IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2022, accepted 6 July 2022, date of publication 18 July 2022, date of current version 25 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3191666

RESEARCH ARTICLE

Language Model Guided Knowledge Graph Embeddings

MIRZA MOHTASHIM ALAM[1], MD RASHAD AL HASAN RONY[2,3], MOJTABA NAYYERI[3], KARISHMA MOHIUDDIN[3], M. S. T. MAHFUJA AKTER[3], SAHAR VAHDATI[1], AND JENS LEHMANN[1,4], (Member, IEEE)

[1]Institute for Applied Informatics Association e.V. (InfAI), 01069 Dresden, Germany
[2]Fraunhofer IAIS, 01069 Dresden, Germany
[3]Institute of Computer Science, University of Bonn, 53113 Bonn, Germany
[4]Amazon, 01097 Dresden, Germany

Corresponding author: Mirza Mohtashim Alam (turzo.mohtasim@gmail.com)

This work was supported in part by the European Union (EU) through the Project TAILOR under Grant EU GA 952215, through the Project CALLISTO under Grant 101004152, and through the Project e-Vita under Grant GA 101016453; in part by the Bundesministerium für Bildung und Forschung (BMBF) through the Project MLwin under Grant 01IS18050, through the Project SPEAKER under Grant BMWi FKZ 01MK20011A, and through the Project JOSEPH (Fraunhofer Zukunftsstiftung); and in part by the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) under Grant IS18026A-F.

ABSTRACT Knowledge graph embedding models have become a popular approach for knowledge graph completion through predicting the plausibility of (potential) triples. This is performed by transforming the entities and relations of the knowledge graph into an embedding space. However, knowledge graphs often include further textual information stored in literal, which is ignored by such embedding models. As a consequence, the learning process stays limited to the structure and the connections between the entities, which has the potential to negatively influence the performance. We bridge this gap by leveraging the capabilities of pre-trained language models to include textual knowledge in the learning process of embedding models. This is achieved by introducing a new loss function that guides embedding models in measuring the likelihood of triples by taking such complementary knowledge into consideration. The proposed solution is a model-independent loss function that can be plugged into any knowledge graph embedding model. In this paper, Sentence-BERT and fastText are used as pre-trained language models from which the embeddings of the textual knowledge are obtained and injected into the loss function. The loss function contains a trainable slack variable that determines the degree to which the language models influence the plausibility of triples. Our experimental evaluation on six benchmarks, namely Nations, UMLS, WordNet, and three versions of CodEx confirms the advantage of using pre-trained language models for boosting the accuracy of knowledge graph embedding models. We showcase this by performing evaluations on top of the five well-known knowledge graph embedding models such as TransE, RotatE, ComplEx, DistMult, and QuatE. The results show an improvement in accuracy up to 9% on UMLS dataset for the Distmult model and 4.2% on the Nations dataset for the ComplEx model when they are guided by pre-trained language models. We additionally studied the effect of multiple factors such as the structure of the knowledge graphs and training steps and presented them as ablation studies.

INDEX TERMS Knowledge graph, knowledge graph embeddings, language models, link prediction.

I. INTRODUCTION

During the last decade, the rise of knowledge graphs (KGs) significantly impacted several machine learning approaches and downstream AI tasks such as question answering, prediction, and recommendation systems [1], [2]. KGs include a multi-relational representation of factual knowledge as triples in the form of (*subject*, *predicate*, *object*), e.g., (*LouisArmstrong*, *occupation*, *Singer*). Even though current KGs are quantitatively large-scale, containing millions of

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

triples, they remain highly incomplete and usually do not capture all relevant knowledge for a domain of interest. Various link prediction approaches have been proposed to tackle the incompleteness of KGs, among which link prediction using knowledge graph embeddings (KGE) has become popular for KG completion tasks.

KGEs typically receive a knowledge graph as a set of correct triples and transfer entities and relations from their symbolic representation to an embedding space (often vectors). In the learning process, the existing triples in the KG are used as positive samples. Additionally, negative samples are generated from positive ones, usually by random corruption techniques. The learning process is performed by employing a score function to compute the plausibility of potentially correct triples. Furthermore, a loss function is used to adjust the randomly initialized embeddings in a way that positive samples get higher scores than negative ones.

Although the above-described procedure works well when aiming to preserve the structural aspects of a KG, the literals in a knowledge graph that contain complementary textual knowledge remain unused. The left side of Figure 1 illustrates a KG where entities and relations contain textual complementary knowledge. The right side of the figure compares to the way typical input is taken by standard KGE models, where only symbolic representation (structure) is considered. However, the performance of KGEs models is influenced when there is a lack of enough structural knowledge in a KG [3]. In such cases, the complementary knowledge from textual descriptions can mitigate the problem in order to enhance the performance of the link prediction task. As an example, there is a lack of structure around the triple (*Louis Armstrong*, *occupation*, *Singer*) shown in Figure 1. As can be seen, further contextual information about the subject **Louis Armstrong** is available (*American Jazz Trumpeter, Composer, and Singer*). Although the entity **Composer** is structurally in the KG, the link is missing to Louis Armstrong. However, the connection can be found from the textual information of entity **Louis Armstrong** if the KGE model is able to perceive it. In the ceiling performances reported by state-of-the-art KGE-based approaches [4], [5], this textual information is not considered by almost any of such models [6]. In order to bridge this gap, we incorporate complementary knowledge into the learning process of KGE models with a unique use of language models. Several methods providing embedding of textual data can facilitate this process, among which are the recent Transformer-based [7] pre-trained language models (PLMs) such as BERT [8], RoBERTa [9], and GPT-2 [10].

In this work, we propose a novel combination of Knowledge Graph Embedding models and pre-trained language models through a unified loss function. This is done for the purpose of utilizing the embedding of textual knowledge in the learning process of KGEs. Through a systematic analysis, we selected fastText [11] to obtain the embeddings in the sub-word or character level and Sentence-BERT [12] to encode the sentence level descriptions. In our approach, the primary score of each triple is calculated from the KGE models. In addition, an auxiliary term is assigned as the plausibility of the same triple obtained from PLMs considering the available textual information. Later, these two criteria are transformed into a likelihood estimation. In this way, we enforce the upper bound for the score of positive samples and add a margin between positive and negative samples. In addition, a confidence function that represents the plausibility of triples is taken into consideration from the language model. This changes the boundary based on structural and textual information and affects the triple prediction performance of the underlying model. In order to optimize the embeddings based on KG and PLM, a log-likelihood loss is computed and maximized for the guidance of the baseline KGE model. The results emphasize the effectiveness of our proposed approach as well as the improved performance of KGE models in the link prediction task.

In summary, our main contributions are:

- We addressed the problem of knowledge graph embedding models ignoring complementary knowledge.
- The gap between knowledge graph embedding models and language models is bridged by a unique approach which performs the inclusion of PLMs in a loss function.
- A novel model-free loss function is proposed that considers embedding of complementary textual knowledge.
- The standard benchmark datasets are adapted to be used for Knowledge Graph Embeddings in the presence of Language Models at once, which is also usable for other similar works.
- An extensive evaluation is performed to study the effect of considering textual knowledge by embedding models on six benchmark datasets and five known knowledge graph embedding models.
- Evaluations are extended on the impact of using PLMs by conducting several studies, including: a) inclusion of both structural and textual knowledge by considering vectors obtained from the PLMs and KGE models into the proposed loss function; b) inclusion of only textual knowledge by considering the vectors obtained by PLMs in the score function of KGE; c) comparison of similarities and differences of the vectors corresponding to the structural and textual knowledge of the KGs.

The rest of the paper is organized as follows. In Section II, we describe notations and required information for understanding the proposed methodology. In Section III, we review the literature on current embedding models which utilize the PLMs in their learning process. Section IV provides the details of the proposed method and the learning process. In section V, we provide the details about the experimental setup and analysis of the obtained results. The current section also contains the ablation studies to support the analysis of the results in subsection V-D. Finally, in Section VI, we summarise the main conclusions and outline future directions.

