MADLINK: Attentive Multihop and Entity Descriptions for Link Prediction in Knowledge Graphs

Russa Biswas *, Harald Sack, and Mehwish Alam,
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure &
Institute for Applied Informatics and Formal Description Systems (AIFB), Karlsruhe Institute of Technology, Karlsruhe Germany
E-mails: russa.biswas@fiz-karlsruhe.de, harald.sack@fiz-karlsruhe.de, mehwish.alam@fiz-karlsruhe.de

Abstract. Knowledge Graphs (KGs) comprise of interlinked information in the form of entities and relations between them in a particular domain and provide the backbone for many applications. However, the KGs are often incomplete as the links between the entities are missing. Link Prediction is the task of predicting these missing links in a KG based on the existing links. Recent years have witnessed many studies on link prediction using KG embeddings which is one of the mainstream tasks in KG completion. To do so, most of the existing methods learn the latent representation of the entities and relations whereas only a few of them consider contextual information as well as the textual descriptions of the entities. This paper introduces an attentive encoder-decoder based link prediction approach considering both structural information of the KG and the textual entity descriptions. Random walk based path selection method is used to encapsulate the contextual information of an entity in a KG. The model explores a bidirectional Gated Recurrent Unit (GRU) based encoder-decoder to learn the representation of the paths whereas SBERT is used to generate the representation of the entity descriptions. The proposed approach outperforms most of the state-of-the-art models and achieves comparable results with the rest when evaluated with FB15K, FB15K-237, WN18, WN18RR, and YAGO3-10 datasets.

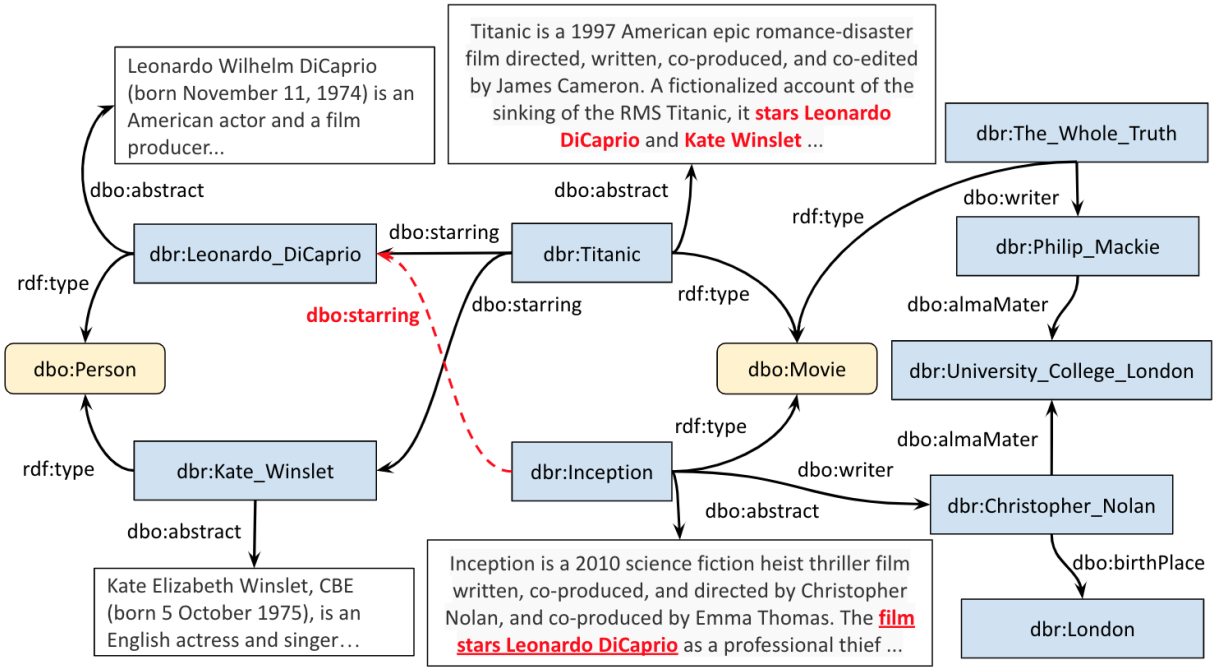
Keywords: Knowledge Graph Embedding, Encoder-Decoder, Link Prediction, Path Selection

1. Introduction

Knowledge Graphs (KGs) have recently gained attention for representing structured knowledge about a particular domain. Since its advent, the Linked Open Data (LOD) cloud[1] has constantly been growing containing many KGs about numerous different domains such as government, scholarly data, biomedical domain, etc. Apart from facilitating the inter-connectivity and interoperability of datasets in LOD cloud, KGs have been used in a variety of applications based on Machine Learning and Natural Language Processing (NLP) such as entity linking [1], question answering [2], recommender systems [3], etc. Some KGs are automatically generated from heterogeneous resources such as text, images, etc., whereas others are manually-curated. These KGs consist of huge amounts of facts in the form of entities (nodes) and relations (edges) between them. Also, the KGs contain facts in which the entities are connected to literals, i.e., text, numbers, images, etc. However, one of the major challenges is that KGs are sparse

*Corresponding author. E-mail: russa.biswas@fiz-karlsruhe.de.
[1]https://lod-cloud.net/

Fig. 1. An excerpt of the graph structure from DBpedia



and often incomplete as the links between the entities are missing. To overcome these challenges, predicting the missing links between the entities is necessary.

Link prediction is a fundamental task that aims to estimate the likelihood of the existence of edges (links) based on the current observed structure of a graph [4]. To-date many algorithms for generating KG embeddings have been proposed for the task of link prediction. Most of the state-of-the-art (SOTA) embedding models learn the vectors of the entities and relations from the triple information [5–9], i.e., only consider one hop information, whereas a very few of them explicitly take into account the neighbourhood information of an entity in the KG [10, 11]. Also, only a handful of them consider the literal information available in the KGs [12–16].

The textual description of the entities in the KGs contains rich semantic information and the graph structure provides the contextual information of the entity from the neighboring nodes. The graph given in Figure 1 contains information about some entities from DBpedia [17] and the relations between them. The textual entity description is given by the property `dbo:abstract`. In this graph, the information that `dbr:Leonardo_DiCaprio` and `dbr:Kate_Winslet` *acted* in the *movie* `dbr: Titanic` is given by the property `dbo:starring`. However, the `dbo:starring` information is missing for the *movie* `dbr:Inception`. But this information about `dbo:starring` is available in the textual entity description of the movie `dbr:Inception` given by the relation `dbo:abstract` which states *'The film stars Leonardo DiCaprio as a professional thief.'* Therefore, the information present in the textual entity description plays a vital role in predicting the missing links.

On the other hand, as shown in Figure 1, the movies `dbr:Inception` and `dbr:The_Whole_Truth` are written by `dbr:Christopher_Nolan` and `dbr:Philip_Mackie` respectively, as given by the relation `dbo:writer`. Also, both these writers have attended the same university `dbr:University _College_London`. Therefore, the movies `dbr: Inception` and `dbr:The_Whole_Truth` are similar in terms of being written by writers who went to the same university. This information from the graph is obtained by 2-hops starting from both source entities `dbr:Inception` and `dbr:The_Whole_Truth` and is given by

$dbr:Inception \xrightarrow{dbo:writer} dbr:Christopher\_Nolan \xrightarrow{dbo:almaMater} dbr:University\_College\_London$, and

$dbr:The\_Whole\_Truth \xrightarrow{dbo:writer} dbr:Philip\_Mackie \xrightarrow{dbo:almaMater} dbr:University\_College\_London$.

Considering the similarity in the given contextual information, the two movies `dbr:The_Whole_Truth` and `dbr:Inception` should be closer to each other in the vector space. Therefore, graph walks is beneficial for mod-

eling the latent representation of the entities and the relations. However, incorporating the contextual information of an entity from the graph is non trivial as not all relations are equally important to an entity.

Primarily, link prediction is the task of predicting the head or tail entities in a triple in a KG. However, triple classification, i.e., the task of finding if a given triple is valid or not in a KG is also considered as link prediction as it determines the validity of links between two entities. This work proposes a novel method, MADLINK, which improves the task of link prediction by combining the graph walks and textual entity descriptions to better capture the semantics of entities and relations. The model also incorporates contextual information about the relations in the triples. MADLINK adapts the seq2seq [18] encoder-decoder architecture with attention layer to obtain a cumulative representation of the paths extracted for each entity from the KG. On the other hand, SBERT [19] has been used to extract the latent representations of the entity descriptions provided as natural language text. DistMult [20] is used as a base model to calculate the score of a triple for head or tail prediction.

The effectiveness of the model is evaluated with the benchmark datasets FB15K [5], FB15K-237 [21], WN18 [5], WN18-RR [22], and YAGO3-10 [22] against different SOTA models with and without entity descriptions. The results show that MADLINK outperforms most of the SOTA models whereas achieves comparable results with the rest. The main contributions of this paper are:

- A path selection approach is introduced by exploiting the importance of a relation w.r.t. an entity in the KG.
- The textual entity descriptions is combined with the contextual information of the entities extracted from the paths for better representation of entities and relations in KGs.
- An end-to-end attention based encoder-decoder framework is proposed to generate better representation of the paths of entities, relations as well as entity descriptions for the link prediction task.

The rest of the paper is organised as follows. To begin with, a review of the related work is provided in Section 2, followed by the problem formulation in Section 3. Section 4 accommodates the outline of the proposed approach followed by experimental results in Section 5. Finally, Section 6 concludes the paper with a brief discussion on future directions.

2. Related Work

A large variety of KG embedding approaches have been proposed for the task of link prediction, such as (1) translational models such as TransE [5] and its variants, RotatE [23], etc., (2) semantic matching models such as SME [24], DistMult [20], RESCAL [25] and its extensions, ComplEx [26], HolE [27], etc., (3) neural network based models such as NTN [28], ConvE [22], ConvKB [29], HypER [30], R-GCN [31], etc., (4) path based methods, e.g., GAKE [10], PTransE [32], RDF2Vec [11], PConvKB [33], etc., (5) entity type based like SSE [34], TKRL [16], HAKE [35], and (6) literal-based models, such as methods making use of (i) textual literals: DKRL [16], Jointly(ALSTM) [36], SSP [37], KG-BERT [38], Multi-task learning KG-BERT [39], BLP [40], (ii) numeric literals: KBLRN [14], LiteralE [41], TransEA [13], MT-KGNN [42], and (iii) multi-modal literals: MKBE [15], IKRL [43], etc. The proposed approach MADLINK exploits both the path information and the textual entity descriptions of the entities to predict the missing links in the KG. Therefore, this section focuses mainly on the state-of-the-art (SOTA) models considering path information and the textual entity descriptions along with the other KG embedding models.

