



UvA-DARE (Digital Academic Repository)

Inductive entity representations from text via link prediction

Daza, D.; Cochez, M.; Groth, P.

DOI

[10.1145/3442381.3450141](https://doi.org/10.1145/3442381.3450141)

Publication date

2021

Document Version

Final published version

Published in

The Web Conference 2021

License

CC BY

[Link to publication](#)

Citation for published version (APA):

Daza, D., Cochez, M., & Groth, P. (2021). Inductive entity representations from text via link prediction. In *The Web Conference 2021: proceedings of the World Wide Web Conference WWW 2021 : April 19-23, 2021, Ljubljana, Slovenia* (pp. 798-808). Association for Computing Machinery. <https://doi.org/10.1145/3442381.3450141>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Inductive Entity Representations from Text via Link Prediction

Daniel Daza
Vrije Universiteit Amsterdam
University of Amsterdam
Discovery Lab, Elsevier
Amsterdam, The Netherlands
d.dazacruz@vu.nl

Michael Cochez
Vrije Universiteit Amsterdam
Discovery Lab, Elsevier
Amsterdam, The Netherlands
m.cochez@vu.nl

Paul Groth
University of Amsterdam
Discovery Lab, Elsevier
Amsterdam, The Netherlands
p.groth@uva.nl

ABSTRACT

Knowledge Graphs (KG) are of vital importance for multiple applications on the web, including information retrieval, recommender systems, and metadata annotation.

Regardless of whether they are built manually by domain experts or with automatic pipelines, KGs are often incomplete. To address this problem, there is a large amount of work that proposes using machine learning to complete these graphs by predicting new links. Recent work has begun to explore the use of textual descriptions available in knowledge graphs to learn vector representations of entities in order to preform link prediction. However, the extent to which these representations learned for link prediction generalize to other tasks is unclear. This is important given the cost of learning such representations. Ideally, we would prefer representations that do not need to be trained again when transferring to a different task, while retaining reasonable performance.

Therefore, in this work, we propose a holistic evaluation protocol for entity representations learned via a link prediction objective. We consider the *inductive* link prediction and entity classification tasks, which involve entities *not* seen during training. We also consider an information retrieval task for entity-oriented search. We evaluate an architecture based on a pretrained language model, that exhibits strong generalization to entities not observed during training, and outperforms related state-of-the-art methods (22% MRR improvement in link prediction on average). We further provide evidence that the learned representations transfer well to other tasks without fine-tuning. In the entity classification task we obtain an average improvement of 16% in accuracy compared with baselines that also employ pre-trained models. In the information retrieval task, we obtain significant improvements of up to 8.8% in NDCG@10 for natural language queries. We thus show that the learned representations are not limited KG-specific tasks, and have greater generalization properties than evaluated in previous work.

CCS CONCEPTS

• **Computing methodologies** → **Statistical relational learning**; • **Information systems** → **Information retrieval**.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-8312-7/21/04.

<https://doi.org/10.1145/3442381.3450141>

KEYWORDS

knowledge graphs, entity representations, link prediction, entity classification, information retrieval

ACM Reference Format:

Daniel Daza, Michael Cochez, and Paul Groth. 2021. Inductive Entity Representations from Text via Link Prediction. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3442381.3450141>

1 INTRODUCTION

Knowledge graphs provide a structured way to represent information in the form of entities and relations between them [12]. They have become central to a variety of tasks in the Web, including information retrieval [6, 13], question answering [19, 43], and information extraction [4, 14, 26]. Many of these tasks can benefit from distributed representations of entities and relations, also known as embeddings.

A large body of work in representation learning in KGs [28, 45] is based on the optimization of a link prediction objective, which results in embeddings that model relations in a vector space. These approaches are often touted as an alternative to logic-based systems for inference in incomplete KGs, as they can assign a score to missing links [10, 15]. They have also been proposed for implementing approximate forms of reasoning for the Semantic Web [18]. However, by design, some of these methods can only compute predictions involving entities observed during training. This results in approaches that fail when applied to real-world, dynamic graphs where new entities continue to be added.