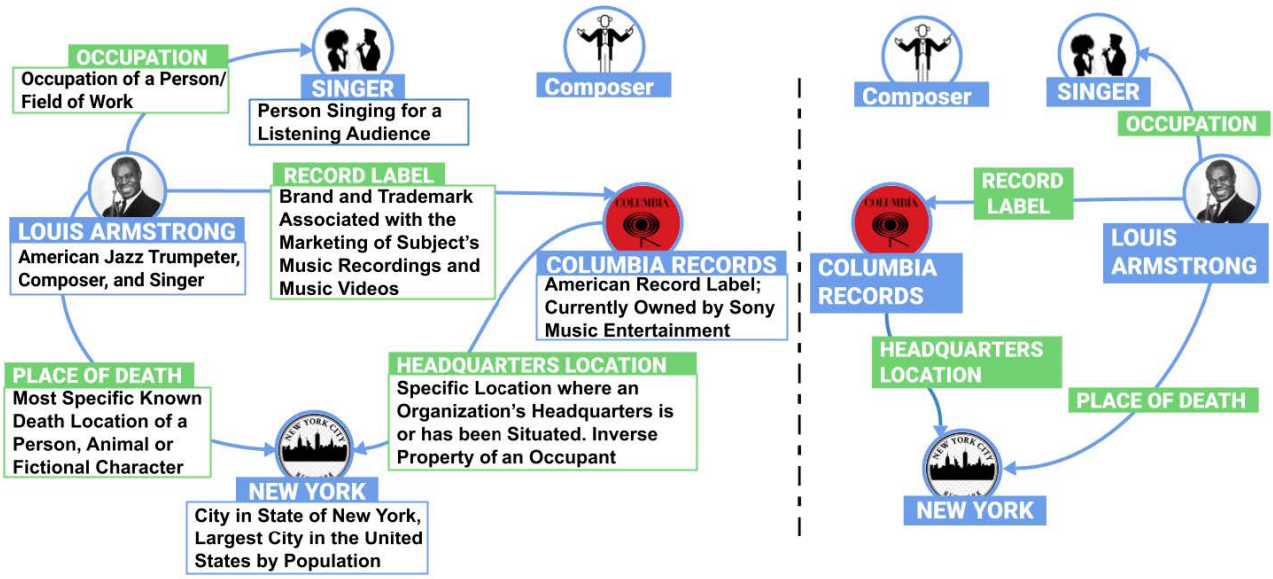


FIGURE 1. KG with/without textual knowledge. This figure shows a comparison of a KG with textual descriptions for entities and relations (left side of the figure) and the only symbolic KG (right side of the figure).

II. PRELIMINARIES

In this section, we provide the preliminaries which are required to understand the methodology of this work. Here, we introduced the core concepts of Knowledge Graph Embedding Models, such as embedding vectors, score function, and loss function. Similarly, the concepts related to the pre-trained language model are discussed as well. In both of the cases, we have introduced the notations which will be used throughout the whole paper.

A. KNOWLEDGE GRAPH

For a set of entities $\mathcal{E}$, and relations $\mathcal{R}$; a Knowledge Graph ($\mathcal{K}$) is a formalism for the triple-based representation of facts shown as $\mathcal{K} = \{(s, p, o)|s, o \in \mathcal{E}, p \in \mathcal{R}\}$ where (s), (p), and (o) refer to the subject, prediction, and object respectively.

B. KNOWLEDGE GRAPH EMBEDDING

In this part, we introduce the core components of KGEs, namely embedding vectors, score, and loss functions.

1) EMBEDDING VECTORS

A $\mathcal{KGE}$ model is a mapping function as $\phi_{KG} : \mathcal{E}/\mathcal{R} \rightarrow \mathbf{E}/\mathbf{R}$ that transfers the symbolic representation of entities in $\mathcal{E}$ and relations $\mathcal{R}$ of a KG into a d dimensional latent feature space. For triples of a KG, a KGE model is defined as $\mathcal{KGE} = \{(\mathbf{s}, \mathbf{p}, \mathbf{o})|\mathbf{s}, \mathbf{o} \in \mathbf{E}, \mathbf{p} \in \mathbf{R}, (s, p, o) \in \mathcal{K}\}$.

2) SCORE FUNCTION

Each KGE model defines a score function $f(s, p, o)$ in which the input is a triple (s, p, o) and the output is a value showing the degree to which the triple is plausible. Usually, a higher value for the score shows a triple to be more plausible.

3) LOSS FUNCTION

The learning process in a KGE starts with random initialization of the embedding vectors. Therefore, the scores of the triples for positive and negative samples are also random. Optimization of a loss function L (often Stochastic Gradient Descent) leads to a better adjustment of the embeddings such that positive samples get higher scores than the negative ones. As the prediction of the triples gets better with each iteration, the loss decreases.

C. PRE-TRAINED LANGUAGE MODEL

Language models are trained on large corpora to generate word tokens. The probability of generating a sentence S is computed by $p(S) = \prod_{i=1}^{|S|} p(s_i|s_{<i})$, where s_i is the token generated at time step i. The trained checkpoint (PLM) contains all the learned weights that we leverage to obtain the embedding representation of the text. Let the PLM function be $\mathcal{PLM}$, which takes a text as an input and returns a learned vector representation as the output. The corresponding text of subject and object entities as well as the predicate can be the input of the function, and as an output $\mathcal{PLM}$ returns the $1 \times d_{LM}$ dimensional vector for each s, p, o. $\mathcal{PLM} : \mathbf{s}^{text} \rightarrow \mathbf{s}^{LM} \in \mathbb{R}^{1 \times d_{LM}}$ (this example represents for subject only, same applies for the object and relation). We denote the embedding of textual descriptions corresponding to the entities and relations of a KG obtained from a pre-trained language model by $\mathbf{s}^{LM}, \mathbf{p}^{LM}, \mathbf{o}^{LM}$. Similar to $KGEs$, this is also a mapping function that transfers text into a d_{LM} dimensional latent space which we denote by $\phi_{LM} : \mathcal{E}/\mathcal{R} \rightarrow \mathbf{E}^{LM}/\mathbf{R}^{LM}$. The difference between the functions ϕ_{KG} and ϕ_{LM} is that the embeddings of the $PLMs$ are not learned. Instead, they are considered as a guide in the learning process of KGE models.

TABLE 1. Score functions of KGEs.

KGE model	Score function
TransE [15]	$\parallel s + p - o \parallel$
RotatE [22]	$\parallel s \circ p - o \parallel$
DistMult [18]	$\parallel s * p * o \parallel$
ComplEx [19]	$Re < s, p, \bar{o} >$
QuatE [21]	$Q'_s \cdot Q_o$

III. RELATED WORK

In this paper, we aim to utilize KG and text embedded by language models to enhance the performance of KGEs in link prediction tasks. Therefore, in this section, we review three categories of embedding models: standard knowledge graph embedding, text-enhanced knowledge graph embedding, and pre-trained language models. The selection of the models to be covered in the related work as well as the evaluations follows the best practices of the embedding models [1], [4], [13]. Same for the choice of pre-trained language models, we selected a handful list of models that are aligned with our work in terms of their methodology in capturing textual patterns [14].

A. KNOWLEDGE GRAPH EMBEDDING MODELS

Generally, KGE models can be classified into two groups [1] based on the design of their score function: translational distance and semantic matching-based model. Here we name several state-of-the-art KGEs that we also use in our evaluation. TransE [15] and the family of its follow-up models (e.g., TransH [16] and TransR [17]) are examples of translational-distance groups. The score function of these models is designed in a way that they encode entities as vectors and relations between them as translation vectors. DistMult [18] is a semantic matching model that uses a predicate-specific matrix in order to capture the pairwise interaction between the subjects and objects [1]. ComplEx [19] is another model in this category that assesses the plausibility of triples by considering the similarity of their latent representations through matrix multiplications. The RotatE model [20] provides a rotation-based score function where the subject is rotated towards the object via the predicate. Both DistMult and ComplEx influence this model. QuatE [21] is designed in the quaternion space and, similar to RotatE represents a predicate as a rotation. However, a rotation in quaternion space is different and more expressive than a rotation in complex space as the dot product. The subjects, predicates, and objects are modeled in quaternion space in which three imaginary components exist in the latent representation.

Here, we use the score function of several state-of-the-art models mentioned in Table 1.

B. ASSISTING KGEs WITH TEXTUAL DESCRIPTION

There is a thread of works that utilizes text embeddings to enhance the performance of KGEs. For an entity classification task, DKRL [23] uses the description of entities employing Word2Vec [24] in the score function of the TransE [15] model. This approach is not bridging PLMs and KGEs, and cannot be generalized to other types of KGEs as it is designed explicitly for TransE. Similar to LiteralE [25], it also ignores the textual information of the relations inside a KG while using a computationally expensive approach to achieve entity descriptions. Additionally, in [26], a method is proposed to use the textual description of entities and relations in a language model. The output of the language model is modeled as a score function to learn the knowledge graph embedding, which is different from our work as we propose a loss function. There were also a few recent works to transfer structural knowledge into PLMs using various knowledge encoding techniques [26]–[29], which solved the problem from the side of language models, not KGEs as considered in our work. In [30], a method is proposed to train language models and KGEs jointly. It utilizes the structural knowledge from KG in order to achieve a better inference capability for PLMs. However, this approach suffers from high computational and memory costs as the LMs are trained through the learning process, which is not the case for our proposed solution. In addition, there is a risk of information loss due to the use of two objective functions separately for PLM and KGE, as well as the shared embedding space (using an encoder). Whereas, we propose one unified loss function and use pre-trained models that lower the computation costs to a large extent. Our approach can be independently plugged into any KGE model because of utilizing the PLM information in the proposed loss function.

