to adopt these algorithms as they have complexity and limitations in their models, which may lead the system into unseemly conditions. Matrix factorization by Nickel et al. [9] and Translation-based embedding networks by Bordes et al. [11] are two other strategies for transforming multi-relational data into low-dimensional representations of entities and relations. This research review focuses on neural embedding models, which have proven to be extremely scalable on a large-scale knowledge base.

The present neural embedding models TransE (Energy-based model) and NTN (Neural Tensor Network) show entities in low-dimensional vector space and represent relations as a set of types between entities [11,12]. Both of these models have different parametrization of relation operators. For illustration, assumed two entity vectors, the TransE represents each relation as a single vector i.e. one to one relation. TransE an energy-based model was the baseline model introduced in the area of Translation based embedded models, which was trained on FB15K [5] and WN18 [4] datasets. TransE gives the idea of negative sampling which was similar to the Skip-gram Model. It gives the idea of Head, which represents H, Relation represents R, and T represents the Tail. While in Skip-gram neural model, we have Context, Word, and Target. In the TransE model, we labelled the Tail with 1 which is positive while 0 is negative. It uses the Deep walk approach to generate embeddings [6,11].

The problem with the TransE model is that it does not contract with multi-relation nodes. It only deals with one-to-one relations. For multi-relational nodes, other approaches such as TransH (Hyper-plane) [16] and TransR (Multi-relational) [17] in Translation-based embedded models were used to generate embeddings for multi-relation nodes. However, implementing these novel approaches on multi-relational nodes is quite challenging as they have limitations and complexity in their algorithms. On the other hand, NTN (Socher et.al. [12]) represents each relation as a bilinear tensor followed by a linear matrix operator. Most methods present each entity as a single vector while the NTN algorithm represents entities as an average of word vectors while also initializing word vectors with pre-trained vectors from outside text corpora.

This study represents the embedding-based extraction described in the section of this publication and is connected to the previous work on logical inference with learned continuous-space representation. Bowman [18] illustrates how a hierarchical neural network may preserve the aspects of ordinary logical reasoning on situations that use iterators a few and all. Rossi et al. [19] investigate the various metal object embeddings for link prediction, which allowed them to increase the scale. They compare the efficiency and efficacy of 16 cutting-edge approaches. H.Li et al. [20] employ the sequential recommendation for usage by forecasting the behaviours that supported their previous behaviour. They proposed CTransRec in the domain of Translation-based embedding models that use category-specific projections and temporal dynamic relaxation to accomplish this.

III. EMBEDDINGS

Embeddings are the structure extracted from the data inform of entities supported by the information attributes and their relations with surrounding entities. These embeddings are used for mapping data such as images, documents, or a graph into a static length of matrix or vector. This vector captures the main key features from the data by reducing the dimensions [21].

Area embeddings are an essential technique in deep learning for enhancing the value of dense representations. It offers a decent target domain for all types of stuff, including, images, phrases, keywords, and so on. The target domain of intellectual vectors of real numbers is inconceivable. However, this is irrelevant because such vectors have the excellent working characteristic of allowing objects to advance. That is, vectors inside the target domain are mapped to vectors with equivalent vector distance metrics [22].

1 Graph Embeddings

Graph embeddings introduce the main structure of deep learning algorithms and strategies for calculating the useful information from the data to utilize and evaluate the different matrixes. These embeddings represent graphs data into a fixed-length of vector space. The whole graph is encoded into a single vector. In this review we literate the TransE [11], Complex [13], DistMult [14], and DensE [23] algorithms, which computes the embeddings for entities and relations types in knowledge graphs. Moreover, these algorithms were used for link prediction in a graph.

2 Link Prediction

Link prediction contains numerous algorithms on graphs. This Sub-Section contains the descriptive view of link prediction in the area of graph embeddings. A link prediction algorithm predicts features of nodes (relation

between nodes) in a graph, for instance, predicting a link whether the link in the graph exists [11,24]. It is used for recommendation, binary classification, or knowledge base completion; it is also used in a knowledge graph or even labelling existing links with a categorical class (binary or multiclass classification) or even predicting a continuous number (regression).

Link prediction is supervised or semi-supervised learning, where the model is trained using a subset of links/edges that have ground-truth labels. For predicting edge existence, the ground truth may just be whether the edge exists in the original data, rather than a separate label [25].

Link prediction can be applied on downstream tasks from node representation embeddings; it can be done by combining the node embedding vectors for the source and target nodes of the edge and training a supervised or semi-supervised classifier against the result. For unsupervised learning algorithms, it can be used as a random walk approach.

A TransE (Energy-based Model)

The concept of translation-based embedded models in the area of knowledge graph (within the domain of deep learning) has gotten incredible reflection; rightfully so because of its better performance using the deep walk approach. This model gave the idea of negative sampling in link prediction similar to the skip-gram neural model. TransE an energy-based model is the baseline approach, which uses Head, Tail, and Relation called a triplet. Using the TransE model each Head or Tail is labeled with 0 value which represents negative, while 1 represents positive triplet. The TransE model used a deep neural walk to perform walks among nodes randomly [11].

This model examines the problem of embedding entities and relationships in multi-relational data in low-dimensional vector space. The main objective is to develop an algorithm that can easily train on a very large dataset while having fewer parameters. The TransE algorithm represents relationships by interpreting them as translations on the entities' low-dimensional embeddings.

The scoring function of the TransE model uses the L1 and L2 norms to compute the similarity between the embeddings of the top H transformed by the embedding of the relation R and therefore the embedding of the tail T.