To overcome this challenge, we look to make use of the textual information within KGs. KGs like YAGO [38], DBpedia [1], and industry-deployed KGs [9, 30], contain rich textual attributes about entities such as names, dates, and descriptions [12]. Therefore, it seems reasonable to assume that for real-world applications, attribute data such as entity descriptions are readily available.

From this perspective, methods that treat KGs merely as a collection of nodes and labeled links are needlessly discarding a valuable source of information. Previous work has proposed to use textual descriptions for learning entity representations [37, 46, 50, 52], which results in a much more flexible approach, since entity representations are computed as a *function* of their textual description, and can thus be obtained even for entities not observed during training. Unfortunately, the evaluation protocol in these works mainly focuses on the task of link prediction, leaving other potential outcomes of such a flexible approach unexplored.

The motivation for seeking representations that generalize well is that they can be applied in a variety of settings for which they

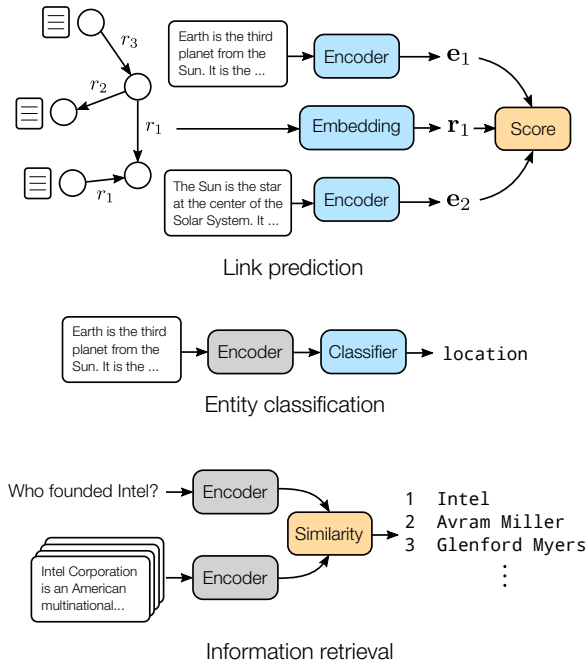


Figure 1: Overview of our work: using entity descriptions, an entity encoder is trained for link prediction in a knowledge graph (top). We show that the encoder can then be used without fine-tuning to obtain informative features for entity classification (middle) and information retrieval (bottom).

were not explicitly trained, while retaining reasonable performance. This avoids having to invest more resources on data collection, labelling, and fine-tuning when faced with a new task.

In this work, we are thus interested in the following research question: **What are the generalization properties of entity representations learned via a link prediction objective?** Our work towards answering this question results in the following contributions:

- (1) We propose the use of a pretrained language model for learning representations of entities via a link prediction objective, and study its performance in combination with four different relational models.
- (2) We propose a holistic evaluation framework for entity representations, that comprises link prediction, entity classification, and information retrieval.
- (3) We provide evidence that entity representations based on pretrained language models exhibit strong generalization properties across all tasks, outperforming the state-of-the-art, and are thus not limited to KG-specific tasks.

The rest of this paper is organized as follows. In section 2 we discuss the related work. In section 3 we introduce the need for learning inductive entity representations, and motivate the use of a pretrained language model for the task. In section 4 we describe the experiments and results for the three tasks mentioned above. Finally, we conclude and highlight directions of future work.

2 RELATED WORK

Multiple methods in the literature of representation learning in KGs propose to learn an embedding for each entity and relation in a KG. Well known examples include RESCAL [29], TransE [2], and DistMult [51], among others [28, 45]. While the state of the art in the task of link prediction continues to improve [35], most models essentially learn a lookup table of embeddings for entities and relations, and thus they are not applicable to the scenario where new entities are added to the graph.