Translation-based Models: In TransE, given a triple $< e_h, r, e_t >$ in a KG G, the relation r is considered as a translation operation between the head (e_h) and tail (e_t) entities on a low dimensional space defined by $\mathbf{e_h} + \mathbf{r} \approx \mathbf{e_t}$, where $\mathbf{e_h}, \mathbf{r}, \mathbf{e_t}$ are the embeddings of the head, relation and the tail entity respectively. TransH [6] extends TransE by projecting the entity vectors to relation specific hyperplanes. TransR [44] models entities and relations into distinct semantic spaces and projects entities from the entity space to the relation spaces. The scoring function of RotatE models the relation as a rotation in a complex plane to preserve the symmetric/anti-symmetric, inverse, and composition relations in a KG. For a triple $< e_h, r, e_t >$, the relation among them can be represented as $e_t = e_h \circ r$, where $|r| = 1$, i.e., restricting it to the unit circle and $\circ$ represents element wise product.

Semantic Matching Models (SME): SME is based on semantic matching using neural network architectures. Given a triple $< e_h, r, e_t >$ in a KG G, it projects entities and relations to their vector embeddings in the input layer.

The relation vector $\mathbf{r}$ is then combined with the head entity vector $\mathbf{e_h}$ and tail entity vector $\mathbf{e_t}$ to get $g_u(\mathbf{e_h}, \mathbf{r})$, and $g_v(\mathbf{e_t}, \mathbf{r})$ respectively in the hidden layer. The final score is given by matching g_u and g_v via their dot product. In DistMult, each entity is mapped to a $d-$dimensional dense vector and each relation is mapped to a diagonal matrix, and the score of a triple is computed with the help of matrix multiplication between the entity vectors and the relation matrix. RESCAL models the triple $< e_h, r, e_t >$ into a three-way tensor, X. In X, two modes hold the concatenated entities (e_h, e_t), and the third mode holds relation r. The model explains triples via pairwise interaction of latent features. The score of the triple is calculated using the weighted sum of all the pairwise interactions between the latent features of the entities (e_h, e_t). ComplEx extends DistMult by introducing complex-valued embeddings so as to better model asymmetric relations. It allows joint learning of head and tail entities by using Hermitian dot product. In this model, the entity and relation embeddings lie in a complex space. Holographic Embedding (HolE) intend to overcome the curse of dimensionality of tensor product used in RESCAL by using circular correlation. Furthermore, this circular Correlation is not commutative allowing HolE to model asymmetric relations.

Neural Network Based Models (NTN): NTN represents an entity using the average of the word embeddings in the entity name. ConvE uses 2D convolutional layers to learn the embeddings of the entities and relations in which the head entity and the relation embeddings are reshaped and concatenated which serves as an input to the convolutional layer. The resulting feature map tensor is then vectorized and projected into a k-dimensional space and matched with the tail embeddings using logistic sigmoid function minimizing the cross-entropy loss. In ConvKB, each triple $< e_h, r, e_t >$ is represented as a 3-column matrix which is then fed to a convolution layer. Multiple filters are operated on the matrix in the convolutional layer to generate different feature maps. Next, these feature maps are concatenated into a single feature vector representing the input triple. The feature vector is multiplied with a weight vector via a dot product to return a score which is used to predict whether the triple is valid or not. HypER model uses a hypernetwork approach to generate convolutional filter weights for each relation. It is a method by which one network generates weights for another network enabling weight-sharing across layers. A hypernetwork projects a relation embedding via a fully connected layer, the result of which is reshaped to give a set of convolutional filter weight vectors for each relation. Relational Graph Convolutional Network (R-GCN) extends Graph Convolutional Network (GCN) to handle different relationships between entities in a KG. In R-GCN, different edge types use different weights and only edges of the same relation type r are associated with the same projection weight. For each node, an R-GCN layer operates in two steps. First, it computes the outgoing message using node representation and weight matrix associated with the edge type. Then the incoming messages are aggregated to generate new node representations. A 2-D matrix is used to define the initial node features and 3-D tensor describes the node hidden features. This tensor is able to encode different relations by stacking r batches of matrices, where each of these batches encodes a single typed relation.

Path Based Models: PTransE extends TransE by introducing a path based translation model. GAKE considers the contextual information in the graph by considering the path information starting from an entity. RDF2Vec uses random walks to consider the graph structure and applies word embedding model on the paths to learn the embeddings of the entities and the relations. However, the prediction of head or tail entities with RDF2Vec is non trivial because it is based on language modeling approach. PConvKB model extends the ConvKB model by exploiting the path information of the KGs. Additionally, it uses an attention mechanism to measure the local importance in relation paths.

Literal Based Models: Another set of algorithms improve KG embeddings by taking into account different kinds of literals such as numeric, text or image literals and a detailed analysis of the methods is provided in [45].

Text Based Models: DKRL extends TransE [5] by incorporating the textual entity descriptions in the model. The textual entity descriptions are encoded using a continuous bag-of-words approach as well as a deep convolutional neural network based approach. The energy function of the model is defined by $E = E_S + E_D$, where E_S represents the energy function of the structure based representation and E_D represents the energy function of the description based representation. The structure based representation is learned by TransE model. Jointly (ALSTM) is another entity description based embedding model which extends the DKRL model with a gate strategy and uses attentive LSTM to encode the textual entity descriptions. KG-BERT is a contextual language model based model which fine tunes the BERT model on the KGs. Each triple $< e_h, r, e_t >$ is considered as a sentence and is provided as an input sentence of the BERT model for fine-tuning. For the entities, KG-BERT has been trained with either the entity names or their textual entity descriptions and for relations, the relation names are used. The first token of every

input sequence is always $[CLS]$, whereas the separator token $[SEP]$ separates the head entity, relation and the tail entity. Multi-task learning KG-BERT model intends to improve the BERT model in which the authors introduce an effective multi-task learning by combining relation prediction and relevance ranking tasks together with the link prediction. This model is trained to learn relational properties in KGs and perform even when lexical similarity occurs. BLP framework on the other hand, uses BERT based entity representations from textual entity descriptions for link prediction on top of various base KG embedding models namely TransE, DistMult, ComplEx, SimplE.

Other methods such as TransEA [13] and KBLRN [14] incorporate numeric literals into their embedding spaces. Also, MKBE is a multi-modal KG embedding model which includes the numeric, text and image literals present in KGs into their embedding spaces. However, in the aforementioned models the neighborhood node structure is not considered together with the textual entity descriptions to learn the embedding models. Therefore, this study proposes a novel model, MADLINK, which includes the contextual structural information as well as the entity descriptions into the embedding space for the task of Knowledge Graph Completion (KGC) using link prediction.

3. Problem Formulation

Given a KG $G = (E, R)$, where E is the set of entities, R is the set of relations. $< e_h, r, e_t > \in T$, represents a triple belonging to the set of triples T in the KG, where $(e_h, e_t) \in E$ are the head and tail entities, and $r \in R$ represents relation between them. Besides, facts in the form of literals such as text, images, numbers, etc. are also connected to the entities in a KG. But in this work we focus on predicting the missing links between entities. Furthermore, most of the KGs comprise of textual descriptions for each entity providing semantic information about it. MADLINK aims to learn the latent representation of the entities and relations to a lower dimensional embedding space, $\mathbb{R}^d$, where d is the dimension of the embedding space for the task of link prediction. This section discusses the research questions to address the challenges.

- *RQ1: Does the contextual information of entities and relations in a KG help in the task of link prediction?*
- *RQ2: What is the impact of incorporating textual entity descriptions in a KG for the task of link prediction?*

4. MADLINK Model

This section comprises a detailed step-wise description of the proposed model and the training approach. The model consists of two parts: (i) structural and (ii) textual representation. Path selection forms the primary step of the structural representation whereas textual representation is the encoding of the textual entity descriptions.

4.1. Path Selection

A directed path in a directed labelled graph is a sequence of edges connecting a sequence of distinct vertices. Given a KG G, a path can be defined as $\left\{e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} ... \xrightarrow{r_m} e_n\right\}$, where $(e_i, r_j), i \in \{1, 2, ..., n\}$ and $j \in \{1, 2, ..., m\}$ are the entities and relations, respectively. Starting from a certain entity, the paths capture the contextual information of an entity in a KG. However, huge amount of information is stored in the KG and not all triples are equally important for an entity. Some of the triples explain the characteristics of an entity better than others. For example, in Figure 1, for the entity `dbr:Christopher_Nolan` the relation `dbo:almaMater` provides more specific information as compared to `dbr:birthPlace` as most of the persons in DBpedia certainly have a birth place. Also as explained in Section 1, the path containing the relation `dbo:almaMater` provides more contextual information for the entities `dbr:Inception` and `dbr:The_Whole_Truth`. Therefore, it is essential to know the general importance of the relations for each entity. Eventually the paths containing these relations would provide more valuable information compared to the paths without these relations. To tackle this challenge, Predicate Frequency Inverse Triple Frequency (PF-ITF) is used to identify the important relations for each entity [46].

Predicate Frequency - Inverse Triple Frequency (PF-ITF)

In order to extract the contextual information related to an entity, paths consisting of $l-$hops are generated for each node. The properties are selected at each hop of the path using PF-ITF. Also, the cycles present in the KGs are straightened and considered as a flat path. Given a KG G, the predicate frequency of outgoing edges is given by $pf_o^e(r,G)$, the inverse triple frequency is given by $itf(r,G)$ and PF-ITF $pf-itf_e(r,G)$ is computed based on Equation 1.

$$\begin{aligned} pf_o^e(r,G) &= \frac{|\varepsilon_o(e)|_{\pi(r)}}{|\varepsilon_o(e)|}, \\ itf(r,G) &= log\frac{|\varepsilon|}{|\varepsilon|_{\pi(r)}}, \\ pf-itf_e(r,G) &= pf_e \times itf, \end{aligned} \tag{1}$$

where $\pi(r)$ is the set of relations, $|\varepsilon_o(e)|_{\pi(r)}$ represents the number of outgoing edges from the entity e w.r.t. to the relation r, $|\varepsilon_o(e)|$ is the total number of outgoing edges for the entity e, $|\varepsilon|$ is the total number of triples, and $|\varepsilon|_{\pi(r)}$ is the total number of triples containing the relation r. In this work, the paths are generated starting from a certain entity, so PF-ITF is calculated using the outgoing edges, i.e., Eq. 1.