C. PRE-TRAINED LANGUAGE MODELS

In recent years, Transformer architecture-based pre-trained language models [7] have revolutionized the domain of natural language processing. PLMs are widely used to solve various downstream tasks such as question answering [31], document retrieval [32], and language evaluation [33], [34]. PLMs are trained on huge corpora with the objective of understanding textual patterns. Architecturally, PLMs such as BERT [8], Sentence-BERT [12], fastText [11], GPT-2 [10], and T5 [35] contain a high number of parameters (e.g., BERT-base with 110 million parameters), which facilitate them to understand a large set of vocabularies and capture a wide range of patterns. However, language models alone do not consider other forms of knowledge representation, such as graph-based structures. Within the scope of this work, we focus on two types of PLMs, namely a Transformer-based model [7] that understands the sentence-level context namely Sentence-BERT and a Skip-gram-based statistical language model [36] namely fastText. fastText was previously trained on the Wikipedia corpus utilizing the Skip-gram model. Both Sentence-BERT and fastText [11] models can capture language patterns on a sub-word level and handle out of vocabulary words, which other models such as Glove [37] and Word2vec [24] remain short. In this work, we leveraged pre-trained language models to obtain the contextualized embedding of the entities to guide the learning process of the knowledge graph embedding models.

Algorithm 1 Training Procedure of the Algorithm

input : Knowledge Graph $\mathcal{K} = \{(s, p, o)|s, o \in \mathcal{E}, p \in \mathcal{R}\}$ ($\mathcal{E}$, $\mathcal{R}$ are the set of entities and relation), embedding dimension d, Embedding from PLM as $\Phi_{\mathbf{LM}} : \xi/\mathcal{R} \rightarrow \mathbf{E}^{\mathbf{LM}}/\mathbf{R}^{\mathbf{LM}}$ (not learnable), The learning rate α, The slack variable λ (learnable).

output: Optimized entity/relation embedding $\phi_{\mathbf{KG}} : \xi/\mathcal{R} \rightarrow \mathbf{E}/\mathbf{R}$.

Initialize all the trainable parameters i.e. embedding vectors of all entities/relations $\boldsymbol{E}/\boldsymbol{R}$;

while *not converged* **do**

  take a random batch $\mathcal{K}_b$ from $\mathcal{K}$;

  generate random negative samples $\mathcal{K}'_b = \{(s', p, o')\}$

  compute the score of triples in $\mathcal{T}'_b$ and $\mathcal{K}_b$ (equation 3);

  $P(s, p, o) \leftarrow \sigma(-d(s, p, o) + \gamma - \lambda\mathcal{C}_{s,p,o})$ from $\mathcal{K}_b = \{(s, p, o)\}$;

  $P(s', p, o') \leftarrow \sigma(d(s', p, o') - \gamma + \lambda\mathcal{C}_{s',p,o'})$ from $\mathcal{K}'_b = \{(s', p, o')\}$;

  Compute the loss value using $log\mathcal{L} \leftarrow \sum_{(s,p,o)\in\mathcal{K}} \left(P(s, p, o) + \sum_{(s',p,o')\in\mathcal{K}'_{(s,p,o)}} P(s', p, o')\right)$ (equation 5);

  Update $\phi_{\mathbf{KG}}$ w.r.t $\nabla \log \mathcal{L}$

end

IV. PROPOSED APPROACH

The proposed approach is a language model-guided loss function that allows the KGE models to employ the pre-trained embedding into the learning process. The loss function gets two different types of embeddings as input, along with the slack variable λ and margin γ. At the beginning of the learning process, the embeddings of all entities $\mathbf{E}$ and relations $\mathbf{R}$ are randomly initialized. These embedding vectors are learned during the whole training process by using the KGE score function and optimization over a loss function. On the other hand, the second set of embeddings comes from PLMs. These are denoted as $\mathbf{E}^{LM}$ for the set of entities $\mathcal{E}$ and $\mathbf{R}^{LM}$ for the set of relations $\mathcal{R}$. A confidence value $\mathcal{C}(s, p, o)$ is computed that represents the plausibility for each triple (s, p, o) only from the textual point of view. Here, $\mathcal{C}_{s,p,o} = \mathcal{C}(s, p, o) = \langle \mathbf{s}^{LM}, \mathbf{p}^{LM}, \mathbf{o}^{LM} \rangle$ as the product of three vectors (subject, predicate and object of a triple). We use a product to compute the confidence from text because if a triple is correct, in the text of subject entity, object and relation might appear (same for the object). Therefore, their embeddings from the language model are close, and the product between the vectors gives a high confidence value. Such confidence value can guide the learning process of KGE models. However, this is not the case for all triples, as the reliability of the confidence value depends on the quality of the gathered text of subject, predicate, and object. Therefore, for all the triples of the underlying KG, a common trainable slack variable λ is defined to regulate the effect of textual information in the final score values. The slack variable conveys the confidence based on the contextual information of the triple to guide the score obtained from KGEs. Finally, the likelihood estimation is computed using a Sigmoid function. Later the likelihood estimation gives the final maximum log-likelihood loss. All the steps of our approach are described in more details in Algorithm 4.

The construction of the loss function is presented in a step-wise manner by providing an example of the distance-based class of embedding models. For each positive triple, the distance should be a small value, upper-bounded by a term containing the confidence obtained from the language model. However, for each negative triple, the distance should be a big value, lower-bounded by a term containing the confidence obtained from a language model. The effect of the boundary after including PLMs confidence is depicted in equation 1.

$$\begin{cases} d(s, p, o) \ll \gamma - \lambda\mathcal{C}_{s,p,o}, & \text{if } (s, p, o) \in \mathcal{K} \\ d(s', p, o') \gg \gamma - \lambda\mathcal{C}_{s',p,o'}, & \text{if } (s', p, o') \in \mathcal{K}', \end{cases} \tag{1}$$

where λ is a learnable parameter, and γ is a hyper-parameter. Note that the parameter λ is used to adjust the scale of the confidence value coming from a language model. Therefore, it balances the negative affect that might be caused by irrelevant text present for entities or relations. Rearranging the components of equation 1, leads to equation 2:

$$\begin{cases} 0 \ll -d(s, p, o) + \gamma - \lambda\mathcal{C}_{s,p,o}, \\ 0 \ll d(s', p, o') - \gamma + \lambda\mathcal{C}_{s',p,o'}. \end{cases} \tag{2}$$

Let us consider $-d(s, p, o) + \gamma - \lambda\mathcal{C}_{s,p,o}$ and $d(s', p, o') - \gamma + \lambda\mathcal{C}_{s',p,o'}$ as x and x'. In order to utilize the likelihood estimation for x a Sigmoid function $\sigma(x) = \frac{e^x}{1+e^x}$ is deployed to scale the values of the equation 2 between 0 and 1. For x' Sigmoid function has been applied in a similar way. After performing the Sigmoid operation in equation 2, the utilization of the likelihood estimation is constructed. The formal definition of this process is shown in equation 3.

$$\begin{cases} P(s, p, o) = \sigma(-d(s, p, o) + \gamma - \lambda\mathcal{C}_{s,p,o}) \approx 1, \\ P(s', p, o') = \sigma(d(s', p, o') - \gamma + \lambda\mathcal{C}_{s',p,o'}) \approx 1. \end{cases} \tag{3}$$

Note that the equation 2 and 3 are equivalent due to using the Sigmoid function where $x \rightarrow \infty$ then $\sigma(x) \rightarrow 1$. In the following equation, we use a maximum likelihood estimation ($\max(\mathcal{L})$) to satisfy the equation 3:

$$\mathcal{L} = \prod_{(s,p,o)\in\mathcal{T}} \left(P(s, p, o) \prod_{(s',p,o')\in\mathcal{T}'_{(s,p,o)}} P(s', p, o')\right), \tag{4}$$

where $\mathcal{T}, \mathcal{T}'_{(s,p,o)}$ are the set of all triples in a KG, and the set of all negative samples obtained by the corruption of a positive sample (s, p, o), respectively. Each negative sample is obtained based on uniform sampling [15], [22].