Besides all these embeddings, the TransE model only deals with one-to-one relations; it is not suitable for multi-relation such as one to many, many to many, etc. For instance, we have a sample of (Los- Angeles, location, USA) and (Hollywood Boulevard, location, USA). After training, the 'Los- Angeles' entity vector will be close with the 'Hollywood' entity vector. But in reality, they don't have such similarities [11,26].

B Complex Approach

In statistical relational learning, the problem of link prediction is the main key to understanding the structure of rich knowledge bases. In an earlier approach, latent factorization was used to solve the problem. The use of complex-valued embeddings, on the other hand, enables the creation of complex embeddings that can handle a wide range of binary relations, including (symmetric and antisymmetric relations) [13].

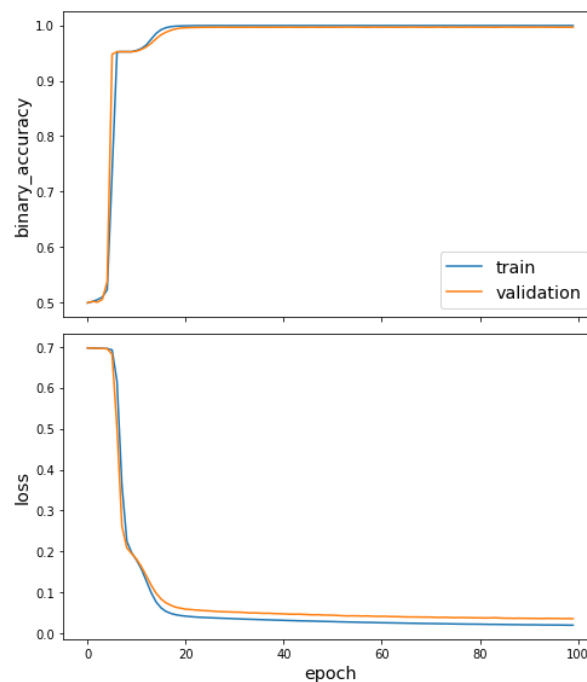


Fig. 1. Complex Model using WN18 Dataset

The Hermitian dot product is used in the innovative technique based on complicated embeddings. The method is scalable to huge datasets since it is linear in both space and time, and it regularly outperforms other algorithms on typical link prediction benchmarks.

C DistMult (Variational Inference)

Yang et al. [14], the original baseline technique having the same embeddings for entities in the Head and Tail positions, resulting in more training data per entity embedding while confining the model to symmetric interactions. Variational inference is the most recent study in this area. An effective method for computing the Bayesian posterior over latent variables based on observations. The study on embedding-based rule extraction is connected to current work on logical inference using trained continuous representations.

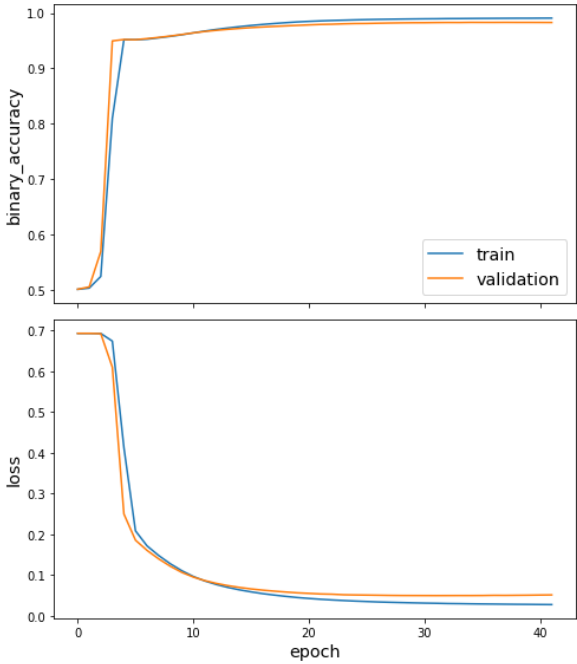


Fig. 2. Complex Model using FB15K Dataset

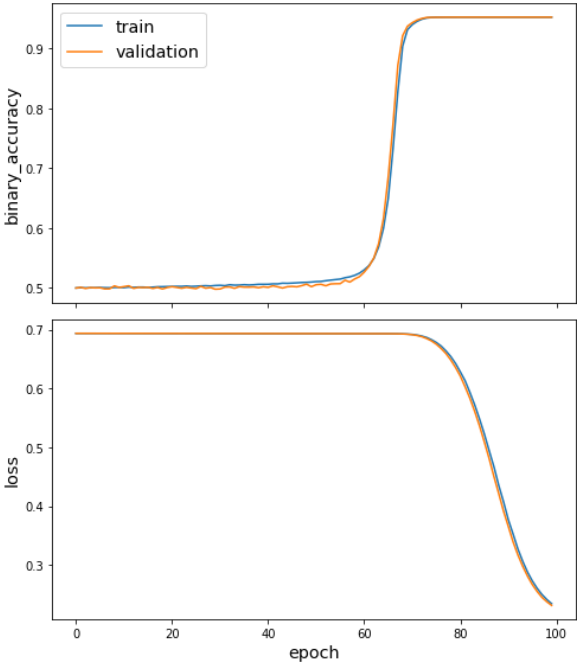


Fig. 3. DistMult Model using WN18 Dataset

Much of the current research in this area is focused on developing logic-based representations for natural language phrases. Prediction knowledge graph embeddings are the most effective approaches for completing an incomplete set of relational information for links. The issue with those models that have a high sensitivity to hyper-parameters, particularly regularizes, which must be extensively modified to get good performance Kadlec et al. [27]. Furthermore, Bamler et al. [28] recently provided an effective technique for large-scale hyper-parameter tweaking in the DistMult model by interpreting these models in a probabilistic context. After introducing pre-entity hyper-parameters into the model, it employs a variational expectation maximization strategy to adjust a large number of such hyper-parameters with minimal effort.