Next, the relations per entity are ranked based on the PF-ITF score. The PF-ITF value increases proportionally with the number of outgoing edges of an entity w.r.t. a relation and is offset by the total number of triples containing the relation which helps to adjust the relations which appear more frequently in general. The highest PF-ITF score of a relation w.r.t. an entity indicates that the triples containing this relation have more information content compared to the rest. Based on the ranks, top-n relations are selected for each entity. Thereafter, paths are generated for the entities in the KG and crawled iteratively until $l-hops$. For computational simplicity, top-m important properties are considered for each entity based on the PF-ITF score.

4.2. Textual Representation

The textual descriptions of an entity provide semantic information. These descriptions are of variable length and are often short, i.e., less than or equal to 3 words. The textual entity descriptions are encoded into a vector representation. Also, an enormous amount of text data is available outside the KGs which can be leveraged for better representation of the entities. The static pre-trained language models such as Word2Vec [47], GloVe [48], etc. as well as the contextual embedding model such as BERT [49] have been extensively used to generate latent representations of Natural Language Text. BERT applies transformers which is an attention-based mechanism to learn contextual relations between the words and/or sub-words in a text. The transformer encoder reads the entire sequence of words at once which allows the model to learn the context of a word based on its surroundings.

Sentence-BERT (SBERT) [19] is a modification of BERT which provides more semantically meaningful sentence embeddings using Siamese and triplet networks. As mentioned in [19], independent sentence embeddings are not computed in BERT. The semantically similar sentences cannot be compared using cosine similarity. Therefore, sentence embeddings are obtained by averaging the outputs to derive a fixed-size vector [50, 51]. This is similar to average word embeddings generated by static models such as GloVe. However, it is also observed by the authors in [19] that the average BERT embeddings perform worse than average GloVe embeddings for various tasks, such as textual similarity, Wikipedia Sections Distinction, etc. SBERT model tackles all the above mentioned problems.

SBERT fine-tunes the BERT model using the siamese and triplet networks to update the weights such that the resulting sentence embeddings are semantically meaningful and semantically similar sentences appear closer to each other in the embedding space. It is fine-tuned with a 3-way softmax classifier objective function for one epoch. The two input sentences (say u and v) to the SBERT model are passed through the BERT model followed by a pooling layer namely, CLS-token, MEAN-strategy, and MAX-strategy are appended on top of it. This pooling layer enables the generation of a fixed-size representation for the input sentences. It is then concatenated with the element-wise difference and multiplied with a trainable weight, W, and is optimized using cross-entropy loss. In order to encode the semantics, the twin network is fine-tuned on Semantic Textual Similarity data. SBERT model is first trained

on Wikipedia via BERT and then fine-tuned on Natural Language Inference (NLI) data. NLI is a collection of 1,000,000 sentence pairs created by combining The Stanford Natural Language Inference (SNLI)[2] and Multi-Genre NLI (MG.NLI) datasets.

Also, [19] shows that the sentence embeddings generated by SBERT outperform BERT for SentEval toolkit, which is popularly used to evaluate the quality of sentence embeddings. In this work, the sentence embeddings from the pre-trained SBERT model which are fine-tuned with SNLI and STS datasets, are extracted. It follows the same approach as followed in [19] for SentEval. Therefore, two sentences are not required as input to obtain the sentence embeddings. In this work, the input to the SBERT model is only the entity descriptions. The similar entities in the KG should have similar textual entity descriptions and hence the embedding obtained for the entity descriptions should exhibit similar characteristics. SBERT is designed to minimize the distance between two semantically similar sentences in the embedding space. Therefore, SBERT is leveraged in this work, to obtain similar encoding of the entity descriptions for similar entities. Also, the authors of [19] fine-tune Roberta with the same approach as SBERT and the result shows that the performance of SRoberta and SBERT are almost similar for different tasks and they outperform their respective base models. Furthermore, SBERT outperforms SRoberta in some of the tasks [19]. Also, the model used in this work is SBERT-SNLI-STS-base model which outperforms SRoberta-SNLI-STS-base model as shown in [19].

The sentence embeddings obtained from SBERT model lose the domain-specific knowledge as it is trained and fine-tuned with two different datasets. Therefore, these sentence embeddings generated by SBERT perform better for a wide variety of tasks. In this work, to encode the textual description of the entities in a KG, the default pooling method of SBERT model, i.e., the MEAN pooling has been used and the entire entity description is considered as one sentence. The fine-tuning of the original BERT model with the textual entity descriptions from both the datasets has not been performed because the original BERT model is trained with Wikipedia and these textual entity descriptions are the abstracts of the Wikipedia articles. Therefore, further fine-tuning would have resulted in overfitting. Since SBERT is already fine-tuned with SNLI data, we opted for this model.

4.3. Encoder - Decoder Framework

In this work, a sequence-to-sequence (seq2seq) learning-based encoder-decoder model [18] is adapted to learn the representation of the path vectors in the KGs, the description vectors as well as the relation vectors. Figure 2 depicts the encoder-decoder architecture to generate the path embeddings.

Encoder The encoder aims at encoding the entire input sequence into a fixed length vector called a context vector. A path $p_i \in P$, where p_i is a path which is a sequence of entities and the relations between them and is given by, $\left\{e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} ... \xrightarrow{r_m} e_n\right\}$ is considered as a sentence and the entities e_i and relations r_j are the words. The input to the encoder is the randomly initialized vectors of entities and relations that appear in the paths. These embeddings are passed through a Bi-directional GRU [52] which encapsulate the information for all input elements and compresses them into a context vector along with the representation of the final hidden states $\mathbf{A} = \mathbf{a_1 a_2 ... a_n}$ where $\mathbf{a_t}$ is given by,

$$\mathbf{a_t} = \mathbf{GRU}(\mathbf{a_{(t-1)}}, \mathbf{embed}(\mathbf{x_t})), \tag{2}$$

where $embed(x_t)$ is the embedding of entities and relations. In a multi-gated GRU, for each element in the input sequence, the following equations are calculated for each layer.

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{ar}a_{(t-1)} + b_{ar}), \tag{3}$$

$$z_t = \sigma(W_{ia}x_t + b_{ia} + W_{ha}a_{(t-1)} + b_{ha}), \tag{4}$$

$$n_t = \tanh\left(W_{in}x_t + b_{in} + r_t * (W_{an}a_{(t-1)} + b_{an})\right), \tag{5}$$

$$a_t = (1 - z_t) * n_t + z_t * a_{(t-1)}, \tag{6}$$

[2] https://nlp.stanford.edu/projects/snli/

Fig. 2. Encoder - Decoder Framework for paths

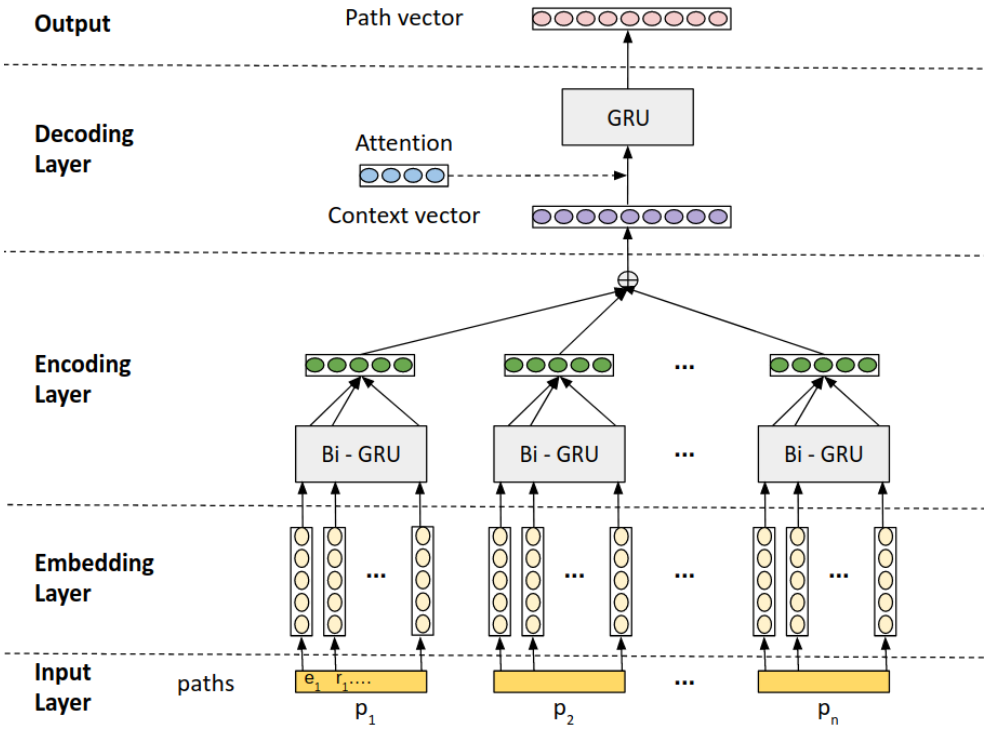
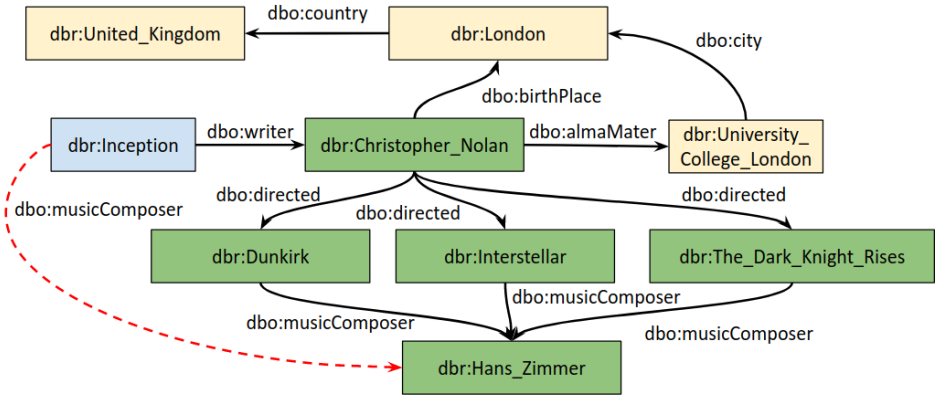