In order to relax the problem, we use a log-likelihood $log\ \mathcal{L}$. We formulated it in equation 5:

$$log\ \mathcal{L} = \sum_{(s,p,o)\in\mathcal{T}} \big(P(s,p,o) + \sum_{(s',p,o')\in\mathcal{T}'_{(s,p,o)}} P(s',p,o')\big). \tag{5}$$

For non-distance based models, we replace $\gamma - d(s,p,o)$ by their score functions $f(s,p,o)$ in the above formulae. In order to perform link prediction, we can use either $f(s,p,o) = \gamma - d(s,p,o)$ (original score function of KGEs) or $P(s,p,o) = \sigma(\gamma - d(s,p,o) - \lambda\mathcal{C}_{s,p,o})$. It has been analysed for each of them together with $\mathcal{C}_{s,p,o}$ as ablation study in section V-D. It is noteworthy that our proposed loss function (equation 5) is a generalization of the loss function proposed in [20] with additional capability of injecting additional textual knowledge.

V. EXPERIMENTS

In this section, we provide the results of an extensive set of experiments that were done to evaluate the effect of the proposed loss function. The following describes the considered KGs and the experimental setup.

A. BENCHMARK DATASETS

We evaluate our approach on several publicly available datasets, namely: Nations [38], UMLS [39], WN9 (constructed in this work), and CoDEx [40]. We conduct experiments on three different splits of the CoDEx dataset (small, medium, and large): CoDEx-S, CoDEx-M, and CoDEx-L. Here, we provide a detailed description of the datasets, and comprehensive statistics are reported in Table 2.

- **Nations** dataset includes a set of relationships between nations and their features. The dataset consists of binary and unary relations.
- **UMLS** dataset (standing for Unified Medical Language System) is a high-level ontology for organizing a large number of terminologies used in the biomedical domain into a unified vocabulary that allows for uniform access to disparate medical resources.
- **WN9** is a subset of WN18 [41] dataset with 9 relations. The textual information is constructed in this work from the associated resources that come with WordNet glosstag files.[1] The XML resource contains <*synset id*> and <terms> tags which refer to the entity ID and name, respectively. In the case multiple <term> information under <terms> tag, we consider first <term> as the entity name.
- **CoDEx** [40] provides three comprehensive knowledge graph datasets that include positive and hard negative triples, entity types, entity, and relation descriptions. The knowledge graph in CoDEx is constructed from Wikidata [42] and Wikipedia[2] datasets.

[1]https://wordnetcode.princeton.edu/glosstag.shtml
[2]https://www.wikipedia.org/

B. EXPERIMENTAL SETUP

The experimental setup includes the introduction of baseline models, evaluation metrics as well as the hyperparameters search setting.

1) BASELINE MODELS

The evaluations of the proposed loss function have been conducted on the following known knowledge graph embedding models: TransE [15], RotatE [22], ComplEx [19], DistMult [18], and QuatE [21]. The respective score functions of these KGEs are reported in Table 1. The selection of these models has been made through a systematic analysis considering these criteria: a) variety in the model type (translation-based, semantic matching) based on the design of their score function; b) diversity of geometric space (Euclidean, Complex, and Quaternion) c) outperforming in their group or beyond.

2) EVALUATION METHODOLOGY & METRICS

We measure the performance of the models in the link prediction task using the following standard metrics as used in [4], [22]. Basically, the evaluation of the Knowledge embedding models aims to solve the link prediction task. Given a set of triples $\mathcal{K}_{test} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ in the evaluation which are not seen during the training process, for each triple $(h,r,t) \in \mathcal{K}_{test}$, we predict either the head or tail. If the model aims to predict the head, then it is called a head prediction where the query is $(?,r,t)$. On the other hand, the tail query is considered as $(h,r,?)$. For both of the cases, ? is replaced with all the possible entities in the Knowledge Graph. The rank of a true triple is considered the position among all the possible choices (all the combinations by replacing the ? with all possible entities) if all the combinations (including the correct triple) are sorted by their score or plausibility. The Mean Rank shown by MR is considered as the average rank of all the test triples. Mean Reciprocal Rank (MRR) indicates the average of the inverse rank for the correct triples. The hits ratio represents the proportion of the correct entities appearing in the top n positions where $n \in \{1,3,10\}$ and these are considered as H@1,3,10. A higher value of MRR and Hits@n represents a better evaluation performance.

3) HYPER-PARAMETER SETTINGS

To make a fair comparison, we obtained results on each dataset with the same hyper-parameters for all the settings. We train the models on CoDEx until 20,000 steps ($\mathcal{S}$), WN9 until 30,000 steps ($\mathcal{S}$), and Nations and UMLS until 3,000 steps ($\mathcal{S}$). Note that each step is one pass of the optimization per batch. The training objective is optimized using Adam [43] with a learning rate α of 0.01. Through several experiments, gamma (γ) of 15 is set for training TransE, RotatE, and QuatE and 50 for ComplEx and DistMult. During the training on Nations and UMLS, a batch size $\mathcal{B}$ of 256 and dimension 100 are used, and for WN9 and CoDEx, the batch size $\mathcal{B}$ is set to 512, and

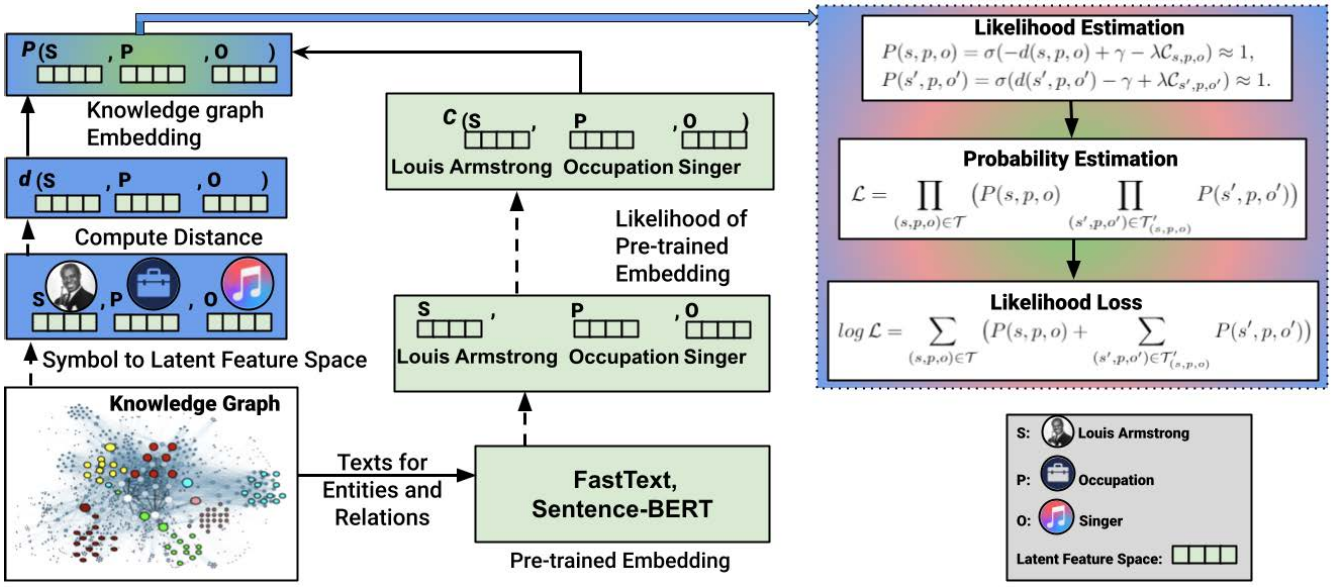


FIGURE 2. System architecture of proposed model. The positive triples are illustrated for KG and LM models to represent the system architecture with simplicity. Later, both positive and negative triples are obtained from the KG and LM models to calculate loss functions.

TABLE 2. Dataset statistics. The table shows the number of entities and relations as well as the number of overall triples, also in the split of train, validation, and test datasets, where *Chars* refers to characters.

Dataset	#entity	#relation	#train	#validation	#test	#triples	Vocabulary	Avg. #Chars (E)	Avg. #Chars (R)
Nations	14	55	1,592	199	201	1,992	1,992	7.786	14.455
UMLS	135	46	5,216	652	661	6,529	2,614	20.830	13.336
WN9	6,555	9	11,741	1,337	1,319	14,397	15,967	80.873	17.33
CoDEx-S	45,869	68	32,888	1,827	1,828	36,543	5,492	50.165	104.357
CoDEx-M	11,941	50	185,584	10,310	10,311	206,205	23,492	45.435	102.725
CoDEx-L	45,869	69	551,193	30,622	30,622	612,437	91,918	39.294	96.754

dimension d of 500 is used. 50 negative sampling ($\mathcal{N}$) and 1.0 adversarial temperature ($\mathcal{T}$) have been chosen for running experiments on Nations, UMLS, and CoDEx, while for WN9, we use 1024 negative samples ($\mathcal{N}$) and 0.5 adversarial temperature ($\mathcal{T}$). All the experiments have been conducted using a Tesla V100 machine with 16GB memory.