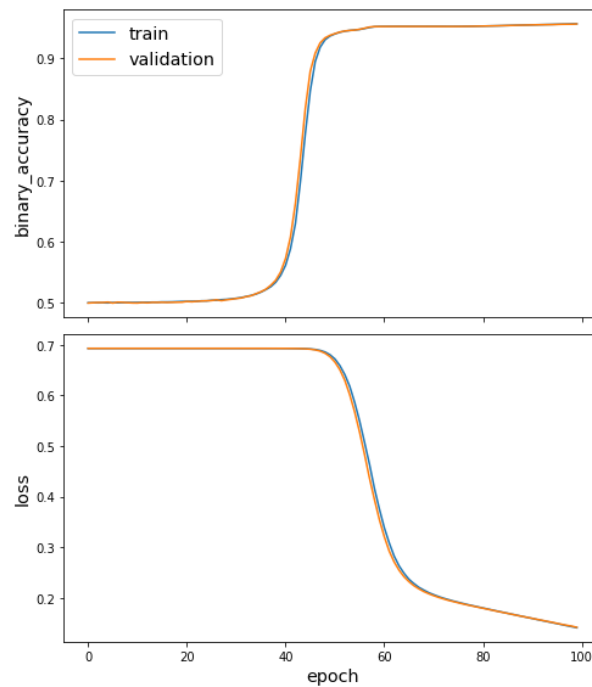


Fig. 4. DistMult Model using FB15K Dataset

D DensE

In the knowledge graph area, the arrangement of relations is a vital task. It is also an important step toward multi-hop reasoning overlearned information. Previous techniques in Translation-based [26] and Rotation-based models [7] such as TransR [17], RotatE [7], have employed the product of a series of complex-valued diagonal matrices to describe the numerous relations. However, these models were challenging to adopt for having complexity and limitation, which lead the system to unseemly conditions.

The DensE [23] model gives enough and can deal with the challenge of composite relations having symmetric, independent of entities, and fixed in scale. DensE employs two approaches. 1). The associated diagonal relation matrices for composite relations can be non-commutative and connected to entity embeddings. 2). It preserves the direct geometrical interpretations, revealing how the relations with diverse patterns are depicted and expands the RotatE concept into a more expressive environment with minimal model complexity.

The link prediction problem has been a long-standing topic in the contemporary data science sector, and the AI community has presented a plethora of methods based on Markov chains and statistical models. In any case, their work has not found the current advancement of complex network investigations; they, in particular, require genuine consideration of the primary qualities of networks, similar to the various organizational and community structure, which may indeed provide useful information and insights for link prediction. Physical procedures, such as random walk techniques and maximum likelihood algorithms, have lately found uses in link prediction. For evaluation, we used FB15K [5] and WN18 [4] datasets for link prediction.

TABLE I. DEFINED PARAMETER MODELS

Datasets	WN18 FB15K	WN18 FB15K	WN18 FB15K	WN18 FB15K
Algorithms	TransE	Complex	DistMult	DensE
epochs	100	100	100	100
Dimensions	200	200	200	200

IV. CONTRIBUTION

To achieve the desired performance, through this review, we compare different graphs embedding models both in translation and semantic-based such as TransE [11], Complex [13], DistMult [14], and DensE [23] with the contribution of adjusting the values of the parameters for a certain observation. The fine-tuning process significantly decreases the time required for processing new graph embedding algorithms as it already contains useful information from pre-existing graph embedding algorithms.

The concept of fine-tuning proves valuable in training new graph embedding algorithms; however, it can be used only when the dataset of existing models and the new graph embedding algorithms are similar to each other [29]. In our case, the datasets that we used for modelling and pre-processing are WN18 [4] and FB15K [5]. We take the graph embedding models that have already been trained for the link prediction.

The process of fine-tuning in those models having similar parameter values provides ease of transferring knowledge. The available data from previous graph embedding models can easily be imported for the new models with slight modifications in the parameter values [29].

This contribution proves useful in training models as shown in Section VII. A huge amount of data is imported from previous models. Using fine-tuning the models were evaluated more concisely and more succinctly.

1 Fine Tuning of Parameters Values

For the deep learning process, the parameters of a model must be adjusted very precisely to fit certain observations. That is why we set up our model parameters such as the number of epochs for training and the total number of dimensions of embedding vectors which we compute for every individual node and each edge type [29,30].

The evaluation is achieved in three main steps,

- Load the data
- Train both models for WN18 and FB15K
- Evaluate both models.