Fig. 3. Illustration of the attention for a path in predicting the 'dbo:musicComposer' for the movie Inception

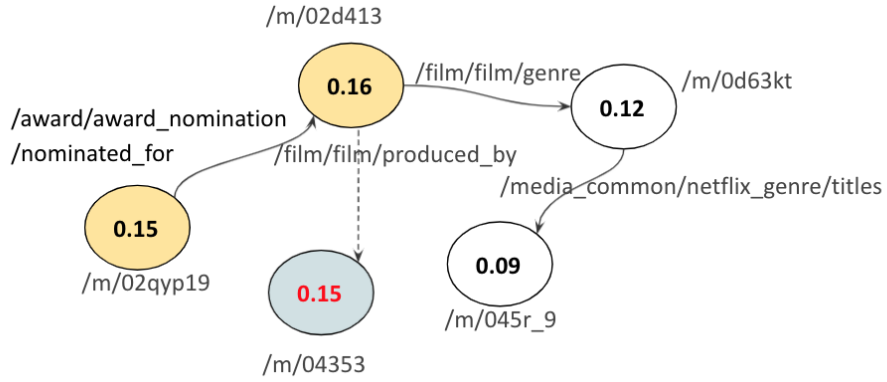


where W_{ir}, W_{ar}, W_{in}, and W_{an} are the weight matrices, x_t is the input at time t, r_t, z_t, n_t are the reset, update and new gates, respectively, and $\sigma, *$ are the sigmoid and Hadamard product, respectively. The context vector and the final hidden state of the encoder model is then passed through an attention layer to learn the weights. Similarly, for relation encoding, instead of the paths the triples containing the relation are considered.

Self-Attention. An attention mechanism allows a neural network to focus on a subset of its inputs or features and is given by,

$$\begin{aligned} \mathbf{attn} &= \mathbf{f}_{\phi}(\mathbf{x}), \\ g &= \mathbf{attn} \odot \mathbf{z}, \end{aligned} \tag{7}$$

Fig. 4. Illustration of the attention weights for an excerpt from FB15k



where $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$ is an input vector, $\mathbf{z} \in \mathbb{R}^{\mathbf{k}}$ is a feature vector, $\mathbf{attn} \in [\mathbf{0}, \mathbf{1}]^{\mathbf{k}}$ is an attention vector, and $f_\phi(x)$ is an attention network with parameters ϕ[3]. As explained in [10], given an input path sequence $\left\{e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} ... \xrightarrow{r_m} e_n\right\}$, not all the relations are equally important to model a specific fact. Some of them might be important for a certain entity but not for others and vice-versa. PF-ITF helps in identifying the important relations with respect to an entity in the KG. Now, the attention mechanism allows to identify the important relations and other entities in the paths w.r.t. a certain entity e_h or e_t. Therefore, it is used to generate the contextual path encoding. For example, in Figure 3, to predict the tail entity of the triple <dbr:Inception, dbo:musicComposer, t>, the nodes marked in green would have greater attentions than the ones marked in yellow. Therefore, the paths starting from the node dbr:Inception which contain the nodes marked in green are impactful in predicting the dbo:musicComposer for Inception. Also, for the relation encoding, not all triples are equally important for a certain relation. Similarly, for textual encoding not all the words and phrases in the entity descriptions are equally important to represent a certain entity. Hence, an attention mechanism is also used here to generate the contextual description encoding depending on different words in the text.

As the attention mechanism, the scaled dot product self attention [53] is used because it is much faster and is more space-efficient. Queries and keys of dimension d_k, and values of dimension d_v are given as input. Then the dot product of the query is computed with all keys. Each of them is then divided by $\sqrt{d_k}$. A softmax function gives the weights on the values as an output. Practically, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. In this work, the final hidden layer of the encoder is taken as the query as well as the key while value is the output, i.e., the context vector. The scaled dot product self-attention is given as,

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{8}$$

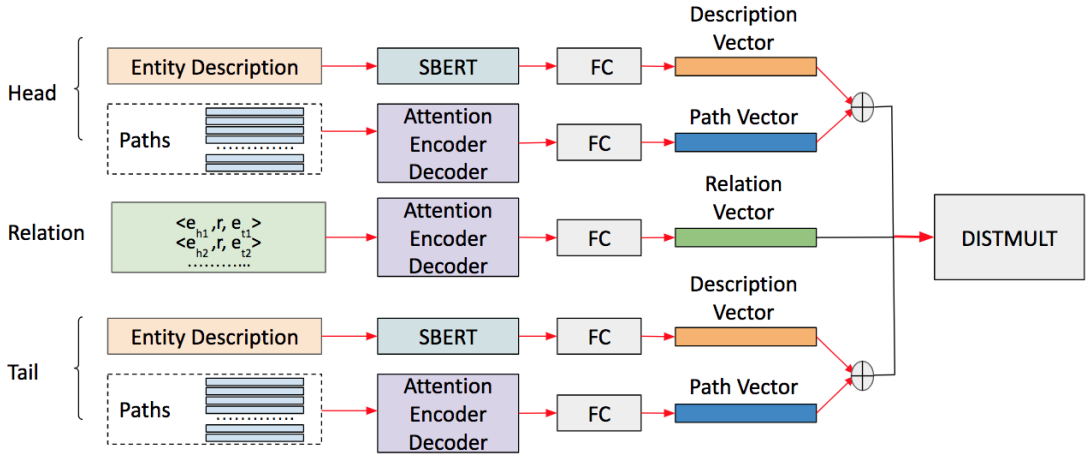
In terms of the MADLINK model, for a given word or relation x the above equation can be rewritten as,

$$\alpha(x) = softmax(\frac{a_t a_t^T}{\sqrt{dim(h_t)}})X, \tag{9}$$

where a_t is the hidden layer, $dim(a_t)$ is the dimension of the hidden layer, and X is the context vector. The attention weights of an excerpt from FB15k is illustrated in Figure 4. The nodes in yellow have the maximum attention which is required to predict the entity $/m/04353$.

[3]http://akosiorek.github.io/ml/2017/10/14/visual-attention.html

Fig. 5. Overall Architecture of the MADLINK model



Decoder The attention layer forms a bridge between the path embeddings and the input path sequences. The decoder network is initialized with the attention weights and the context vector which is then fed to a layer of GRU to obtain the final *Path Vector* for each entity. Therefore, this Path Vector gives a representation of the entity. Figure 2 illustrates the encoder-decoder architecture used in the paper.

The main advantage of using this seq2seq based encoder-decoder in the MADLINK architecture is that it can generate output sequence after seeing the entire input. The attention mechanism allows focusing on specific parts of the input automatically to help generate a useful encoding, even for longer input. Therefore, the proposed MADLINK model looks into all the input paths for a certain entity, focuses on the specific parts of the input, and then generates an encoding for the entity.

4.4. Overall Training

Given a triple $< e_h, r, e_t >$, the encoding of the head and tail entities are generated by the respective path vectors as discussed in Section 4.3. Similarly, for a relation r, all the triples in the KG containing that relation are considered to generate the relation encoding. The textual representation of the entities are obtained from the embeddings of the entity descriptions. The overall architecture of the MADLINK model is depicted in Figure 5. Therefore, the parameters of the model are as follows: $\theta = \{\mathbf{D_h}, \mathbf{D_t}, \mathbf{P_h}, \mathbf{P_t}, \mathbf{R}, \mathbf{GRU_1}, \mathbf{GRU_2}\}$, where $\mathbf{D_h}, \mathbf{D_t}$ are the description embedding of the head and tail entities, respectively obtained from SBERT model, $\mathbf{P_h}, \mathbf{P_t}$ are the path embeddings of the head and tail entities, respectively, $\mathbf{R}$ is the relation embeddings, $\mathbf{GRU_1}$ and $\mathbf{GRU_2}$ represent the parameters from the Bi-directional GRU and the decoder GRU, respectively. Finally, the path $(\mathbf{P_h}, \mathbf{P_t})$, relation $(\mathbf{R})$ and the description embeddings $(\mathbf{D_h}, \mathbf{D_t})$ are passed through one fully connected layer with the same weights. The dimension of the SBERT embeddings of textual entity descriptions is 1024 which is reduced to a dimension of 100 or 150 based on the size of input embedding vector to the DistMult model for different datasets.

In this work, DistMult is used as the final scoring function of the model. The model uses a simplification of bilinear interaction between the entities and the relations. DistMult model uses the trilinear dot product as a scoring function

$$f_{DistMult} =< r_p, e_h, e_t >, \tag{10}$$

where e_h, e_t, r_p are the embeddings of the head, tail, and relation, respectively. In MADLINK, the D_h and P_h are concatenated and initialized as the head embedding to the scoring function. A similar operation is done with the tail entity as well.

Table 1
Statistics of the benchmark datasets

Datasets	FB15K	FB15K-237	WN18	WN18RR	YAGO3-10
#Entities	14,951	14,541	40,943	40,943	123,182
#Relations	1,345	237	18	11	37
#Entities with Description	14,515	14,541	40,943	40,943	107,326
Train set	483,142	272,115	141,441	86,834	1,079,040
Test Set	59,071	20,466	5,000	3,134	5,000
Valid Set	50,000	17,535	5,000	3,034	5,000

5. Experiments and Results

This section discusses the benchmark datasets and the experiments conducted for showing the feasibility of MADLINK and its empirical evaluation on two KGC tasks, namely, link prediction, i.e., head or tail prediction and triple classification.