C. EVALUATIONS AND RESULTS

Table 3 and Table 4 report the evaluation results on the benchmark datasets. The approaches with ST and FT are our methods on incorporating Sentence-Transformer(ST) (also known as Sentence-BERT) and fastText(FT) embedding, respectively. In both Table 3 and Table 4, the deltas (Δ_1 and Δ_2), denote the performance difference between standard knowledge graph embedding models and our proposed ST and FT-based methods, respectively. We show the results for a standard KGE model, Sentence-BERT guided KGE model (KGE model+*ST*), and fastText guided KGE model (KGE model+*FT*). Our outperforming results in these tables are highlighted in blue and bold; for other models, we underline them. The δ_1 and δ_2 represent the differences between the baseline models and our models, ST and FT, respectively. The improvement is highlighted in green otherwise we highlight them in pink. The results in Table 3 show that the inclusion of text embedding in the learning process enables the knowledge graph embedding models to achieve better results. Some of the significant improvements can be seen in Nations and UMLS for RotatE. In nations, RotatE model has an improvement of Hit@3 from 0.435 to 0.455 in TransE+FT. In the UMLS dataset, the RotatE+FT model provides an improvement in the Hit@1 from 0.874 to 0.886, and Hit@3 also increased from 0.952 to 0.961. By using the proposed approach, we found the computation time is almost similar. For this reason, we have collected the runtime of RotatE model for 5 different runs for UMLS dataset. Both with or without ST are considered to be checked for the runtime comparison. In both cases with 3000 stepsize ($\mathcal{S}$), dimension 100 (d) the difference between the average runtime is 0.487 seconds, which means our approach does not increase the runtime drastically.

An overall improvement in Hits@1 and Hits@3 scores is evident across most datasets, where comparable results are noticeable in other metrics such as Hits@10 and MRR. Obtaining improved results in Hits@1 is generally challenging. However, the results in Table 3 exhibit a considerable improvement in Hits@1 on the test set across several

TABLE 3. Comparisons on nations, UMLS and WN9 datasets.

Model	Δ	Nations				UMLS				WN9			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE		70.1	56.7	78.6	99.0	92.8	89.5	95.0	98.8	89.7	88.3	90.7	91.8
TransE+ST		70.6	56.7	**80.8**	98.5	92.1	88.3	95.2	98.5	**89.7**	**88.6**	90.6	91.7
TransE+FT		**71.2**	**57.5**	79.9	98.5	92.6	89.0	**95.5**	98.6	**89.7**	88.5	**90.7**	**91.9**
	Δ_1	+0.5	0.0	+2.2	-0.5	-0.7	-1.2	+0.2	-0.3	0.0	+0.3	-0.1	-0.1
	Δ_2	+1.1	+0.8	+1.3	-0.5	-0.2	-0.5	+0.5	-0.2	0.0	+0.2	0.0	+0.1
RotatE		62.4	43.5	76.9	99.0	91.6	87.4	95.2	98.9	89.9	88.5	90.9	92.2
RotatE+ST		63.6	45.0	76.1	**99.2**	91.8	87.3	95.7	98.9	89.7	88.1	90.9	**92.3**
RotatE+FT		**62.8**	**45.5**	76.4	98.8	**92.7**	**88.6**	**96.1**	**99.0**	0.897	88.2	**91.0**	92.2
	Δ_1	+0.2	+1.5	-0.8	+0.2	+0.2	-0.1	+0.5	0.0	-0.2	-0.4	0.0	+0.1
	Δ_2	+0.4	+2.0	-0.5	-0.2	+1.1	+1.2	+0.9	+0.1	-0.2	-0.3	+0.1	0.0
ComplEx		58.7	40.0	69.9	98.2	72.8	62.0	80.9	91.6	90.8	90.0	91.4	92.0
ComplEx+ST		55.2	34.1	68.1	98.3	71.5	60.1	80.0	91.7	90.7	90.0	91.2	**92.0**
ComplEx+FT		**58.7**	38.6	**74.1**	**99.5**	72.1	60.5	**80.9**	**92.4**	**90.9**	**90.2**	**91.5**	**92.0**
	Δ_1	-3.5	-5.9	-1.8	+0.1	-1.3	-1.9	-0.9	+0.1	-0.1	0.0	-0.2	0.0
	Δ_2	0.0	-1.4	+4.2	+1.3	-0.7	-1.5	0.0	+0.8	+0.1	+0.2	+0.1	0.0
DistMult		66.9	52.7	74.9	99.3	61.4	51.4	65.9	82.5	70.5	53.2	87.2	91.1
DistMult+ST		67.0	52.2	**75.3**	98.8	67.8	60.2	72.8	**85.9**	70.4	**53.1**	87.1	**91.1**
DistMult+FT		**67.1**	**52.7**	74.6	99.0	**68.9**	**60.5**	**73.2**	85.1	70.4	**53.1**	**87.3**	91.0
	Δ_1	+0.1	-0.5	+0.4	-0.5	+6.4	+8.8	+6.9	+3.4	-0.1	-0.1	-0.1	0.0
	Δ_2	+0.2	0.0	-0.3	-0.3	+7.4	+9.1	+7.3	+2.6	-0.1	-0.1	+0.1	-0.1
QuatE		52.6	32.3	66.4	96.8	68.9	57.0	76.5	92.2	82.3	78.9	84.9	87.7
QuatE+ST		48.4	27.1	59.2	96.5	**70.1**	**58.0**	**79.0**	92.1	82.3	79.0	**84.9**	**87.9**
QuatE+FT		49.7	28.4	61.2	**97.8**	68.4	56.4	76.2	90.9	**82.5**	**79.1**	84.8	87.6
	Δ_1	-4.2	-5.2	-7.2	-0.3	+1.2	+1.0	+2.5	-0.1	0.0	+0.1	0.0	+0.2
	Δ_2	-2.9	-3.9	-5.2	+1.0	-0.5	-0.6	-0.3	-1.3	+0.2	+0.2	-0.1	-0.1

benchmarks. Table 4 shows the results of the baseline model on three splits of the CoDEx dataset. For different splits of CoDEx, a remarkable improvement in the performance of all the metrics is noticeable across the baseline models. Overall, we observe improved results across several datasets. Only in RotatE and DistMult on CoDEx-M dataset, our proposed approach achieves slightly low yet comparable results. One of the possible reasons for not achieving the desired result in some cases can be the disagreement between the connection of the graph and the PLM vectors which, is studied below.

1) TRAINED EMBEDDING VS PLM EMBEDDING

In Figure 3, through a systematic analysis, we sampled entities from CodEx-S, for which we computed cosine similarity. It can be seen that there are often dis-similarities between the trained embeddings for capturing structural information and PLM. In many cases, the similarity between the entities is high if they are trained from the KGE model. In the case of embedding coming from PLMs, the lower score is due to disagreement with the structural information. For example, ''Canadian musician'' and ''American rapper'' have high similarity in the trained embedding from KGE, but in the case of PLM vectors, they are not very similar. Our observation confirms that this can be caused by the textual difference between ''Canadian'' and ''American''. This situation can lead to performance degradation due to the anomaly in the conveyed information. For this reason, only using PLM vectors for the link prediction task does not perform as expected, which can be clearly seen from Table 5. As mentioned, having a very small value for λ mitigates the negative effect of disagreements between structural and textual information. However, this does not solve the problem in all scenarios due to the fact that λ is shared between all triples during the training phase. Overall, this becomes problematic when triples in the test set contain less informative textual information compared to the overall textual information in the training triples.

D. ANALYSIS OF THE ABLATION STUDIES

We further analyze several things: Firstly, we analyze the effect of using only PLMs. The effect of the embedding dimension and the influence of the training step were analyzed in order to understand further dynamics such as overfitting.