For evaluation, we used two different optimization techniques such as Adam optimizer [31] and AdaGrad optimizer [32]. These are the methods used to change the attributes such as weights and learning rate to reduce the losses of the neural network. Optimizer methods are used to solve the optimization problems by minimizing the function. In this review paper, for the evaluation of the Complex [13] and DistMult [14] algorithm, we used both optimizer's algorithms [33]. For that, we used the same dimensions and epoch values for Adam optimizer and AdaGrad optimizer with parameters value of 100 epochs and with an embedding dimension of 200. The corrupted edges are 20. For the implementation of the DensE [23] model, we use the Adam optimizer and tune the hyper-parameter according to other defined models having similar learning values as shown in TABLE I. Through this paper, we compared the DensE with top-performing baseline models for knowledge graph link prediction; those top-performing baseline models include TransE [11], Complex [13], and DistMult [14]. These baseline approaches contain both translational-based and semantic models.

2 Adam Optimizer

It is a combination of the RMSProp and AdaGrad algorithms that provides an optimization approach for dealing with sparse gradients on noisy issues. Adam is a stochastic gradient descent alternative technique for training deep learning models. It, like RMSProp, employs squared gradients to learn the learning rate and takes advantage of momentum by utilizing the moving average of the gradient rather than the gradient itself, like SGD with momentum does [31,33].

3 AdaGrad Optimizer

Adaptive Gradient [32] Algorithm is an algorithm used for gradient-based optimization. It is used to perform smaller updates. It is mainly used with sparse data i.e. image recognition or NLP [34]. It improves the performance of problems with sparse gradients as each parameter has its learning rate.

4 Entity Representation

Many data scientists have utilized entity representation as a knowledge base such as WN18 [4] and FB15K [5]. It is used to evaluate and represent the high-dimensional data into a single vector space. These knowledge datasets contain a set of information in the form of entities and relations and ontology defines as $Z_G = (E, R, O)$.

The representation of the input entity relates to a high dimensional vector, it may be an n-hot feature vector or a one-hot index vector. Represents by X_{e1} and X_{e2} the input vectors for entity $e1$ and $e2$, respectively. Denotes by W the first layer projection matrix. The learned entity representations, y_{e1} and y_{e2} that can be represented as:

$$y_{e1} = F(W_{e1}), y_{e2} = F(W_{e2}) \quad (1)$$

F can be either a linear or a nonlinear function, and the parameter matrix W can be randomly initialized using pre-trained vectors. Most well-known embedding models, like the one-hot vector representation technique, have been presented as an alternative to Socher et al. (2013), NTN model, which defines each entity as an average of its word vectors. This may be achieved by taking an input containing the bag of word vectors and learning a projection matrix composed of word vectors [12].

V. DATASETS

Dataset is a collection of facts and figures which can be used for analysis, storage, and management. Dataset is an important tool for scientific progress and advancement of knowledge. Through this research, to train the Complex and DistMult algorithm, we have used FB15K and WN18 datasets. Each dataset is used to set up parameters for both models. We compute the number of epochs for training, and the dimensions of the embedding vectors for each node and every individual edge type [35].

1 WN18 (WordNet)

Knowledge base relation triplets are stored in the lexical database. The Cognitive Science Laboratory developed it in the mid-1980s. The database's most recent version was released in June 2011. These datasets are utilized to obtain results from a specific approach's entities and relations. It is made up of 40943 entities with 18 relationships. The training set has 141442 triplets, the validation set has 5000 triplets, and the test set has 5000 triplets [4].

2 FB15K (FREEBASE)

It is a collaborative knowledge repository created in March 2007 by Metaweb, an American software business. We train the defined models on FB15K with a learning rate of 0.01, to predict the link based on negative and positive sampling, to extract the approach of both algorithms. FB15K is a subset of Freebase, which is a curated knowledge base of general facts. It is a vast and ever-expanding information base made up of triplets. There are 1345 connections between the 14951 nodes. The training set has 483142 triplets, the validation set has 50000 triplets, and the test set has 59071 triplets [5].

VI. TRAINING A MODEL

This review paper uses the WN18 and FB15K dataset to train both models. This dataset comes with a defined Train, Valid, and Test. Each set consists of Head H, Relation R, and Tail T, representing ontologies or Triplets. Using the load method of the WN18 and FB15K datasets, we load a StellarGraph python library with all of the triplets as well as the individual splits as Pandas data frames.

The TransE model translates embeddings for modelling multi-relational data. To train the model, bring the sum of the Head vector and the Relation vector as near to the Tail vector as feasible. It measures using the L1 and L2 norms. With negative sampling, the loss function is the max-margin [11].

$$L(y, y') = \max(0, \text{margin} - y + y') \quad (2)$$

Where y represents the positive triplets and y' is the negative triplets. TransE uses the distance function to represent the score. For that it adds a minus to the equation, the loss function for knowledge representation is:

$$L(h, r, t) = \max(0, d_{pos} - d_{neg} + \text{margin}) \\ \text{and } d \text{ represents the } d = ||h + r - t|| \quad (3)$$

This is the L1 and L2 norm. To get the negative triplets it replaces the Head or Tail with 0.

Looking at other models in the research the Complex [13] and DistMult [14] both algorithms contain some embedding layers and a scoring layer; however, the DistMult object means these specifics are invisible to us. Both of these models take "Knowledge graph triplets" which can be created in a suitable format using *KGTripleGenerator*.

For training, the input is produced by calling the *KGTripleGenerator* flow method. In this method, it takes a data frame with Source, Label, and Target column, where every individual row uses a true edge in the knowledge graph. The Negative samples parameter controls the number of edges that are randomly created for every positive edge to use as negative examples for training.

The DensE neural model has a network layer that is densely linked. Each neuron in the thick layer receives information from all neurons in the layer before it. The thick layer was discovered to be the most often employed layer in the model.