A natural way to avoid this problem is to train *entity encoders*, that operate on a vector of entity attributes. Such encoders have been implemented using feed-forward and graph neural networks [5, 16, 23, 36]. While they can produce representations for new entities, they require fixing a set of attributes before training (e.g. bag-of-words, or numeric properties) restricting the domain of application. Furthermore, as recently proposed inductive methods [40], they can only produce representations for new entities, *using their relations with existing entities*, which is unsuitable for inductive link prediction, particularly in the challenging setting where all entities were not seen during training.

Recent work has explored using textual descriptions of entities and relations for link prediction, and proposes architectures to assign a score given the description of a relation and the entities involved in it [37, 52]. However, these architectures take as input simultaneously descriptions of entities and relations and output a score. This unavoidably mixes entity and relation representations, and prevents their transfer to other tasks such as entity classification and information retrieval.

The closest methods to our work are based on the idea of training an entity encoder with a link prediction objective. DKRL [50] consists of a convolutional neural network (CNN) that encodes descriptions. The performance of this method is limited as it does not take stop words into account, which discards part of the semantics in entity descriptions. Furthermore, its CNN architecture lags behind recent developments in neural networks for natural language processing, such as self-attention [44].

Pre-trained language models that use self-attention, such as BERT [8] have been shown to be effective at capturing similarity between texts using distributed representations [33, 49]. In KEPLER [46], the authors propose a model that uses BERT as an entity encoder, trained with an objective that combines language modeling and link prediction. The language modeling objective translates into increased training times, computing resources, and the requirement of a large corpus with long entity descriptions, of up to 512 tokens. In our work, we propose to use a pre-trained language model trained exclusively for link prediction, and obtain significant improvements at a reduced computational cost.

The evaluation protocol for both DKRL and KEPLER contains two fundamental issues that we address here. Firstly, the methods are implemented with a translational relational model [2]. However, in principle this is not necessarily the best model for any description encoder, and so it remains an open question if other models, such as multiplicative interaction models [42, 51], are better suited. We address this by considering four different relational models in our experiments, and show that a choice of model does matter.

Table 1: Examples of scoring functions for triples, proposed for TransE [2], DistMult [51], ComplEx [42], and Simple [22]. For a triple (e_i, r_j, e_k) , we denote as e_i, r_j and e_k the embeddings of its constituents (in Simple these have two parts that we indicate with indices). $\|\cdot\|_p$ indicates the p -norm; $\langle \cdot, \cdot \rangle$ is the generalized three-way dot product; $\text{Re}(\cdot)$ is the real part of a complex number; and e_k is the complex conjugate of a complex-valued vector e_k .

Model	Function
TransE	$-\ e_i + r_j - e_k\ _p$
DistMult	$\langle e_i, r_j, e_k \rangle$
ComplEx	$\text{Re}(\langle e_i, r_j, e_k \rangle)$
Simple	$\frac{1}{2} (\langle e_{i1}, r_{j1}, e_{k1} \rangle + \langle e_{i2}, r_{j2}, e_{k2} \rangle)$

Secondly, the evaluation of generalization in these works is limited. In DKRL, the entity representations are evaluated in a limited inductive link prediction setting, and in an entity classification task where entities in the test set were also used for training. In KEPLER, the authors only consider the link prediction task. In our work, we detail a more extensive evaluation framework that addresses these issues, including two different formulations of the inductive setting for link prediction.

3 INDUCTIVE ENTITY REPRESENTATIONS

We define a *knowledge graph with entity descriptions* as a tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$, consisting of a set of entities \mathcal{E} , relation types \mathcal{R} , triples \mathcal{T} , and entity descriptions \mathcal{D} . Each triple in \mathcal{T} has the form (e_i, r_j, e_k) , where $e_i \in \mathcal{E}$ is the head entity of the triple, $e_k \in \mathcal{E}$ is the tail entity, and $r_j \in \mathcal{R}$ is the relation type. For each entity $e_i \in \mathcal{E}$, there exists a description $d_{ei} = (w_1, \dots, w_n) \in \mathcal{D}$, where all w_i are words in a vocabulary \mathcal{V} .