5.1. Datasets

The statistics of the datasets FB15K, FB15K-237, WN18, WN18RR, and YAGO3-10 used for the purpose of the evaluation are provided in Table 1. FB15K is a dataset extracted from large scale cross-domain KG, Freebase [54]. As mentioned in [21, 22], FB15K has 80.9% test leakage, i.e., a large number of test triples are obtained by inverting the triples of the test set. For example, <*Republic, /government/form_of_government/countries, Paraguay*> is a triple from the training set of FB15K and its inverse <*Paraguay,* $/location/country/form\_of\_government$*, Republic*> is a triple in the test set, where *Republic* and *Paraguay* are the entities in Freebase and $/government/form\_of\_government/countries$ is the relation between them. The triple from the test set is the inverse of the triple from the training set. As mentioned by the authors, this might lead to the models for learning relations and their corresponding inverse relations for the link prediction task instead of modelling the actual KG. Therefore, FB15K-237 dataset has been introduced by [21], which is a subset of FB15K without the inverse relations. Similarly, WN18 is extracted from WordNet [55] which contains word concepts and lexical relations between the concepts. WN18RR is a subset of WN18 without inverse relations.

YAGO3-10 [22] is extracted from the large scale cross-domain KG, YAGO [56]. It consists of those entities having at least 10 relations in YAGO. The authors state that the majority of triples in YAGO are the properties of people such as citizenship or gender. The poor performance of the inverse model ConvE [22] on YAGO3-10 implies that it is free from test leakage [57]. Therefore, it is observed, that most of the recent KG embedding models designed for the task of link prediction are evaluated on the FB15k-237 and WN18RR instead of FB15k and WN18. However, the proposed model MADLINK has been evaluated on all the aforementioned 5 benchmark datasets. Besides, MADLINK model uses textual entity descriptions along with the structural information of entities.

5.2. Experimental Setup

In the path selection process, the following parameters are used: number of hops 4, and number of paths per entity 1000. The hyper-parameters used in the MADLINK model are as follows: the dimension of SBERT vectors 1024, a learning rate of the encoder - decoder framework 0.001, batch size 100, loss margin 1, dropout 0.5. In the pre-processing step of the textual entity description, only punctuation removal is done. The experiments with MADLINK have been performed on an Ubuntu 16.04.5 LTS system with 503GiB RAM. The training with DistMult, SBERT, and the encoder-decoder framework is performed with TITAN X (Pascal) GPU.

5.3. Hyper-parameter Optimization

The hyper-parameter optimization is performed using grid search as provided in [58] and the hyper-parameters are selected with the best performance on the validation dataset. For all the benchmark datasets, the search space

Table 2
Hyper-parameter Search Space for MADLINK

Hyper-parameters	Range
Batches	{32, 64, 100}
Epochs	{500, 1000}
Embedding Size	{50, 100, 150, 200}
Eta (η)	{1, 5, 10}
Loss	Multiclass NLL
Regularizer Type	{L1, L2, L3}
Regularizer (λ_r)	{1e - 3, 1e - 4}
Optimizer param (lr)	{0.1, 0.01, 0.001}
Optimizer param (λ_o)	{1e - 5, 1e - 4, 1e - 3, 1e -2}

provided in Table 2 and the hyper-parameters used are provided in Table 3. Adam optimizer is used for the base model.

Table 3
Optimized hyper-parameters used in the training of MADLINK

Parameters	FB15K	FB15K-237	WN18	WN18RR	YAGO3-10
Batches	64	64	64	64	100
Embedding Size	150	150	150	150	150
Epochs	1000	1000	1000	1000	1000
Learning Rate (lr)	0.001	0.001	0.001	0.001	0.001
Regularizer	L3	L3	L3	L3	L3
Regularizer (λ_r)	1e - 4	1e - 4	1e - 5	1e - 5	1e - 4
Optimizer param (lr)	0.001	0.001	0.01	0.01	0.001
Optimizer param (λ_o)	1e - 3	1e - 3	1e - 3	1e - 3	1e - 3
Loss	Multiclass NLL	Multiclass NLL	Multiclass NLL	Multiclass NLL	Multiclass NLL

5.4. Link Prediction

Formally, link Prediction is a sub-task of KGC which aims at predicting the missing head (e_h) or tail entity (e_t) given (e_h, r) or (r, e_t) respectively [24]. Given a KG $G = (E, R)$, where E and R are the set of entities and relations, link prediction can be defined by a mapping function that assigns a score to every possible triple $(e_i, r, e_j) \in E \times R \times E$. A high score of a triple indicates it to be true [45]. However, instead of considering the best score triple, a set of candidates is considered based on the ranking of the scores.

Evaluation metrics Following the model in [5] the evaluation metrics used are as follows: (**1**) Mean Reciprocal Rank (MRR) is the average of the reciprocal ranks of the correct entities, and (**2**) Hits@k is the proportion of the correct entities in top-k predictions. To evaluate most of the embedding models, negative sampling is used to generate corrupted triples by removing either the head or the tail entity. In doing so, some of the generated corrupted triples might actually occur in the KG and should be considered as a valid triple. Therefore, all the triples which are true and are present in the training, test, and validation set are removed from the corrupted triples set and is termed as 'filtered' setting in the evaluation. Also, the triples containing unseen entities are removed from the test and the validation sets.

Baselines The effectiveness of the proposed model, MADLINK, is illustrated by comparing with the following baseline models. These baselines are selected based on the diversity of the nature of the embedding models such as translation-based, neural network-based, textual entity description-based, rotational-based, etc.

- TransE [5] is a translation-based embedding model.
- DistMult [20] is bilinear diagonal model.

Table 4
Comparison of MADLINK results with the textual entity description-based baseline models on the 4 benchmark datasets [4]

FB15K-237				
Models	MRR	Hits@1	Hits@3	Hits@10
DKRL	0.19	0.11	0.167	0.215
Jointly(ALSTM)	0.21	0.19	0.21	0.258
KG-BERT	0.237	0.144	0.26	0.427
Multitask-BERT	0.267	0.172	0.298	0.458
BLP-TransE	0.195	0.113	0.213	0.363
MADLINK^1	**0.347**	**0.252**	**0.38**	**0.529**
MADLINK^2	0.341	0.249	0.377	0.52
WN18RR				
Models	MRR	Hits@1	Hits@3	Hits@10
DKRL	0.112	0.05	0.146	0.288
Jointly(ALSTM)	0.21	0.112	0.156	0.31
KG-BERT	0.219	0.095	0.243	0.497
Multitask-BERT	0.331	0.203	0.383	**0.597**
BLP-TransE	0.285	0.135	0.361	0.580
MADLINK^1	**0.477**	**0.438**	**0.479**	0.549
MADLINK^2	0.471	0.43	0.469	0.535
FB15K				
Models	MRR	Hits@1	Hits@3	Hits@10
DKRL	0.311	0.192	0.359	0.548
Jointly(ALSTM)	0.345	0.21	0.412	0.65
SSP	-	-	-	0.771
MADLINK^1	**0.712**	**0.722**	**0.788**	**0.81**
MADLINK^2	0.69	0.714	0.78	0.798
WN18				
Models	MRR	Hits@1	Hits@3	Hits@10
DKRL	0.51	0.31	0.542	0.61
Jointly(ALSTM)	0.588	0.388	0.596	0.77
SSP	-	-	-	0.932
MADLINK^1	**0.95**	**0.898**	**0.911**	**0.96**
MADLINK^2	0.944	0.88	0.9	0.9
YAGO3-10				
Models	MRR	Hits@1	Hits@3	Hits@10
DKRL	0.19	0.119	0.234	0.321
Jointly(ALSTM)	0.22	0.296	0.331	0.41
MADLINK^1	**0.538**	**0.457**	**0.580**	**0.68**
MADLINK^2	0.528	0.447	0.573	0.67

– ConvE [22] is a CNN-based embedding model.
– ConvKB [59] is also a CNN-based embedding model in which each triple is represented by a 3-column matrix which is then fed to a convolution layer to generate different feature maps. These feature maps are then concatenated into a single feature vector representing the input triple.
– DKRL [16] and Jointly (ALSTM) [36] are textual entity description-based embedding models. The former uses a CNN approach, whereas the latter uses an LSTM.
– RotatE [23] is a model which defines each relation as a rotation from the source to the target entity in the complex vector space. The authors propose a self adversarial negative sampling technique for the model.

- HypER [30] uses a hyper network to generate 1D relation specific vectors convolutional filter weights for each relation which is then used by another network to enable weight sharing across layers.
- R-GCN [31] is a relation aware Graph Convolutional Network, in which the encoder learns the latent representation of the entities and the decoder is a tensor factorization model exploiting these representations for the task of link prediction.
- QuatE [60]. In Quaternion embeddings, hyper complex-valued embeddings with three imaginary components, are utilized to represent entities. Relations are modelled as rotations in the quaternion space.
- MDE [61] (Multiple Distance Embedding model) is a framework to collaboratively combine variant latent distance-based terms by using a limit-based loss and by learning independent embedding vectors. It uses a neural network model that allows the mapping of nonlinear relations between the embedding vectors and the expected output of the score function.
- TuckER [62] model is based on Tucker decomposition of the third-order binary tensor of triples. Tucker decomposition factorizes a tensor into a core tensor multiplied by a matrix along with each mode.
- KG-BERT [38] and Multi-task BERT [39] are the two models which exploit the working principle of the contextual language model BERT to predict missing links in a KG.
- BLP-TransE is one of the models proposed in Inductive Entity Representations from Text via link prediction [40], in which entity representations are generated from their textual entity descriptions using BERT and different KG embedding models such as TransE, DistMult, etc., are used on top of it for the task of link prediction.
- LiteralE [41] is a literal embedding model which uses DistMult, ConvE, and ComplEx as the base model. The main model is based on numeric literal which is easily extendable with text and image literal.
- SSP [37], the Semantic Space Projection (SSP) model jointly learns from the symbolic triples and textual descriptions which uses TransE as the base model. It follows the principle that triple embedding is considered always as the main procedure and textual descriptions must interact with triples in order to learn better representation. Therefore, triple embedding is projected onto a semantic subspace such as a hyperplane to allow strong correlation by adopting quadratic constraint.

Results

The proposed model MADLINK is compared against the aforementioned baseline models on the 5 benchmark datasets FB15k, FB15K-237, WN18, WN18RR, and YAGO3-10 as depicted in Tables 4, 5, and 6. In these tables, MADLINK^1 represents the results of the experiments in which the entities without textual entity descriptions are removed, whereas MADLINK^2 represents the results of the experiments containing all the entities in the datasets.