The dense layer conducts matrix-vector multiplication during training. The values in the matrix are parameters

that may be learned and modified via backpropagation. The dense layer produces an M-dimensional vector as output. The Dense layer is used to modify the vector's dimensions [23].

VII. EVALUATION OF MODEL

To ensure the quality, the evaluation of a model, which is based on its proper outcomes and effects, is an important part of any research. It clearly defines the model design, specification, and code. This Section evaluates the defined link prediction algorithms such as TransE [11], Complex [13], DistMult [14], and Dense [23] based on WN18 and FB15K datasets.

FB15K is a subset of Freebase, a knowledge base of generic facts that have been curated [5]. While the WN18 dataset is a subset of WordNet, a database that contains lexical relationships between words. Positive triplets may be found in both datasets [4]. For TransE (Energy-based model), we employ the original training, validation, and test set splits provided by Border et al. 2013 [11]. They construct negative entities by changing the Head or Tail of a triplet at random to make negative samples. For each batch of training positive instances, this negative sampling is performed during runtime.

To inspect the desired algorithms, a close analysis has been done to search the entities relations; the projected algorithm was evaluated based on MRR and the top n ratio performance. We measure the quality of the ranking of each test among all possible Head and Tail such as (H', R, T) or (H, R, T'), $\forall H', \forall T' \in \mathcal{E}$. MRR (Mean Reciprocal Rank) and Hits ratio are the two common assessment methods for these datasets, which are divided into two sections, Raw and Filtered, as initially described in Bordes et al 2013 publication. The filtered measures are derived by excluding from the ranking any positively observed triplets that exist in either training, validation, or test, but the raw metrics do not [11,17].

For evaluation of model performance, we analyzed raw and filtered MRR and Hits @ 1, 3, and 10 as shown below in Sub-Section A and B.

A. MRR (Mean Reciprocal Rank)

MRR is a probability metric used to evaluate the performance of the algorithm based on a sample of queries arranged by the probability of accuracy.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (4)$$

Where $rank_i$ indicates the rank position of the first relevant script for the i th query. For evaluating the defined models such as TransE, Complex, DistMult, and Dense, it is used to predict the measure of both algorithm performances.

B. Hit@ratio

The hit ratio is a metric and measure of the model used to evaluate the performance traditionally associated with their results. In the research, we evaluate TransE [11], Complex [13], DistMult [14], and Dense [23] algorithm by predicting the top n link in a graph.

To do this, the technique computes proximity C for each pair of nodes (x,y). In this case, C can stand for graph distance, Adamic/Adar, common neighbour's, or Jaccard's coefficient. In this study, C stands for the number of common neighbour's $C_{(x,y)}$ of x and y. Sort pairings in decreasing order of score [32].

$$E * new := E_{new(Core*core)} \quad (5)$$

Predict edges only when both ends are in the core. Predict the top n pairings of links as new connections.

TABLE II. DIFFERENT GRAPH EMBEDDING MODELS

Graph distance	Distance between x and y coordinates
Common neighbours	$ \Gamma(x) \cap \Gamma(y) $
Adamic/Adar	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log \Gamma(z) }$
Jaccard's coefficient	$\frac{ \Gamma(x) \cap \Gamma(y) }{ \Gamma(x) \cup \Gamma(y) }$

Preferential attachment	$ \Gamma(x) \cdot \Gamma(y) $
-------------------------	---------------------------------

VIII. RESULTS

The WN18 [4] knowledge base comprises antisymmetric relations with characterizing the lexical and semantic hierarchies between concepts. Using the WN18 knowledge base as an input, with similar parameter variables on desired algorithms such as TransE [11], Complex [13], DistMult [14], and DensE, shows different results based on MRR and Hit ratio. Indeed, the DensE outperformed other models, which are on par with respective raw MRR scores of 0.950. On the other, side Complex, secure the highest filtered MRR score with 0.927 as shown in TABLE IV.

To visualize the result, in Fig. 6 with a vertical bar chart of individual measures of each model. The bar chart represents the most relevant measure and frequency of each model. According to the chart, the Complex model shows a high measuring rate on raw MRR and DensE on filtered MRR.

TABLE III. MODELS PERFORMANCE ON WN18 DATASET (RAW AND FILTERED). WE REPRODUCED THE PERFORMANCE METRICS WITH THREE DECIMALS BY USING THE CODE PROVIDED BY THE AUTHORS.

Models	MRR (Raw)	MRR (Filtered)
TransE	0.335	0.454
Complex	0.601	0.927
DistMult	NAN	0.00017
DensE	0.950	NAN

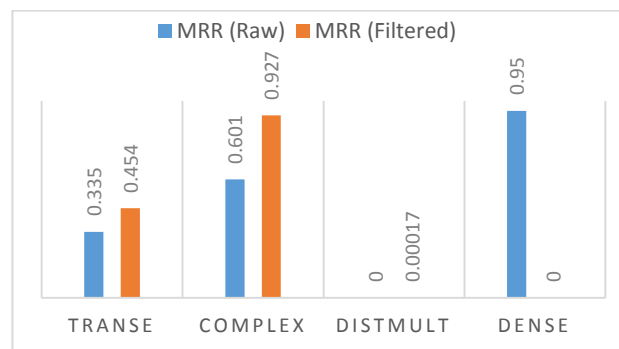


Fig. 5. Visualizing the model performance using WN18 dataset for Hit @ratio

For more evaluation, hit ratio metrics were used to evaluate the performance of models with the number of times that are correct prediction was made in ration to the number of total prediction links. TABLE IV shows the DensE model predicts a high rating on raw data with top 1, 3, and 10 links in a graph, while Complex performs well prediction on filtered data.