For an entity $e_i \in \mathcal{E}$, we denote its embedding as a vector $e_i \in \mathbb{R}^d$, and similarly $r_j \in \mathbb{R}^d$ for the embedding of a relation $r_j \in \mathcal{R}$, where d is the dimension of the embedding space. We consider the problem of optimizing the embeddings of entities and relations in the graph via link prediction, so that a scoring function $s(e_i, r_j, e_k)$ assigns a high score to all observed triples $(e_i, r_j, e_k) \in \mathcal{T}$, and a low score to triples not in \mathcal{T} . This can be achieved by minimizing a loss function such as a margin-based loss [2, 3],

$$\sum_{(e_i, r_j, e_k) \in \mathcal{T}} \max(0, 1 - s(e_i, r_j, e_k) + s(e'_i, r_j, e'_k)), \quad (1)$$

where e'_i and e'_k are embeddings for an unobserved *negative triple* $(e'_i, r_j, e'_k) \notin \mathcal{T}$. Other suitable loss functions include the binary and multi-class cross-entropy [21, 42].

In general, for each triple in the KG, these loss functions can be written in the form $\mathcal{L}(s_p, s_n)$, as a function of the score s_p for a positive triple, and s_n for a negative triple. We list some of the scoring functions proposed in the literature in Table 1.

The previous optimization objective is pervasive in *transductive* methods for representation learning in knowledge graphs [28, 35, 45], which are limited to learning representations for entities in a fixed set \mathcal{E} . In these methods, the entity and relation embeddings are optimized when iterating through the set of *observed* triples.

Therefore, by design, *prediction at test time is impossible for entities not seen during training*.

We can circumvent this limitation by leveraging statistical regularities present in the description of entities [37, 46, 47, 50]. This can be realized by specifying a parametric *entity encoder* f_θ that maps the description d_{ei} of an entity to a vector $e_i = f_\theta(d_{ei}) \in \mathbb{R}^d$ that acts as the embedding of the entity. The learning algorithm is then carried out as usual, by optimizing the parameters θ of the entity encoder and the relation embeddings $r_j \forall r_j \in \mathcal{R}$, with a particular score and loss function. This process allows the encoder to learn *inductive entity representations*, as it can embed entities not seen during training, as long as they have an associated description.

3.1 BERT for entity descriptions

Transformer networks [44] have been shown to be powerful encoders that map text sequences to contextualized vectors, where each vector contains information about a word in context [11]. Furthermore, pre-trained language models like BERT [8], which have been optimized with large amounts of text, allow fine-tuning the encoder for a different task that benefits from the pre-training step.

We select BERT for the entity encoder in our method, but other pre-trained models based on Transformers are equally applicable. Note that unlike DKRL [50], this entity encoder is well suited for inputs in natural language, rather than processed inputs where stop words have been removed. We expect accepting raw inputs helps the encoder better capture the semantics needed to learn more informative entity representations.

Given an entity description $d_{ei} = (w_1, \dots, w_n)$, the encoder first adds special tokens [CLS] and [SEP] to the beginning and end of the description, respectively, so that the input to BERT is the sequence $\hat{d}_{ei} = ([CLS], w_1, \dots, w_n, [SEP])$. The output is a sequence of $n + 2$ contextualized embeddings, including those corresponding to the added special tokens:

$$\text{BERT}(\hat{d}_{ei}) = (\mathbf{h}_{\text{CLS}}, \mathbf{h}_1, \dots, \mathbf{h}_n, \mathbf{h}_{\text{SEP}}). \quad (2)$$

Similarly as in works that employ BERT for representations of text [33, 46], we select the contextualized vector $\mathbf{h}_{\text{CLS}} \in \mathbb{R}^h$, where h is the hidden size of BERT. This vector is then passed through a linear layer that reduces the dimension of the representation, to yield the output entity embedding $e_i = \mathbf{W}\mathbf{h}_{\text{CLS}}$, where $\mathbf{W} \in \mathbb{R}^{d \times h}$ is a parameter.