1) Effect OF USING ONLY PLMs AS SCORING FUNCTION

As a first step of the ablation study, we only explore the effect of pre-trained language models. In this case, we evaluated the score ($\mathcal{C}_{s^{LM},p^{LM},o^{LM}}$) which is obtained from the language models only. The evaluations are performed on UMLS, Nations, CoDEx-S and are shown in Table 5. We observe a significant decrease in the results when considering only PLMs (highlighted in red). More specifically, in the case of the UMLS dataset, Hits@10 dropped from 0.996 (RotatE+ST) to 0.160(Only ST); and

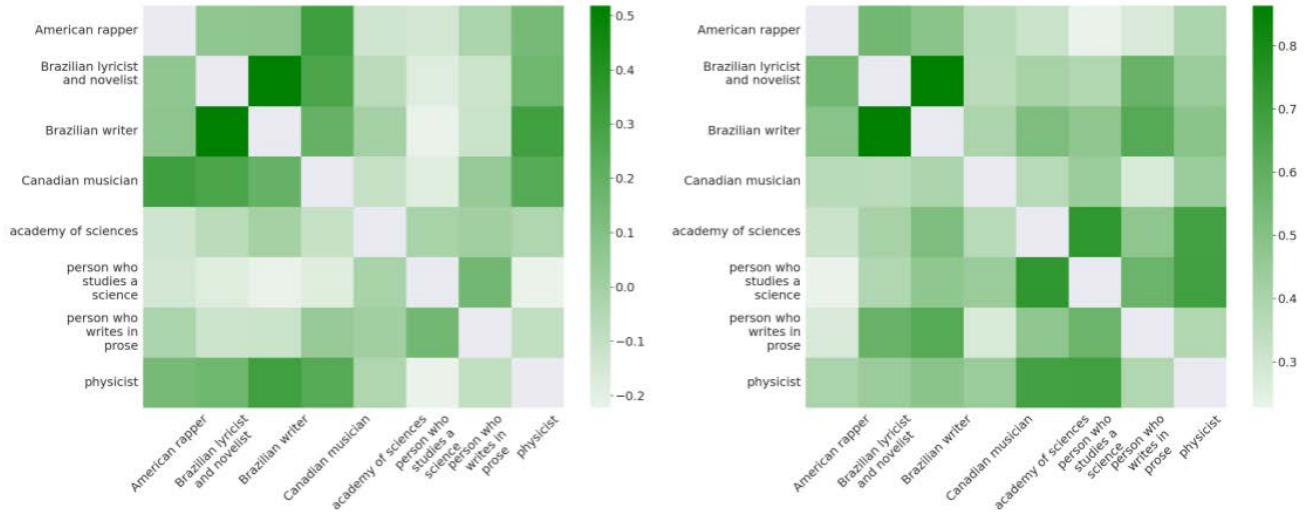


FIGURE 3. Heatmap visualization between trained embedding vs PLM embedding (ST).

TABLE 4. Comparisons on CoDEx-S, CoDEx-M and CoDEx-L datasets.

Model	Δ	CoDEx-S				CoDEx-M				CoDEx-L			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE		48.8	40.0	52.1	66.7	43.5	36.8	46.4	56.2	40.7	34.8	43.3	52.0
TransE+ST		48.1	39.3	51.1	66.6	**43.7**	37.1	46.5	56.3	40.8	34.9	**43.4**	52.1
TransE+FT		**48.8**	**40.0**	51.9	**66.8**	**43.7**	**37.2**	**46.6**	**56.4**	**40.9**	**35.0**	43.3	**52.2**
	Δ_1	-0.7	-0.7	-1.0	-0.1	+0.2	+0.3	+0.1	+0.1	+0.1	+0.1	+0.1	+0.1
	Δ_2	0.0	0.0	-0.2	+0.1	+0.2	+0.4	+0.2	+0.2	+0.2	+0.2	+0.0	+0.2
RotatE		54.4	46.5	58.3	69.5	47.8	41.8	50.4	59.3	46.8	41.4	49.1	57.2
RotatE+ST		54.3	**46.7**	57.4	69.5	**47.8**	41.7	50.2	59.1	46.7	41.1	**49.3**	**57.2**
RotatE+FT		**54.4**	46.4	57.4	**69.6**	**47.8**	41.7	50.3	**59.4**	**46.8**	41.2	49.2	**57.2**
	Δ_1	-0.1	+0.2	-0.9	0.0	0.0	-0.1	-0.2	-0.2	-0.1	-0.3	+0.2	0.0
	Δ_2	0.0	-0.1	-0.9	+0.1	0.0	-0.1	-0.1	+0.1	0.0	-0.2	+0.1	0.0
ComplEx		37.1	29.8	38.2	51.9	37.3	33.1	37.9	45.3	42.2	37.5	44.0	51.2
ComplEx+ST		37.6	**30.9**	38.2	**52.3**	**37.5**	**33.5**	**38.1**	**45.4**	**43.2**	**38.5**	**44.9**	**52.0**
ComplEx+FT		**37.9**	30.7	**38.6**	**52.3**	37.2	33.0	37.8	**45.4**	42.2	37.6	43.9	51.0
	Δ_1	+0.5	+1.1	0.0	+0.4	+0.2	+0.4	+0.2	+0.1	+1.0	+1.0	+0.9	+0.8
	Δ_2	+0.8	+0.9	+0.4	+0.4	-0.1	-0.1	-0.1	+0.1	0.0	+0.1	-0.1	-0.2
DistMult		39.2	32.1	40.3	54.3	37.9	32.8	39.5	47.9	40.2	35.0	42.1	50.0
DistMult+ST		**39.9**	**33.0**	**41.4**	54.2	37.7	32.6	39.2	47.8	**41.3**	**36.1**	**43.5**	**51.5**
DistMult+FT		39.3	32.3	40.6	53.7	**38.0**	**32.8**	39.4	**48.2**	38.2	33.2	39.8	47.9
	Δ_1	+0.7	+0.9	+1.1	-0.1	-0.2	-0.2	-0.3	-0.1	+1.1	+1.1	+1.4	+1.5
	Δ_2	+0.1	+0.2	+0.3	-0.6	+0.1	0.0	-0.1	+0.3	-2.0	-1.8	-2.3	-2.1
QuatE		38.1	30.2	39.3	56.5	33.4	30.9	32.9	37.7	39.9	37.4	40.5	44.4
QuatE+ST		37.8	**30.4**	38.1	55.8	33.3	30.8	32.9	37.2	40.0	37.4	40.7	44.5
QuatE+FT		37.3	29.7	37.8	55.8	**33.5**	**30.9**	**33.1**	**38.1**	**40.2**	**37.6**	**40.8**	**44.9**
	Δ_1	-0.3	+0.2	-1.2	-0.7	-0.1	-0.1	0.0	-0.5	+0.1	0.0	+0.2	+0.1
	Δ_2	-0.8	-0.5	-1.5	-0.7	+0.1	0.0	+0.2	+0.4	+0.3	+0.2	+0.3	+0.5

for the CoDEx-S dataset, Hits@10 dropped from 0.696 (RotatE+ST) to 0.012 (Only PLM ST). Here ST and FT correspond to sentence transformer and fastText, respectively. Since PLMs lack the structural information of the KG, it is not sufficient to use only PLMs for link prediction. However, the evaluations on the Nation dataset show satisfactory results, which are highly affected by the structure of the KG. This shows that contextual meaning corresponding to the entities and relations has a high correlation with the structural connectivity of KG, and the number of entities and relations. Structural analysis of the Nation dataset also approved that the connection of the graph is generally well reflected in the contextual meaning of the entities and relations. In order to perform this evaluation, we fixed the dimension d to 100, the learning rate α to 0.01, and the number of negative samples $\mathcal{N}$ to 10. The batch size of these evaluations was set to $\mathcal{B}$ to 256, adversarial temperature $\mathcal{T}$ to 0.5, margin γ to 10, and step size $\mathcal{S}$ to 5000. One possibility is that since this dataset

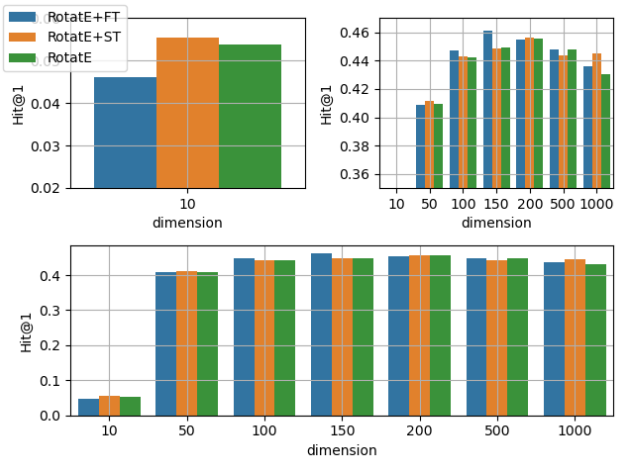


FIGURE 4. Comparison of Hits@1 on different dimensions in the CoDEx-S dataset.

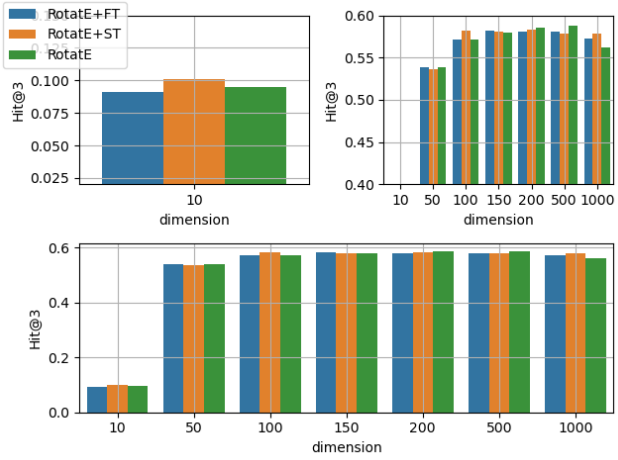


FIGURE 5. Comparison of Hits@3 on different dimensions in the CoDEx-S dataset.