TABLE IV. MODELS PERFORMANCE ON HIT@RATIO USING WN18 DATASET

Models	Hits@1 (raw)	Hits@1 (raw)	Hits@1 (raw)
TransE	0.049	0.080	0.125
Complex	0.482	0.615	0.820
DistMult	NAN	NAN	0.924
DensE	0.945	0.954	0.959
Models	Hits@1 (filtered)	Hits@1 (filtered)	Hits@1 (filtered)
TransE	0.089	0.823	0.934
Complex	0.908	0.939	0.953
DistMult	NAN	NAN	0.0001
DensE	NAN	NAN	NAN

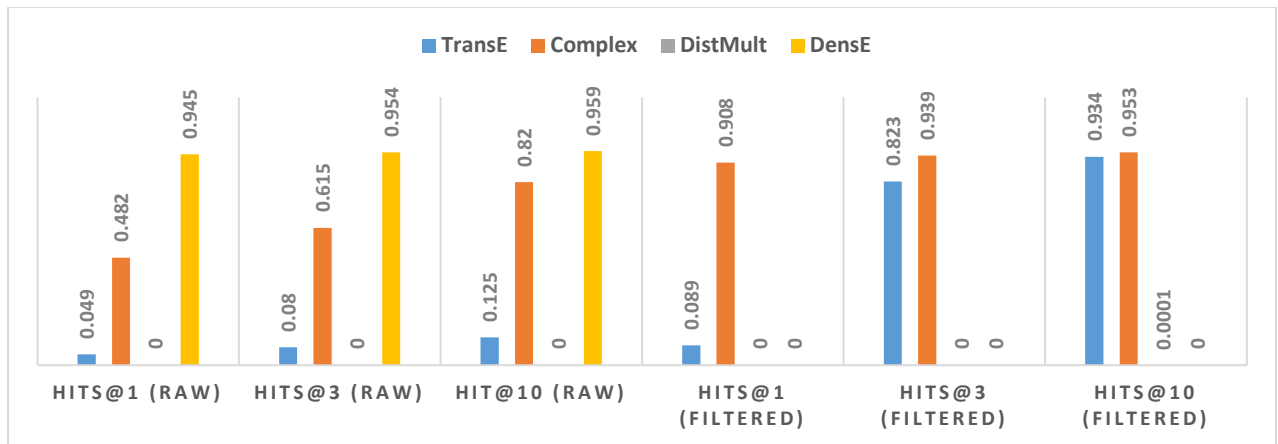


Fig. 6. Visualizing the models performance on WN18 Dataset (Raw and Filtered)

In this part of the result section. TABLE V shows the performance results based on the FB15K knowledge dataset. According to results, the TransE, an energy-based model, shows high MRR both on raw and filtered data.

Both entities and relations are represented as vectors in the same space by TransE. Given a fact (h,r,t), the relation is acknowledged as a translation vector 'r' such that the embedded entities h and t may be linked by r with minimal errors [11].

TABLE V. MODELS PERFORMANCE ON FB15K DATASET (RAW AND FILTERED). WE REPRODUCED THE PERFORMANCE METRICS WITH THREE DECIMALS BY USING THE CODE PROVIDED BY THE AUTHORS

Models	MRR (Raw)	MRR (Filtered)
TransE	<u>0.4303</u>	<u>0.9627</u>
Complex	0.2566	0.6144
DistMult	NAN	0.0677
DensE	0.349	0.8167

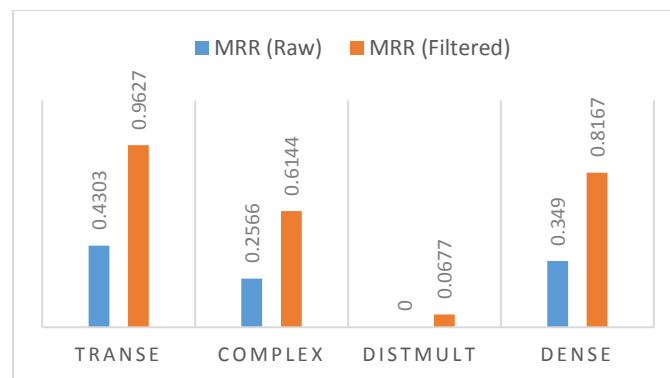


Fig. 7. Visualizing the models performance using FB15K dataset for (Raw and Filtered)

We reproduced the performance metrics with three decimals by using the code provided by the authors. Best results are labelled in bold and underlined.

TABLE VI. PERFORMANCE COMPARISON ON BENCHMARK DATASET I.E. FB15K. BEST RESULTS ARE LABELED IN BOLD AND UNDERLINED.