Comparison with textual entity description-based baseline models

It is to be noted that, out of the above-mentioned baselines, DKRL, Jointly(ALSTM), KG-BERT, Multitask-BERT, and BLP-TransE use textual entity descriptions as to their features for link prediction. Therefore, these models form the primary baseline for our proposed model MADLINK as shown in Table 4. For FB15K-237 dataset, MADLINK^1 outperforms the SOTA models for all the metrics with an improvement of 8% for MRR and Hits@1, 8.2% for Hits@3, and 7.1% for Hits@10 better than the best baseline model Multitask-BERT. Both DKRL and Jointly(ALSTM) models have the same experimental setup as MADLINK^2 variant, in which the models are trained on datasets that contain entities without text descriptions. MADLINK^2 variant outperforms DKRL with an increase of 15.1% on MRR, 13.1% on Hits@1, 21% on Hits@3, and 30.5% on Hits@10. Also, the proposed model outperforms the Jointly(ALSTM) model by 13.1% increase on MRR, 5.9%, 16.7%, and 26.2% on Hits@1, Hits@3, and Hits@10 respectively. For WN18RR, MADLINK outperforms the baseline models with considerable improvement with all the metrics except for Hits@10. Multi-task BERT model performs best for the WN18RR dataset for Hits@10, whereas for other metrics MADLINK outperforms the same model with an improvement of 14.6% on MRR, 23.5% on Hits@1, and 9.6% on Hits@3. Furthermore, both the variants of MADLINK considerably outperform all the baseline models DKRL, Jointly(ALSTM), KG-BERT, and BLP-TransE.

[4]The results marked in bold are the best results and the underlined ones are the second best.'-' is provided if the corresponding results are not available in the respective papers.

Table 5
Comparison of MADLINK results with the structure-based baseline models on FB15k-237 and WN18RR datasets

FB15K-237				
Models	MRR	Hits@1	Hits@3	Hits@10
TransE	0.31	0.22	0.35	0.5
DistMult	0.247	0.161	0.27	0.426
ConvE	0.26	0.19	0.28	0.38
ConvKB	0.23	0.15	0.25	0.40
RotatE	0.298	0.205	0.328	0.480
HypER	0.341	0.252	0.376	0.520
R-GCN	0.228	0.128	0.25	0.419
QuatE	0.311	0.221	0.342	0.495
MDE	0.344	-	-	0.531
TuckER	**0.358**	**0.266**	**0.394**	**0.544**
MADLINK[1]	0.347	0.252	0.38	0.529
MADLINK[2]	0.341	0.249	0.377	0.52
WN18RR				
Models	MRR	Hits@1	Hits@3	Hits@10
TransE	0.22	0.03	0.37	0.54
DistMult	0.438	0.424	0.442	0.478
ConvE	0.45	0.42	0.47	0.520
ConvKB	0.39	0.33	0.42	0.48
RotatE	0.476	-	-	**0.571**
HypER	0.465	0.436	0.477	0.465
R-GCN	0.39	0.338	0.431	0.49
QuatE	**0.481**	0.436	0.5	0.564
MDE	0.458	-	-	0.56
TuckER	0.47	**0.443**	**0.482**	0.526
MADLINK[1]	0.477	0.438	0.479	0.549
MADLINK[2]	0.471	0.43	0.469	0.535

The models KG-BERT, Multitask-BERT, and BLP-TransE have not been evaluated on FB15k, WN18 due to test leakage problem in the original papers [38]. Therefore, MADLINK is compared against the DKRL and Jointly(ALSTM) models. It is noted that both the MADLINK variants significantly outperform both the text-based baseline models for all the metrics. Similarly, for YAGO3-10 dataset, both the MADLINK variants show major improvement from both the baselines.

It is to be noted that all the aforementioned text-based KG embedding models, such as DKRL, and Jointly (ALSTM), exploit the structural information of the KG in form of triples explicitly together with the textual entity descriptions. However, KG-BERT and Multitask-BERT use the triple information implicitly as the triples are considered as input sentences to the BERT model. On the other hand, in BLP-TransE, the textual entity and relation representations are provided as input to the TransE model in form of triple inputs. Therefore, the results infer that the textual entity descriptions together with the structural information captured via the paths in the MADLINK model capture better semantics of the KG for the task of link prediction.

Comparison with structure-based baseline models

The proposed model MADLINK is compared with the baseline KG embedding models such as TransE, DistMult, ConvE, ConvKB, RotatE, HypER, R-GCN, QuatE, MDE, and TuckER, that consider the triple information to generate the latent representation of the entities for the task of link prediction in KGs. The results are shown in Table 5 on FB15k-237 and WN18RR and Table 6 on FB15k, WN18, and YAGO3-10 datasets.

MADLINK achieves the second best results after TuckER for FB15k-237 on MRR, Hits@1, and Hits@3 metrics, whereas for Hits@10, TuckER performs slightly better with an improvement of 1.5% over MADLINK. On the other

Table 6
Comparison of MADLINK results with the structure-based baseline models on FB15k, WN18, and YAGO3-10 datasets.

FB15K				
Models	MRR	Hits@1	Hits@3	Hits@10
TransE	0.63	0.5	0.73	0.85
DistMult	0.432	0.302	0.498	0.68
ConvE	0.5	0.42	0.52	0.66
ConvKB	0.65	0.55	0.71	0.82
RotatE	**0.797**	-	-	0.884
HypER	0.790	0.734	0.829	0.885
R-GCN	0.69	0.6	0.72	0.8
QuatE	0.77	0.7	0.821	0.878
MDE	0.652	-	-	0.857
TuckER	0.795	**0.741**	**0.833**	**0.892**
MADLINK[1]	0.712	0.722	0.788	0.81
MADLINK[2]	0.69	0.714	0.78	0.798
WN18				
Models	MRR	Hits@1	Hits@3	Hits@10
TransE	0.66	0.44	0.88	0.95
DistMult	0.755	0.615	0.891	0.94
ConvE	0.93	0.91	0.94	0.95
RotatE	0.949	-	-	0.959
HypER	0.951	0.947	**0.955**	0.958
R-GCN	0.71	0.61	0.88	0.932
QuatE	0.949	0.941	0.954	0.96
MDE	0.871	-	-	0.956
TuckER	**0.953**	**0.949**	**0.955**	0.958
MADLINK[1]	0.95	0.898	0.911	**0.96**
MADLINK[2]	0.944	0.88	0.9	0.9
YAGO3-10				
Models	MRR	Hits@1	Hits@3	Hits@10
TransE	0.51	0.41	0.57	0.67
DistMult	0.354	0.262	0.4	0.537
ConvE	0.4	0.33	0.42	0.53
ConvKB	0.3	0.21	0.34	0.5
HypER	0.533	0.455	**0.58**	0.678
R-GCN	0.12	0.06	0.113	0.211
TuckER	0.427	0.331	0.476	**0.609**
MADLINK[1]	**0.538**	**0.457**	**0.58**	**0.68**
MADLINK[2]	0.528	0.447	0.573	0.67

hand, for WN18RR, MADLINK achieves the second best result for MRR, Hits@1, and Hits@3 after TuckER. The latter performs marginally better than MADLINK with an improvement of 0.7% on MRR, 0.5% on Hits@1, 0.3% on Hits@3. However, MADLINK performs better than TuckER for Hits@10 by 2.3%. RotatE performs the best for Hits@10 amongst the mentioned baseline models and its performance is 2.2% better than MADLINK.

TuckER model outperforms all the baseline models as well as MADLINK for FB15k. However, for the YAGO3-10 dataset, MADLINK outperforms the TuckER model with an improvement of 11.1% on MRR, 12.6% on Hits@1, 10.4% on Hits@3, and 7.1% on Hits@10. Additionally, MADLINK outperforms all the other baseline models and achieves SOTA results for the YAGO3-10 dataset over all the metrics. For WN18, TuckER performs better than all the baseline models and MADLINK for all the metrics except for Hits@10.

Table 7
Comparison of MADLINK with LiteralE on FB15k-237, FB15k, and YAGO3-10

FB15K-237				
Models	MRR	Hits@1	Hits@3	Hits@10
LiteralE (Numeric+Text)	0.32	0.234	-	0.488
LiteralE (Numeric)	0.317	0.232	0.348	0.483
MADLINK	**0.347**	**0.252**	**0.38**	**0.529**
FB15k				
Models	MRR	Hits@1	Hits@3	Hits@10
LiteralE (Numeric)	0.676	0.589	0.733	**0.825**
MADLINK	**0.712**	**0.722**	**0.788**	0.81
YAGO3-10				
Models	MRR	Hits@1	Hits@3	Hits@10
LiteralE (Numeric)	0.479	0.4	0.525	0.627
MADLINK	**0.538**	**0.457**	**0.580**	**0.68**

Additionally, MADLINK outperforms its base model DistMult for all the metrics in all the datasets. For FB15K-237, there is an improvement of 10% in MRR, 9.1%, 11%, and 10.3% in Hits@1, Hit@3, and Hits@10 respectively. Similarly, for FB15K, a considerable increase of 28% in MRR, 42% in Hits@1, 29% in Hits@3, and 13% in Hits@10 have been achieved. On the other hand, for WN18, MADLINK shows a rise of 19.5% in MRR, 28.3% in Hits@1, and 2% for both Hits@3 and Hits@10. Also, for WN18RR similar increment of the results has been achieved with an increment of 3.9% in MRR, 1.4% in Hits@1, 3.7% in Hits@3, and 4.5% in Hits@10. Identical improvement has also been obtained for the YAGO3-10 dataset with an improvement of an average of 17.55% overall the evaluation metrics with an increase of 18.4% in MRR, 19.5% in Hits@1, 18%, and 14.3% in Hits@3, and Hits@10 respectively.

Also, from the results of Hits@k, for all the datasets it can be inferred that MADLINK correctly ranks many true triples in top-k as it achieves SOTA results for FB15K-237, WN18RR, WN18, and YAGO3-10 whereas comparable results for FB15K. Furthermore, MADLINK works better for the datasets without the reverse relations such as FB15K-237 and WN18RR as compared to FB15K and WN18 because in this work directed paths are considered and not undirected edges. However, in this work, the evaluation metric Mean Rank (MR) has not been used because it is sensitive to outliers as mentioned in other related work [63].