For relation embeddings, we use randomly initialized vectors $r_j \in \mathbb{R}^d$, for each $r_j \in \mathcal{R}$. We then apply stochastic gradient descent to optimize the model for link prediction: for each positive triple $(e_i, r_j, e_k) \in \mathcal{T}$ we compute a positive score s_p . By replacing the head or the tail with a random entity, we obtain a *corrupted* negative triple, for which we compute a score s_n . The loss is calculated as a function of s_p and s_n . This approach is quite general and admits different loss and scoring functions. The complete procedure is presented in Algorithm 1.

Note that our proposed algorithm is fundamentally different from KEPLER [46], which is trained with an additional language modeling objective that is more expensive to compute, and requires more training data.

Algorithm 1: Learning inductive entity representations via link prediction

Input: Knowledge graph $(\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$, entity encoder f_θ with parameters θ , learning rate η , scoring function s , loss function \mathcal{L}

$\mathbf{r}_j \leftarrow$ initialize randomly for each $r_j \in \mathcal{R}$;

$\theta := \{\theta\} \cup \{\mathbf{r}_j | r \in \mathcal{R}\}$;

for $(e_i, r_j, e_k) \in \mathcal{T}$ **do**

$(e'_i, r_j, e'_k) \leftarrow$ corrupt (e_i, r_j, e_k) ;

$s_p \leftarrow s(f_\theta(d_{ei}), \mathbf{r}_j, f_\theta(d_{ek}))$;

$s_n \leftarrow s(f_\theta(d_{eiv}), \mathbf{r}_j, f_\theta(d_{ek'}))$;

$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(s_p, s_n)$

end

return θ

3.2 Computational complexity

A significant portion of the cost in Algorithm 1 comes from the entity encoder. Encoding a sequence of length n with BERT has a complexity of $O(n^2)$, thus the time complexity for training is $O(|\mathcal{T}|n^2)$. In practice, a fixed value of n can be selected (such as 32 or 64 in our experiments), so if we consider it equal for all entities, the algorithm remains linear with respect to the number of triples in the graph, up to a constant factor.

At test time, the embeddings for all entities can be pre-computed. In this case, link prediction for a given entity and relation is linear in the number of entities in the graph, and the entity encoder is only applied to new entities.

3.3 Theoretical motivation

Multiple models for modeling relations in KGs have been proposed as a form of factorization [24, 28], by representing the relations in the graph as a third-order tensor $\mathbf{Y} \in \{0, 1\}^{|\mathcal{E}| \times |\mathcal{R}| \times |\mathcal{E}|}$, where the entry $y_{ijk} = 1$ if $(e_i, r_j, e_k) \in \mathcal{T}$, and $y_{ijk} = 0$ otherwise. For each $r_j \in \mathcal{R}$, $\mathbf{E} \mathbf{R}_j \mathbf{E}^\top$ is a *tensor decomposition* of \mathbf{Y} , where $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times d}$ and $\mathbf{R}_j \in \mathbb{R}^{d \times d}$, and the i -th row of \mathbf{E} contains the embedding \mathbf{e}_i for $e_i \in \mathcal{E}$. Examples of models optimized for this kind of decomposition are RESCAL [29], DistMult [51], ComplEx [42], and SimpleE [22].