Models	Hits@1 (raw)	Hits@1 (raw)	Hits@1 (raw)
TransE	0.219	0.376	0.475
Complex	0.1397	0.226	<u>0.514</u>
DistMult	NAN	NAN	0.321
DensE	<u>0.535</u>	<u>0.384</u>	0.256
Models	Hits@1 (filtered)	Hits@1 (filtered)	Hits@1 (filtered)
TransE	NAN	0.249	0.349

Complex	<u>0.489</u>	<u>0.642</u>	<u>0.829</u>
DistMult	NAN	NAN	0.132
DensE	NAN	NAN	NAN

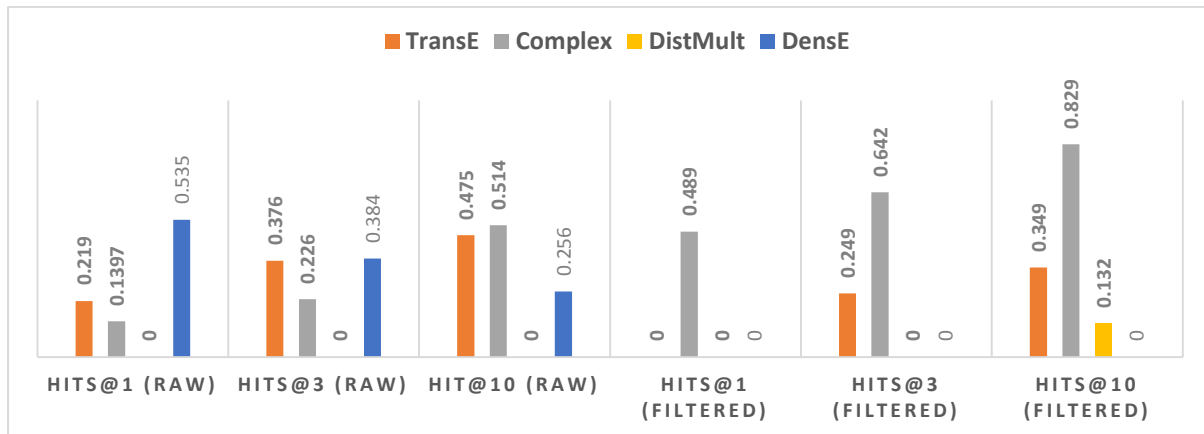


Fig. 8. Visualizing the model performance on FB15K dataset for Hit @ratio

Experiment results based on Hit ratio metric on two benchmark knowledge graphs datasets, WN18 [4] and FB15K [5], show that the DensE model performs better than the existing state of the art models for missing link prediction, particularly on composite relations on raw data, whereas the Complex model performs well on filtered data to predict missing links. Fig. 8 **Error! Reference source not found.** depicts the experimental results from the FB15K dataset, as indicated in TABLE VI.

IX. CONCLUSION AND FUTURE WORK

We conducted a thorough comparative study of translation-based and semantic-based models based on knowledge graph embedding in this paper. In this study, we study and compare numerous cutting-edge graph-embedding models, such as TransE, Complex, DistMult, and DensE, for link prediction utilizing MRR and Hit@ratio ranking approaches. The experiment is carried out on two well-recognized benchmark datasets, WN18 [4] and FB15K [5].

We conduct a series of ablation experiments using the fine-tuning approach to assess the efficacy of each module in defined models. We fully addressed current assessment processes, indicating that they may be based on a wide range of parameter values. We explored the elements that make models most sensitive to these principles, offering useful insights for future research.

This review paper facilitates the system to predict the links in a knowledge base graph having similar parameter values. While comparing these models and comparing them to alternate state-of-the frameworks, these models can have a comprehensive assessment for future work.

In the future, these approaches will have a general improvement for predicting other areas such as Diagnosis of Cancer patients, Crime control, etc. For technology-based assessment to store large and updated data for model learning, AWS services can be used to improve the facilities of the models such as the problem of the overload on the local server [36].

ACKNOWLEDGEMENT

This review paper is supervised by the National Center in Big Data and Cloud Computing from the University of Engineering and Technology, Peshawar, Pakistan.

REFERENCES

- [1] C. Zhang *et al.*, 'DeepDive: declarative knowledge base construction', *Commun. ACM*, vol. 60, no. 5, pp. 93–102, Apr. 2017, doi: 10.1145/3060586.
- [2] G. Kobilarov and C. Bizer, 'DBpedia - A Linked Data Hub and Data Source for Web and Enterprise Applications', p. 3, Apr. 2009.
- [3] S. Brin and L. Page, 'The anatomy of a large-scale hypertextual Web search engine', *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998, doi: 10.1016/S0169-7552(98)00110-X.
- [4] G. A. Miller, 'WordNet: a lexical database for English', *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: 10.1145/219717.219748.