MADLINK has also been compared with LiteralE [41], which uses numerical literal to predict the missing links. The results shown in Table 7 illustrate that MADLINK performs better than LiteralE. Additionally for FB15k-237, MADLINK performs better than LiteralE variant with both numeric and text data. It is to be mentioned that, the results of DistMult variant of LiteralE is considered in Table 7 for a fair comparison with MADLINK as both of them use DistMult.

The main advantage of MADLINK over the structure-based baseline models is that link prediction can be performed for unpopular entities in a KG, i.e., the entities without any relations or less number of relations associated with them. This is because MADLINK considers the textual entity descriptions of the entities apart from the structural information. Similarly, since it considers the structural information of the entities in the forms of paths, therefore, missing links can be predicted for entities having triple information but without textual entity descriptions.

Therefore, it can be concluded that the path information of the entities when coupled with the textual entity descriptions in KGs provide better results in link prediction which is further analysed in Section 5.6.

Table 8
Triple Classification (Accuracy in %)

Models	FB15K	FB15K-237	WN18	WN18RR
TransE	82.9	75.6	87.6	74
DistMult	-	73.9	-	80.4
ConvE	87.3	78.2	95.4	78.3
ConvKB	87.9	80.1	96.4	79.1
Jointly(ALSTM)	91.5	-	97.8	-
PConvKB	89.5	82.1	97.6	80.3
MADLINK	**92.1**	**82.8**	**98**	**81.2**

5.5. Triple Classification

Triple Classification is the task of determining if a given triple is correct or not. It is a binary classification task, where a given triple (e_h, r, e_t) is to be classified into either 0 (false) or 1 (true) as proposed by [28]. Since all the triples in the training set are true, negative triples are generated for this task, by replacing the head and the tail entities. Also, using this negative sampling method, some of the triples would be generated which would be true. Therefore, all the generated negative triples which are present in the training, test, and validation set are removed. As mentioned by the authors, a threshold ρ_r is set for triple classification maximizing the classification accuracy on the validation set. A triple is considered as positive if the conditional probability, $P(e_t|e_h, r) \geqslant \rho_r$ [28] holds.

However, for MADLINK a Convolutional Neural Network (CNN) binary classifier has been used on top of the embeddings of entities and relations obtained from the model. The classifier is trained with positive triples from the training set and negative triples obtained from the negative sampling model. The test set is also complemented with negative examples for proper evaluation. Triple Classification for all 4 datasets has also been compared against all the above mentioned SOTA models along with PConvKB [33]. PConvKB is an embedding model that incorporates relation paths locally and globally which are then passed through a convolutional neural model. The results are depicted in Table 8. The proposed model achieves the SOTA results with an improvement of 0.2% to 0.8% over the SOTA models for all the benchmark datasets. Furthermore, the accuracy of triple classification for the YAGO3-10 dataset is 80.1% but it has not been provided in Table 8 because of the lack of results from the SOTA models.

5.6. Ablation Study

This section discusses the analysis of different features considered in the MADLINK model for the task of link prediction.

Impact of Only Textual Entity Description.

The impact of only the textual entity description without the structural information for the task of link prediction has been evaluated along with the triples. The latent representation of the textual entity descriptions is obtained using SBERT vectors. The results as depicted in Table 9 show that it outperforms the base model DistMult for all the datasets. It is observed that the improvement for WN18 and WN18RR is very small compared to the other datasets. This is due to the fact that both the datasets contain 5780 entities for which the length of the textual entity description is less than or equal to five providing much less information. However, Table 4 shows textual entity descriptions together with path information exhibits considerable improvement in link prediction.

Impact of Only Structural Information.

The results depicted in Table 10 illustrate the impact of using only the structural information of the KGs in form of paths. The number of paths increases exponentially with the number of hops, for e.g., in FB15k, the average neighbour for each node is 30, therefore, the total number of possible paths of 4 hops would be 810,000. PF-ITF is used to filter out the uncommon relations which in turn reduces the number of paths. As mentioned earlier, 1000 paths with 4 hops are selected for each entity because the relevant contextual information w.r.t. the starting node decreases with more hops. Also, when the walk reaches a dead end, i.e., a node without any outgoing edges, the walk ends in that dead-end node, even if the maximum hops is not reached. However, there could be more than 1000

Table 9
Impact of Textual Entity Descriptions in MADLINK (without path information)

Datasets	Models	MRR	Hits@1	Hits@3	Hits@10
FB15K	DistMult	0.432	0.302	0.498	0.68
	MADLINK	**0.481**	**0.348**	**0.512**	**0.692**
FB15K-237	DistMult	0.2471	0.161	0.271	0.426
	MADLINK	**0.249**	**0.179**	**0.279**	**0.431**
WN18	DistMult	0.755	0.615	0.891	0.94
	MADLINK	**0.758**	**0.618**	**0.895**	**0.943**
WN18RR	DistMult	0.438	0.424	0.442	0.478
	MADLINK	**0.439**	**0.425**	**0.448**	**0.48**
YAGO3-10	DistMult	0.354	0.262	0.4	0.537
	MADLINK	**0.361**	**0.267**	**0.411**	**0.54**

Table 10
Impact of Structural Information in MADLINK (without textual entity description)

Datasets	Models	MRR	Hits@1	Hits@3	Hits@10
FB15K	DistMult	0.432	0.302	0.498	0.68
	MADLINK	**0.477**	**0.328**	**0.498**	**0.682**
FB15K-237	DistMult	0.2471	0.161	0.271	0.426
	MADLINK	**0.249**	**0.169**	**0.273**	**0.426**
WN18	DistMult	0.755	0.615	0.891	0.94
	MADLINK	**0.758**	**0.63**	**0.898**	**0.947**
WN18RR	DistMult	0.438	0.424	0.442	0.478
	MADLINK	**0.44**	**0.426**	**0.45**	**0.482**
YAGO3-10	DistMult	0.354	0.262	0.4	0.537
	MADLINK	**0.365**	**0.262**	**0.42**	**0.542**

paths starting from a certain node in which the first 3 hops consist of the same entities and relations. But this does not provide any meaningful insight into the source code. Therefore, amongst the paths, we restrict the paths with the same sequence to a maximum of 30. For example, any path starting with this $e_1 \xrightarrow{r_1} e2 \xrightarrow{r_2} e_3 \xrightarrow{r_2} e_4$ can occur maximum of 30 times amongst the 1000 paths generated from node e_1. If the number of paths is less than 1000 for any entity, then all paths for that entity are considered.

The results in Table 10 show that the MADLINK model with only the structural information outperforms the base model DistMult for all the metrics across all the 5 benchmark datasets. Additionally, MADLINK with only structural information works slightly better than MADLINK with only textual entity descriptions for WN18 and WN18RR datasets. Therefore, it can be inferred that the neighbourhood information is well captured for these two datasets in the paths.

Influence of Attention in the network

To analyse the impact of the attention mechanism in encoding the path vectors, experiments have been conducted without the attention layer as depicted in Table 11. The result depicts that there is an improvement in all the evaluation metrics for all the datasets if MADLINK is used with the attention mechanism. Therefore, with an improvement of an average of 5% over Hits@10 for FB15K, WN18, and YAGO3-10 as well as 3.1% for FB15K-237 and 1.4% for WN18RR, it can be seen that the attention mechanism helps in identifying the important entities and relations in a path for the task of link prediction.

Table 11
Impact of Attention Mechanism in MADLINK

Datasets	Models	MRR	Hits@1	Hits@3	Hits@10
FB15K	MADLINK (w/o Attn.)	0.48	0.388	0.502	0.701
	MADLINK (with Attn.)	**0.51**	**0.412**	**0.591**	**0.758**
FB15K-237	MADLINK (w/o Attn.)	0.331	0.211	0.35	0.498
	MADLINK (with Attn.)	**0.347**	**0.252**	**0.38**	**0.529**
WN18	MADLINK (w/o Attn.)	0.92	0.822	0.85	0.91
	MADLINK (with Attn.)	**0.95**	**0.898**	**0.911**	**0.96**
WN18RR	MADLINK (w/o Attn.)	0.412	0.401	0.411	0.509
	MADLINK (with Attn.)	**0.477**	**0.438**	**0.479**	**0.549**
YAGO3-10	MADLINK (w/o Attn.)	0.411	0.331	0.524	0.623
	MADLINK (with Attn.)	**0.461**	**0.372**	**0.580**	**0.68**

6. Conclusion and Future Work

In this paper, a novel approach has been proposed for combining the contextual structural information of an entity from the KGs as well as textual entity descriptions in the embedding space to address the problem of KG completion using link prediction and triple classification. Moreover, an attention-based encoder-decoder approach is introduced to measure the importance of paths. Experimental results show that MADLINK achieves the SOTA results for the textual entity description-based embedding models for the link prediction task on all the 5 benchmark datasets. Furthermore, MADLINK outperforms most of the baseline models whereas it achieves comparable results with the rest. In this paper, two major research questions are formulated and presented in Section 3. The answers to these questions are given as follows:

- *RQ1: Does the contextual information of entities and relations in a KG help in the task of link prediction?*
  - The contextual information of the entities and the relations in a KG are captured by generating paths using random walks. Also, the attention mechanism on the encoder-decoder model helps in identifying the important entities within a path. Handling the path information separately (as shown in Table 10) in the MADLINK model yields better results than the base model DistMult which uses only the triple information.
- *RQ2: What is the impact of incorporating textual entity descriptions in a KG for the task of link prediction?*
  - The latent representations of the textual entity descriptions are generated using the SBERT model. The impact of the textual entity descriptions in the link prediction task is dependent on the length of textual information available for the corresponding entities. It can be observed from the results of MADLINK as depicted in Table 9 that link prediction works better for FB15k and FB15k-237 compared to WN18 and WN18RR. This is because Freebase entities have detailed and longer text descriptions than WordNet entities. Also, only the textual description-based variant of MADLINK outperforms the base model DistMult for all the 5 benchmark datasets. Therefore, the textual entity descriptions play an important role in the task of link prediction in KGs.

The obtained results suggest that the impact of the textual entity description and the contextual structural information is different for different KGs. However, the combination of contextual structural information together with the textual entity descriptions in the MADLINK model outperforms all the text-based KG embedding models.