For an entity description $d_{ei} = (w_1, \dots, w_n)$, let $\mathbf{W}_{ei} \in \mathbb{R}^{d \times n}$ be a matrix of word embeddings, with the embedding of word w_j in the j -th column. Approximating such a decomposition with an entity encoder thus requires correctly mapping \mathbf{W}_{ei} to the embedding of the entity \mathbf{e}_i in \mathbf{E} . In a recent result, Yun et al. [53] show that Transformers are universal approximators of continuous functions with compact support¹ $g : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$. Therefore, if such a function exists so that $g(\mathbf{W}_{ei}) = \mathbf{e}_i$, there is a Transformer that can approximate the corresponding tensor decomposition. While the existence of this function is not obvious, it further motivates an empirical study on the use of BERT for entity embeddings in a KG.

¹Their results are shown for functions whose range is $\mathbb{R}^{d \times n}$, but here we state a special case where we select one column from the output.

Table 2: Statistics of datasets used in the link prediction task.

	WN18RR	FB15k-237	Wikidata5M
Relations	11	237	822
Training			
Entities	32,755	11,633	4,579,609
Triples	69,585	215,082	20,496,514
Validation			
Entities	4,094	1,454	7,374
Triples	11,381	42,164	6,699
Test			
Entities	4,094	1,454	7,475
Triples	12,087	52,870	6,894

4 EVALUATING ENTITY REPRESENTATIONS

The use of an entity encoder to obtain entity representations is a more flexible approach that would be useful not just for link prediction, but for other tasks that could benefit from a vector representation that is a function of the textual description of an entity.

To better explore the potential of such an approach, we propose an evaluation framework that comprises inductive link prediction, inductive entity classification, and information retrieval for entity-oriented search. We present the results of the encoder proposed in section 3, and compare with recently proposed methods for each task. Our implementation and all the datasets that we use are made publicly available².

4.1 Link prediction

Link prediction models can be evaluated via a ranking procedure [3], using a test set of triples \mathcal{T}' disjoint from the set of training triples \mathcal{T} . For a test triple (e_i, r_j, e_k) , a prediction for the tail is evaluated by replacing e_k with an entity \hat{e}_k in a set of incorrect candidates $\hat{\mathcal{E}}$, and a score is computed as $s(\mathbf{e}_i, \mathbf{r}_j, \hat{\mathbf{e}}_k)$. Ideally, all the scores for incorrect candidates should be lower than the score of the correct triple. A prediction for the head is evaluated similarly, by replacing e_i .

In the transductive setting, the entities in a test triple are assumed to be in the set of training entities. Furthermore, the set of incorrect candidates is the same as the set of training entities. In the inductive setting, we instead consider a test graph $(\mathcal{E}', \mathcal{R}', \mathcal{T}', \mathcal{D}')$. The sets of triples \mathcal{T}' and \mathcal{T} are disjoint, and for relations, we always assume that $\mathcal{R}' \subseteq \mathcal{R}$. According to the way the set of incorrect candidates $\hat{\mathcal{E}}$ is determined, we define two inductive evaluation scenarios:

Dynamic evaluation. In a test triple, a new entity may appear at the head, tail, or both positions. The set of incorrect candidates $\hat{\mathcal{E}}$ is the union of training and test entities, $\mathcal{E} \cup \mathcal{E}'$. This represents a situation in which new entities are added to the KG, and is challenging as the set of incorrect candidates is larger at test time than at training.

²<https://github.com/dfdazac/blp>

Transfer evaluation. In a test triple, both entities at the head and tail position are new, and the set of incorrect candidates $\hat{\mathcal{E}}$ is \mathcal{E}' , where \mathcal{E}' is disjoint from the training set of entities \mathcal{E} . This represents a setting where we want to perform link prediction within a subset of entities, that was not observed during training. Such a situation is of interest, for example, when transferring a trained model to a specific sub-domain of new entities.

We consider that both scenarios are highly relevant for tasks of link prediction in incomplete KGs, in contrast with previous works that only consider one or the other [46, 50].