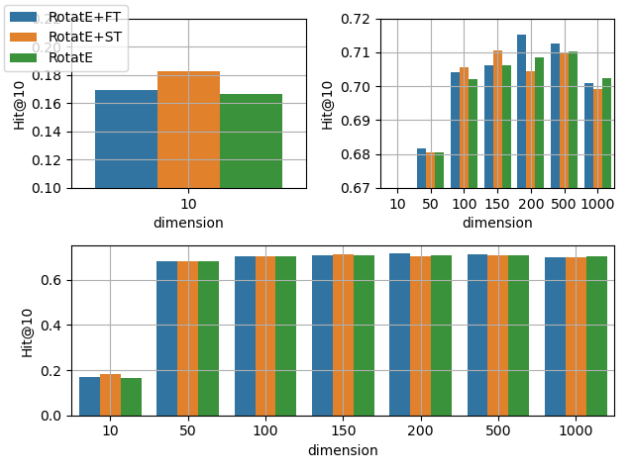


FIGURE 6. Comparison of Hits@10 on different dimensions in the CoDEx-S dataset.

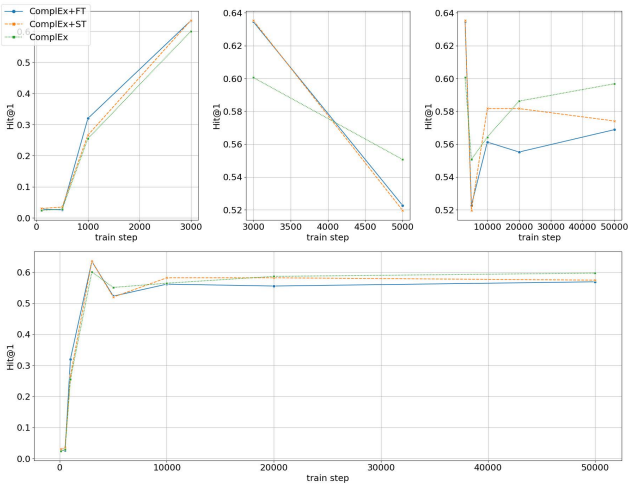


FIGURE 7. Comparison of Hits@1 on different step sizes in the UMLS dataset.

only has 13 entities, the evaluation with a focus on Hits@10 might result in high values for this metric.

2) EFFECT OF EMBEDDING DIMENSION

As an ablation study, we evaluated the effect of dimension on the performance of the models with/without using PLMs. Here, we demonstrate the results of the RotatE model in the respective Figures 4, 5, and 6.

The comparison of the performance on Hits@1 for both PLMs + RotatE, in contrast to baseline RotatE, illustrates most for a large spectrum of selected dimensions {10, 50, 100, 150, 200, 500, 1000} in Figure 4 is improved after the inclusion of PLMs. It is visible that, in most of the cases, either FT or ST with RotatE achieves better performance. The same effect can be seen in Figure 5. The improvement in Hits@1 is higher compared to Hits@3 considering the dimensions (can be seen between Figures 4 and 5). Except for a few cases, almost in every dimension, the inclusion of PLM illustrates improvement.

Figure 6 also demonstrates a significant improvement in the selected dimensions considering Hits@10. Generally, it is observed that from dimensions 10 to 200, the scale of improvement is larger than the rest of the dimensions. By increasing the dimension, the models can learn more structural information with higher complexity. Therefore, the structural information becomes more influential in overall performance. In order to perform these analyses, the same hyper-parameters are used as in 5 other than the dimension d, which is varied in order to show its effect.

3) INFLUENCE OF TRAINING STEP

Since the λ parameter is changing in each step size, we conducted an ablation on whether increasing the number of training steps improves performance. In order to do so, the step size is divided into three regions (low, medium, and high). A low step region includes the step $\mathcal{S}_{low} = \{100, 500, 1000\}$.

Furthermore, the training step $\mathcal{S}_{mid} = \{3000, 5000\}$ remains medium and $\mathcal{S}_{high} = \{10000, 20000, 50000\}$ in the

TABLE 5. Comparison of results between embeddings from the PLMs in contrast to the one from RotatE & RotatE+ST.

Embedding Source	UMLS				Nations				CoDEx-S			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RotatE	0.963	0.944	0.978	0.995	0.675	0.517	0.794	0.990	0.513	0.422	0.556	0.690
RotatE+ST	0.965	0.946	0.977	**0.996**	**0.682**	**0.527**	**0.806**	**0.993**	0.514	0.422	0.560	0.692
RotatE+FT	**0.970**	**0.953**	**0.981**	0.995	0.652	0.488	0.749	0.990	**0.521**	**0.432**	**0.564**	**0.694**
PLM (ST)	0.081	0.024	0.076	0.160	0.418	0.199	0.532	0.986	0.04	0.00	0.02	0.04
PLM (FT)	0.063	0.028	0.045	0.095	0.450	0.239	0.545	0.968	0.08	0.00	0.03	0.012

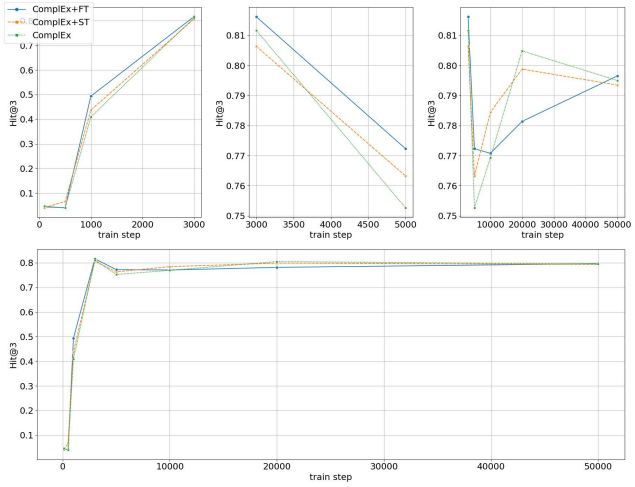


FIGURE 8. Comparison of Hits@3 with different step size with UMLS dataset.

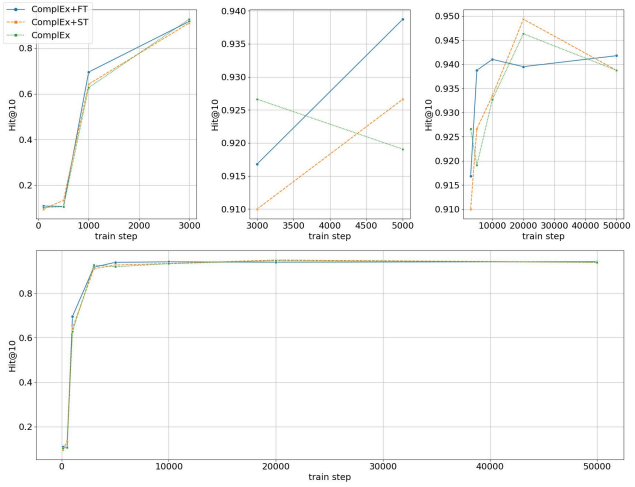


FIGURE 9. Comparison of Hits@10 with different step size with UMLS dataset.

high region. Figures 7, 8, and 9 demonstrate the effect of training steps. These evaluations considering Hits@1 can be seen in Figure 7. It is observed that in $\mathcal{S}_{low}$ and $\mathcal{S}_{mid}$ regions (i.e., until the step size 3000) the improvement for PLMs+KGE is noticeable. The baseline KGE starts to improve compared to our approach in the $\mathcal{S}_{high}$ region (step size 500 onward).

Figure 8 also shows noticeable improvement, specifically in $\mathcal{S}_{low}$ and $\mathcal{S}_{mid}$ regions–until training step 10000. Afterward, the results for the used baseline KGE model (ComplEx) started to improve. The same behavior as Hits@1 and Hits@3 is observable for Hits@10 which is shown in Figure 9. In a nutshell, the improvement can be insignificant or worse depending on whether the training step is considered too large.

The hyper-parameters we have chosen in this regards include negative sample size $\mathcal{N} = 50$, embedding dimension $d = 100$, batch size $\mathcal{B} = 256$, adversarial temperature: $\mathcal{T} = 1.0$, learning rate $\alpha = 0.01$ and the KGE model as ComplEx.

4) LIMITATIONS

The proposed approach highly depends on the quality of textual information. There are knowledge graphs in which such complementary knowledge is completely missing. In order to make use of language models, the textual information needs to be collected through a complex procedure. Additionally, the knowledge graph with textual information needs to go through a complex data quality check. Our model falls short when the dissimilarity between the structural and textual information is huge.