In future work the following research directions will be considered to further improve the model:

- Explore the translational embedding models such as TransR to learn the initial embeddings of the entities and relations.
- Explore the different scoring functions such as ConvE, translational models, etc. for the base model to analyze the embeddings for the link prediction task.
- Use different multi-hop strategies to generate the context information.
- Multiple text literals available for the entities in the KGs as labels, summary, comments, etc. can also be incorporated in the model. Also for relations, relation name labels can be considered as the textual description.
- Include explicit external text information such as from Wikipedia into the model.

References

[1] J. Hoffart, M.A. Yosef and I.B. et al., Robust Disambiguation of Named Entities in Text, in: *Proceedings of the 2011 Conf. on Empirical Methods in Natural Language Processing*, 2011.

[2] A. Bordes, S. Chopra and J. Weston, Question Answering with Subgraph Embeddings, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[3] F. Zhang, N.J. Yuan, D. Lian, X. Xie and W.-Y. Ma, Collaborative Knowledge Base Embedding for Recommender Systems, in: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[4] P. Zhao, C. Aggarwal and G. He, Link prediction in graph streams, in: *Proceedings of the IEEE 32nd International Conference on Data Engineering*, 2016.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating embeddings for modeling multi-relational data, *Proceedings of the Advances in neural information processing systems* (2013).

[6] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

[7] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing*, 2015.

[8] G. Ji, K. Liu, S. He and J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: *Proceedings of the Thirtieth AAAI conference on artificial intelligence*, 2016.

[9] Y. Jia, Y. Wang, H. Lin, X. Jin and X. Cheng, Locally Adaptive Translation for Knowledge Graph Embedding, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[10] J. Feng, M. Huang, Y. Yang and X. Zhu, GAKE: Graph Aware Knowledge Embedding, in: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

[11] P. Ristoski and H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: *Proceedings of the International Semantic Web Conference*, Springer, 2016.

[12] M. Chen, Y. Tian, K.-W. Chang, S. Skiena and C. Zaniolo, Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-Lingual Entity Alignment, *arXiv preprint arXiv:1806.06478* (2018).

[13] Y. Wu and Z. Wang, Knowledge Graph Embedding with Numeric Attributes of Entities, in: *Proceedings of the Rep4NLP@ACL*, 2018.

[14] A. García-Durán and M. Niepert, Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features, *arXiv preprint arXiv:1709.04676* (2017).

[15] P. Pezeshkpour, L. Chen and S. Singh, Embedding Multimodal Relational Data for Knowledge Base Completion, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[16] R. Xie, Z. Liu, M. Sun et al., Representation Learning of Knowledge Graphs with Hierarchical Types., in: *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016.

[17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, Dbpedia: A nucleus for a web of open data, in: *The semantic web*, 2007.

[18] I. Sutskever, O. Vinyals and Q.V. Le, Sequence to sequence learning with neural networks, in: *Proceedings of the Advances in neural information processing systems*, 2014.

[19] N. Reimers and I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 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*, 2019.

[20] B. Yang, W. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[21] K. Toutanova and D. Chen, Observed versus latent features for knowledge base and text inference, in: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 2015.

[22] T. Dettmers, M. Pasquale, S. Pontus and S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, 2018.

[23] Z. Sun, Z. Deng, J. Nie and J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, in: *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[24] A. Bordes, J. Weston, R. Collobert and Y. Bengio, Learning structured embeddings of knowledge bases, in: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
[25] M. Nickel, V. Tresp and H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: *Proceedings of the International Conference on Machine Learning*, 2011.
[26] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex embeddings for simple link prediction, in: *Proceedings of the International conference on machine learning*, 2016.
[27] M. Nickel, L. Rosasco and T. Poggio, Holographic embeddings of knowledge graphs, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.
[28] R. Socher, D. Chen, C.D. Manning and A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: *Proceedings of the Advances in neural information processing systems*, 2013.
[29] T.D.N. Dai Quoc Nguyen, D.Q. Nguyen and D. Phung, A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.
[30] I. Balažević, C. Allen and T.M. Hospedales, Hypernetwork knowledge graph embeddings, in: *Proceedings of the International Conference on Artificial Neural Networks*, 2019.
[31] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov and M. Welling, Modeling relational data with graph convolutional networks, in: *Proceedings of the European semantic web conference*, 2018.
[32] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao and S. Liu, Modeling Relation Paths for Representation Learning of Knowledge Bases, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.
[33] N. Jia, X. Cheng and S. Su, Improving Knowledge Graph Embedding Using Locally and Globally Attentive Relation Paths, in: *European Conference on Information Retrieval*, 2020.
[34] S. Guo, Q. Wang, B. Wang, L. Wang and L. Guo, SSE: Semantically Smooth Embedding for Knowledge Graphs, *IEEE Transactions on Knowledge and Data Engineering* (2017).
[35] Z. Zhang, J. Cai, Y. Zhang and J. Wang, Learning hierarchy-aware knowledge graph embeddings for link prediction, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
[36] J. Xu, X. Qiu, K. Chen and X. Huang, Knowledge graph representation with jointly structural and textual encoding, in: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.
[37] H. Xiao, M. Huang, L. Meng and X. Zhu, SSP: semantic space projection for knowledge graph embedding with text descriptions, in: *Proceedings of the Thirty-First AAAI conference on artificial intelligence*, 2017.
[38] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for knowledge graph completion, *arXiv preprint arXiv:1909.03193* (2019).
[39] B. Kim, T. Hong, Y. Ko and J. Seo, Multi-task learning for knowledge graph completion with pre-trained language models, in: *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.
[40] D. Daza, M. Cochez and P. Groth, Inductive Entity Representations from Text via Link Prediction, in: *Proceedings of the Web Conference 2021*, 2021.
[41] A. Kristiadi, M.A. Khan, D. Lukovnikov, J. Lehmann and A. Fischer, Incorporating literals into knowledge graph embeddings, in: *Proceedings of the International Semantic Web Conference*, 2019.
[42] Y. Tay, L.A. Tuan, M.C. Phan and S.C. Hui, Multi-task neural network for non-discrete attribute prediction in knowledge graphs, in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.
[43] R. Xie, Z. Liu, H. Luan and M. Sun, Image-embodied knowledge representation learning, in: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.
[44] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: *Proceedings of the 29th AAAI conference on artificial intelligence*, 2015.
[45] G.A. Gesese, R. Biswas, M. Alam and H. Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, *Semantic Web* (2021).
[46] G. Pirrò, Explaining and Suggesting Relatedness in Knowledge Graphs, in: *Proceedings of the 14th International Semantic Web Conference*.
[47] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient Estimation of Word Representations in Vector Space, *CoRR* (2013).
[48] J. Pennington, R. Socher and C.D. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
[49] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
[50] Y. Qiao, C. Xiong, Z. Liu and Z. Liu, Understanding the Behaviors of BERT in Ranking, *arXiv preprint arXiv:1904.07531* (2019).
[51] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov and Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, *Advances in neural information processing systems* (2019).
[52] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.
[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention is All you Need, in: *Proceedings of the Advances in Neural Information Processing Systems*, 2017.

[54] K.D. Bollacker, R.P. Cook and P. Tufts, Freebase: A Shared Database of Structured General Human Knowledge, in: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 2007.
[55] G.A. Miller, *WordNet: An electronic lexical database*, MIT press, 1998.
[56] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, in: *Proceedings of the 16th international conference on World Wide Web*, 2007.
[57] A. Rossi, D. Barbosa, D. Firmani, A. Matinata and P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2021).
[58] L. Costabello, S. Pai, C.L. Van, R. McGrath, N. McCarthy and P. Tabacof, AmpliGraph: a Library for Representation Learning on Knowledge Graphs, 2019.
[59] T.D. Nguyen, D.Q. Nguyen, D. Phung et al., A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*, 2018.
[60] S. ZHANG, Y. Tay, L. Yao and Q. Liu, Quaternion Knowledge Graph Embeddings, 2019.
[61] A. Sadeghi, D. Graux, H. Shariat Yazdi and J. Lehmann, MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs, in: *Proceedings of the European Conference on Artificial Intelligence*, 2020.
[62] I. Balažević, C. Allen and T. Hospedales, TuckER: Tensor Factorization for Knowledge Graph Completion, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
[63] M. Nickel, L. Rosasco and T.A. Poggio, Holographic Embeddings of Knowledge Graphs, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[64] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng and P. Wang, K-BERT: Enabling language representation with knowledge graph, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
[65] Y. Wu and Z. Wang, Knowledge graph embedding with numeric attributes of entities, in: *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018.
[66] P. Pezeshkpour, L. Chen and S. Singh, Embedding Multimodal Relational Data for Knowledge Base Completion, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
[67] 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 ACM SIGMOD international conference on Management of data*, 2008.
[68] R. Xie, Z. Liu, T.-S. Chua, H.-B. Luan and M. Sun, Image-embodied Knowledge Representation Learning, in: *IJCAI*, 2017.
[69] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu and X. Zhu, Commonsense Knowledge Aware Conversation Generation with Graph Attention, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, J. Lang, ed., 2018.
[70] J. Weston, A. Bordes, O. Yakhnenko and N. Usunier, Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction, in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.*, 2013.
[71] R. Blanco, P. Mika and S. Vigna, Effective and Efficient Entity Search in RDF Data, in: *Proceedings of the 10th International Semantic Web Conference*, 2011.
[72] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in: *Proceedings of the 33nd International Conference on Machine Learning*, JMLR Workshop and Conference Proceedings, 2016.
[73] R. Wang, B. Li, S. Hu, W. Du and M. Zhang, Knowledge Graph Embedding via Graph Attenuated Attention Networks, *IEEE Access* (2019).
[74] Y. Dai, S. Wang, N.N. Xiong and W. Guo, A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks, *Electronics* (2020).
[75] A. Bordes, X. Glorot, J. Weston and Y. Bengio, Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing, in: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2012.
[76] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Transactions on Knowledge and Data Engineering* (2017).
[77] M. Nickel, V. Tresp and H.-P. Kriegel, Factorizing Yago: Scalable Machine Learning for Linked Data, in: *Proceedings of the 21st international conference on World Wide Web*, 2012.