4.1.1 Experiments.

Datasets. We make use of FB15k-237 [41] and WN18RR [7], which are datasets widely used in the link prediction literature. To obtain entity descriptions we employ the datasets made available by Yao et al. [52]. FB15k-237 is a subset of Freebase, where most entities correspond to people, movies, awards, and sport teams. Descriptions were obtained from the introduction section of the Wikipedia page of each entity. In WN18RR each entity corresponds to a word sense, and descriptions are their definitions. Instead of using the conventional splits used for these datasets, we implement a *dynamic evaluation* scenario. We select 10% of entities and their associated triples to form a test graph, 10% for validation, and the remaining 80% for training. At test time, all entities are used as incorrect candidates. For these datasets we choose a maximum length of entity descriptions of 32 tokens. This value was chosen as using more tokens did not bring significant improvements, while increasing monotonically the time required for training (we include details for these results in Appendix C).

For the *transfer evaluation*, we present results on Wikidata5M, with the splits provided by Wang et al. [46]. The graph is a subset of Wikidata, containing 4.6 million entities, and descriptions from the introduction section of Wikipedia. To further experiment with the scalability of our approach, we increased the description length to 64 tokens. Dataset statistics are listed in Table 2.

Experimental setup. Following the definition in subsection 3.1, we implement an entity encoder using the BERT-base configuration from the Transformers library [48], followed by a linear layer with 128 output units.

We study the performance of our method in combination with four relational models: TransE, DistMult, ComplEx, and Simple. We aim to cover early translational and multiplicative models (i.e. TransE and DistMult) as well as more recent models that have been shown to result on state-of-the-art performance for link prediction [35] (i.e. ComplEx and Simple). We denote the resulting models as BERT for Link Prediction (BLP) followed by the employed relational model (e.g. BLP-TransE).

As a baseline we consider DKRL, proposed by Xie et al. [50]. In our implementation of DKRL, we use GloVe embeddings [31] with a dimension of 300 for the input, and an output dimension of 128. We also reproduce their Bag-Of-Words (BOW) baseline, in which an entity is encoded as the average of embeddings of words in the description. We denote these models as GloVe-DKRL and GloVe-BOW, respectively. Following recent works on the properties and applications of static embeddings from the input layer of BERT

[20, 32], we also consider variants of the baselines that use context-insensitive BERT Embeddings (BE). We denote these as BE-DKRL and BE-BOW.

For all models, we run grid search using FB15k-237, and we select the hyperparameters with the best performance on the validation set. We reuse these hyperparameters for training with WN18RR and Wikidata5M, as we found that they performed well on these datasets.

For BLP models, in the grid search we consider the following – loss function: {margin, negative log-likelihood}, learning rate: {1e-5, 2e-5, 5e-5}, L2 regularization coefficient: {0, 1e-2, 1e-3}. We use the Adam optimizer with a learning rate decay schedule with warm-up for 20% of the total number of iterations. We train for 40 epochs with a batch size of 64 with WN18RR and FB15k-237, and 5 epochs with a batch size of 1,024 with Wikidata5M.

For the BOW and DKRL baselines the values for grid search were the following – learning rate: {1e-5, 1e-4, 1e-3}, L2 regularization coefficient: {0, 1e-2, 1e-3}. We use Adam with no learning rate schedule, and we train the models for 80 epochs with a batch size of 64 with WN18RR and FB15k-237, and 10 epochs with a batch size of 1,024 with Wikidata5M.

The number of negative samples is 64 in all experiments.

Metrics. Given the score for a correct triple, and the corresponding set of negative scores obtained by replacing the head of the triple by an incorrect candidate, we sort them to obtain a ranked list. Let r_{th} be the position of the correct triple in the rank. The *reciprocal rank* is then $1/r_{th}$. This procedure is repeated by replacing the tail, to obtain the reciprocal rank $1/r_{tt}$. The Mean Reciprocal Rank is the mean of these two values, averaged across all triples in the knowledge graph:

$$\text{MRR} = \frac{1}{2|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\frac{1}{r_{th}} + \frac{1}{r_{tt}} \right) \quad (3)$$

The Hits at 1 metric (H@1) is obtained by counting the number of times the correct triple appears at position 1, and averaging as for the MRR. The H@3 and H@10 are computed similarly, considering the first 3 and 10 positions, respectively.