VI. CONCLUSION AND FUTURE WORK

This work proposes a novel approach for leveraging pre-trained language models in order to guide the knowledge graph embedding models. The main contribution of this work is the design and development of a model free loss function that utilizes the additional textual information through PLMs, which can be plugged into any KGE model. The empirical evaluations demonstrate that by using additional embeddings corresponding to the textual knowledge of entities and relations of a knowledge graph through pre-trained language models in a maximum log-likelihood loss, the performance of baseline embedding models improves substantially. The compassion of resultant embeddings from PLMs alone does not lead to high performance. In addition, we further addressed observations regarding the effect of datasets and their structural information as well as the evaluation setting in terms of training steps, and dimensions.

VII. FUTURE WORK

We plan to investigate the proposed loss function by using embedding generated by contemporary large-scale language models such as GATO [44] and OPT [45]. While this

work was using sentence level textual information, the next step is to leverage pre-trained embedding on the word, and document levels. We strongly believe that our findings will motivate researchers to investigate the use of PLMs in enhancing representation learning further. This is expected to influence the downstream AI tasks that are using KG embedding models in practice for recommendation, prediction, or question answering systems.

ACKNOWLEDGMENT

Prof. Lehmann contributed to this work while at University of Bonn, 53113 Bonn, Germany. The evaluations have been done using High Performance Computing Center, Dresden, Saxony. *(Mirza Mohtashim Alam and Md Rashad Al Hasan Rony contributed equally to this work.)*

REFERENCES

[1] Q. Wang, Z. Mao, and B. Wang, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[2] L. Bellomarini, E. Sallinger, and S. Vahdati, "Knowledge graphs: The layered perspective," in *Knowledge Graphs and Big Data Processing*. Cham, Switzerland: Springer, 2020, pp. 20–34.

[3] L. Bellomarini, E. Sallinger, and S. Vahdati, "Reasoning in knowledge graphs: An embeddings spotlight," in *Knowledge Graphs and Big Data Processing*. Cham, Switzerland: Springer, 2020, pp. 87–101.

[4] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, and J. Lehmann, "Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 4, 2021, doi: 10.1109/TPAMI.2021.3124805.

[5] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 2, pp. 1–49, Apr. 2021.

[6] V. Henk, S. Vahdati, M. Nayyeri, M. Ali, H. S. Yazdi, and J. Lehmann, "Metaresearch recommendations using knowledge graph embeddings," in *Proc. RecNLP Workshop AAAI Conf.*, 2019, pp. 1–6.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[8] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI Blog, Tech. Rep., 2019, vol. 1, no. 8, p. 9.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2016.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1–11.

[13] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, May 2020.

[14] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," 2022, *arXiv:2201.05337*.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[16] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, 2014, pp. 1–8.

[17] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1–7.

[18] B. Yang, W.-T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," 2014, *arXiv:1412.6575*.

[19] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2016, pp. 2071–2080.

[20] M. Nickel, V. Tresp, and H.-P. Kriegel, "Factorizing YAGO: Scalable machine learning for linked data," in *Proc. 21st Int. Conf. World Wide Web*, Apr. 2012, pp. 271–280.

[21] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," 2019, *arXiv:1904.10281*.

[22] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.

[23] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 1–7.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[25] A. Kristiadi, M. A. Khan, D. Lukovnikov, J. Lehmann, and A. Fischer, "Incorporating literals into knowledge graph embeddings," in *Proc. Int. Semantic Web Conf.* Auckland, New Zealand: Springer, 2019, pp. 347–363.

[26] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," 2019, *arXiv:1909.03193*.

[27] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: Enabling language representation with knowledge graph," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 2901–2908.

[28] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: ACL, Jul. 2019, pp. 1441–1451.

[29] K. Faldu, A. Sheth, P. Kikani, and H. Akbari, "KI-BERT: Infusing knowledge context for better language and domain understanding," 2021, *arXiv:2104.08145*.

[30] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and J. Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 176–194, Feb. 2021.

[31] M. R. A. H. Rony, D. Chaudhuri, R. Usbeck, and J. Lehmann, "Tree-KGQA: An unsupervised approach for question answering over knowledge graphs," *IEEE Access*, vol. 10, pp. 50467–50478, 2022.

[32] S. Borgeaud *et al.*, "Improving language models by retrieving from trillions of tokens," 2021, *arXiv:2112.04426*.

[33] M. R. A. H. Rony, L. Kovriguina, D. Chaudhuri, R. Usbeck, and J. Lehmann, "Rome: A robust metric for evaluating natural language generation," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5645–5657.

[34] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "MoverScore: Text generation evaluating with contextualized embeddings and Earth mover distance," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 563–578.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, *arXiv:1910.10683*.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Represent. (ICLR)*, Scottsdale, ARI, USA, May 2013, pp. 1–12.

[37] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[38] S. Kok and P. Domingos, "Statistical predicate invention," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 433–440.

[39] A. T. McCray, "An upper-level ontology for the biomedical domain," *Comparative Funct. Genomics*, vol. 4, no. 1, pp. 80–84, 2003.

[40] T. Safavi and D. Koutra, "CoDEx: A comprehensive knowledge graph completion benchmark," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8328–8350.

[41] X. Glorot, A. Bordes, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, 2014.

[42] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proc. 21st Int. Conf. companion World Wide Web*, 2012, pp. 1063–1064.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[44] S. Reed, K. Zolna, E. Parisotto, S. Gomez Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. Tobias Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A generalist agent," 2022, *arXiv:2205.06175*.
[45] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.

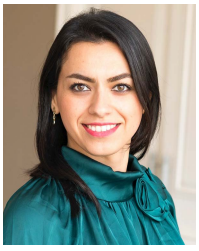
MIRZA MOHTASHIM ALAM received the bachelor's degree from the Department of Computer Science, BRAC University, Bangladesh, and the master's degree from the University of Bonn, Germany. He is currently pursuing the Ph.D. degree with the Smart Data Analytics (SDA) Research Group. He is a Senior Research Scientist with the Nature-Inspired Machine Intelligence Research Group, Institute of Applied Informatics (InfAI). His research interests include knowledge graph analysis, machine learning, and pattern recognition. He was awarded the Vice-Chancellors Gold Medal for his outstanding performance during his bachelor's degree.

MD RASHAD AL HASAN RONY received the bachelor's degree from BRAC University, Bangladesh, and the master's degree from the University of Bonn, Germany, where he is currently pursuing the Ph.D. degree with the Smart Data Analysis Group. He is a Research Scientist with Fraunhofer IAIS, Dresden. He has worked on several research and industry projects related to dialogue systems, question answering, and machine reading comprehension. His primary research interests include knowledge graph-based dialogue systems, question answering systems, evaluation of generative systems, machine reading comprehension, and language models.

MOJTABA NAYYERI received the B.Sc. and M.Sc. degrees in computer engineering from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Smart Data Analysis Group, University of Bonn, Germany. His current research interests include machine learning, knowledge graphs, pattern recognition, and the semantic web.

KARISHMA MOHIUDDIN is currently pursuing the master's degree in computer science with the Smart Data Analytics (SDA) Research Group, University Bonn. She is a Student Research Assistant with Fraunhofer IAIS. Her research interests include machine learning, data analysis, and knowledge graph.

M. S. T. MAHFUJA AKTER received the master's degree in computer science from the Smart Data Analytics (SDA), University of Bonn. She has been working as a Research Assistant at Fraunhofer SCAI, since 2019. She is currently working as a Research Associate with the Meteorology Institute, University of Bonn. Her research interests include machine learning, knowledge graph analysis, and language embedding.



SAHAR VAHDATI received the M.Sc. and Ph.D. degrees in computer science from the University of Bonn. She has been a Senior Researcher and a Postdoctoral Researcher at Oxford University, U.K. She is currently leading the Nature-Inspired Machine Intelligence Research Group, Institute of Applied Informatics (InfAI), University of Leipzig. Her research interests include on using knowledge representation, analyze knowledge graphs, and artificial intelligence (AI).

JENS LEHMANN (Member, IEEE) received the master's degree in computer science from the Technical University of Dresden and the University of Bristol and the Ph.D. degree *(summa cum laude)* from the University of Leipzig. He is currently the Head of the Smart Data Analysis Research Group, a full-time Professor at the University of Bonn, and a Lead Scientist at Fraunhofer IAIS. He has contributed to various open-source projects, such as DL-Learner, SANSA, LinkedGeoData, and DBpedia. He has authored more than 100 publications, which were cited more than 18000 times. He has won 12 international awards. His research interests include semantic web technologies, question answering, machine learning, and knowledge graph analysis.