- [5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, 'Freebase: a collaboratively created graph database for structuring human knowledge', in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, Vancouver, Canada, 2008, p. 1247. doi: 10.1145/1376616.1376746.
- [6] Q. Wang, Z. Mao, B. Wang, and L. Guo, 'Knowledge Graph Embedding: A Survey of Approaches and Applications', *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017, doi: 10.1109/TKDE.2017.2754499.
- [7] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, 'RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space', *arXiv:1902.10197 [cs, stat]*, Feb. 2019, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1902.10197>
- [8] K. Hayashi and M. Shimbo, 'A Non-commutative Bilinear Model for Answering Path Queries in Knowledge Graphs', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 2422–2430. doi: 10.18653/v1/D19-1246.
- [9] M. Nickel, V. Tresp, and H.-P. Kriegel, 'A Three-Way Model for Collective Learning on Multi-Relational Data', p. 8.
- [10] V. Kuleshov, A. T. Chaganty, and P. Liang, 'Tensor Factorization via Matrix Factorization', *arXiv:1501.07320 [cs, stat]*, May 2015, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1501.07320>
- [11] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, 'Translating Embeddings for Modeling Multi-relational Data', p. 9.
- [12] D. Chen, R. Socher, C. D. Manning, and A. Y. Ng, 'Learning New Facts From Knowledge Bases With Neural Tensor Networks and Semantic Word Vectors', *arXiv:1301.3618 [cs]*, Mar. 2013, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1301.3618>
- [13] T. Trouillon, J. Welbl, and S. Riedel, 'Complex Embeddings for Simple Link Prediction', *Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA*, vol. 48, p. 10, 2016.
- [14] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, 'Embedding Entities and Relations for Learning and Inference in Knowledge Bases', *arXiv:1412.6575 [cs]*, Aug. 2015, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1412.6575>
- [15] M. Richardson and P. Domingos, 'Markov logic networks', *Mach Learn*, vol. 62, no. 1–2, pp. 107–136, Feb. 2006, doi: 10.1007/s10994-006-5833-1.
- [16] Z. Wang, J. Zhang, J. Feng, and Z. Chen, 'Knowledge Graph Embedding by Translating on Hyperplanes', p. 8.
- [17] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, 'Learning Entity and Relation Embeddings for Knowledge Graph Completion', p. 7.
- [18] S. R. Bowman, 'Can recursive neural tensor networks learn logical reasoning?', *arXiv:1312.6192 [cs]*, Feb. 2014, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1312.6192>
- [19] A. Rossi, D. Firmani, A. Matinata, P. Merialdo, and D. Barbosa, 'Knowledge Graph Embedding for Link Prediction: A Comparative Analysis', *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, pp. 1–49, Apr. 2021, doi: 10.1145/3424672.
- [20] H. Li, Y. Liu, N. Mamoulis, and D. S. Rosenblum, 'Translation-Based Sequential Recommendation for Complex Users on Sparse Data', *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1639–1651, Aug. 2020, doi: 10.1109/TKDE.2019.2906180.
- [21] I. Balažević, C. Allen, and T. M. Hospedales, 'Hypernetwork Knowledge Graph Embeddings', *arXiv:1808.07018 [cs, stat]*, vol. 11731, pp. 553–565, 2019, doi: 10.1007/978-3-030-30493-5_52.
- [22] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, 'Representing Text for Joint Embedding of Text and Knowledge Bases', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1499–1509. doi: 10.18653/v1/D15-1174.
- [23] H. Lu and H. Hu, 'DensE: An Enhanced Non-Abelian Group Representation for Knowledge Graph Embedding', *arXiv:2008.04548 [cs]*, Aug. 2020, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/2008.04548>
- [24] H. Yuliansyah, Z. A. Othman, and A. A. Bakar, 'Taxonomy of Link Prediction for Social Network Analysis: A Review', *IEEE Access*, vol. 8, pp. 183470–183487, 2020, doi: 10.1109/ACCESS.2020.3029122.
- [25] Z. Peng, H. Song, X. Zheng, and L. Yi, 'Construction of hierarchical knowledge graph based on deep learning', in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, Jun. 2020, pp. 302–308. doi: 10.1109/ICAICA50127.2020.9181920.
- [26] M. He, B. Wang, and X. Du, 'HI2Rec: Exploring Knowledge in Heterogeneous Information for Movie Recommendation', *IEEE Access*, vol. 7, pp. 30276–30284, 2019, doi: 10.1109/ACCESS.2019.2902398.

- [27] R. Kadlec, O. Bajgar, and J. Kleindienst, 'Knowledge Base Completion: Baselines Strike Back', in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, 2017, pp. 69–74. doi: 10.18653/v1/W17-2609.
- [28] R. Bamler, F. Salehi, and S. Mandt, 'Augmenting and Tuning Knowledge Graph Embeddings', *arXiv:1907.01068 [cs, stat]*, Jul. 2019, Accessed: Jul. 28, 2021. [Online]. Available: <http://arxiv.org/abs/1907.01068>
- [29] M. A. Wani and S. Afzal, 'A New Framework for Fine Tuning of Deep Networks', in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, Dec. 2017, pp. 359–363. doi: 10.1109/ICMLA.2017.0-135.
- [30] K. Jain and S. Kaushal, 'A Comparative Study of Machine Learning and Deep Learning Techniques for Sentiment Analysis', in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, Aug. 2018, pp. 483–487. doi: 10.1109/ICRITO.2018.8748793.
- [31] Z. Zhang, 'Improved Adam Optimizer for Deep Neural Networks', in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, AB, Canada, Jun. 2018, pp. 1–2. doi: 10.1109/IWQoS.2018.8624183.
- [32] J.-K. Fang, C.-M. Fong, P. Yang, C.-K. Hung, W.-L. Lu, and C.-W. Chang, 'AdaGrad Gradient Descent Method for AI Image Management', in *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, Taoyuan, Taiwan, Sep. 2020, pp. 1–2. doi: 10.1109/ICCE-Taiwan49838.2020.9258085.
- [33] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, 'A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks', in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Belgaum, India, Dec. 2018, pp. 92–99. doi: 10.1109/CTEMS.2018.8769211.
- [34] Y. H. Chang *et al.*, 'Deep learning based Nucleus Classification in pancreas histological images', in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Seogwipo, Jul. 2017, pp. 672–675. doi: 10.1109/EMBC.2017.8036914.
- [35] K. Gregory, 'A dataset describing data discovery and reuse practices in research', *Sci Data*, vol. 7, no. 1, p. 232, Dec. 2020, doi: 10.1038/s41597-020-0569-5.
- [36] G. Linden, B. Smith, and J. York, 'Amazon.com recommendations: item-to-item collaborative filtering', *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003, doi: 10.1109/MIC.2003.1167344.