When scoring candidates for a given triple, we consider the *filtered* setting [2] where for each triple we consider as incorrect candidates all entities in the set $\hat{\mathcal{E}}$ minus those that would result in a correct triple, according to the training, validation, and test sets.

For more specific technical details on datasets and training, we refer the reader to Appendices A and B.

4.1.2 Results. We report the Mean Reciprocal Rank (MRR), and Hits at 1, 3, and 10, on the test set in Table 3. For reference, we also show results reported by Wang et al. [46] for KEPLER. We observe that in both the dynamic evaluation (WN18RR and FB15k-237) and the transfer evaluation (Wikidata5M), BLP-TransE consistently outperforms all the baselines across all metrics.

We note that TransE results in higher link prediction performance with BLP, compared to alternatives like DistMult, ComplEx, and Simple in WN18RR and FB15k-237. ComplEx and Simple improve on performance in Wikidata5M, which contains around two orders of magnitude more triples for training. This suggests that more elaborate relational models might be less data efficient compared with TransE, when used with BERT for link prediction. We

Table 3: Results of filtered metrics for link prediction involving entities not seen during training. We use WN18RR and FB15k-237 for dynamic evaluation, and Wikidata5M for the transfer evaluation (see text for more details). Results for KEPLER are reported by Wang et al. [46].

Method	WN18RR				FB15k-237				Wikidata5M			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
GloVe-BOW	0.170	0.055	0.215	0.405	0.172	0.099	0.188	0.316	0.343	0.092	0.531	0.756
BE-BOW	0.180	0.045	0.244	0.450	0.173	0.103	0.184	0.316	0.362	0.082	0.586	0.798
GloVe-DKRL	0.115	0.031	0.141	0.282	0.112	0.062	0.111	0.211	0.282	0.077	0.403	0.660
BE-DKRL	0.139	0.048	0.169	0.320	0.144	0.084	0.151	0.263	0.322	0.097	0.474	0.720
KEPLER	–	–	–	–	–	–	–	–	0.402	0.222	0.514	0.730
BLP-TransE	0.285	0.135	0.361	0.580	0.195	0.113	0.213	0.363	0.478	0.241	0.660	0.871
BLP-DistMult	0.248	0.135	0.288	0.481	0.146	0.076	0.156	0.286	0.472	0.242	0.646	0.869
BLP-ComplEx	0.261	0.156	0.297	0.472	0.148	0.081	0.154	0.283	0.489	0.262	0.664	0.877
BLP-Simple	0.239	0.144	0.265	0.435	0.144	0.077	0.152	0.274	0.493	0.289	0.639	0.866

note that the gap in performance between BLP-TransE and baselines is larger in WN18RR than in FB15k-237. We hypothesize that the definitions of words in WN18RR can have subtle variations of syntax that a BERT encoder captures better, while entities in FB15k-237 can be more easily identified by keywords, so that ignoring syntax does not result in a large drop in performance.

Interestingly, we observe that in Wikidata5M, KEPLER results in lower performance despite using a joint training objective that combines language modeling and link prediction. While it has been suggested that such an objective leads to increased performance [46], here we observe that this is not the case: all BLP variants outperform KEPLER when using only the link prediction objective, which also results in a reduced computational cost during training.

Despite our best efforts, we could not find a DKRL model that performed better than BOW models. This is surprising since unlike DKRL, BOW models do not take word order into account. Interestingly, for both BOW and DKRL, BE models yields consistently better results than models using GloVe embeddings, while BE models use 80% less parameters due to the use of WordPiece embeddings.

This can be attributed to differences in the data used to pre-train the embeddings, but more importantly, to the size of the embeddings: the size of BERT and GloVe embeddings is 768 and 300, respectively, which translates into a larger number of parameters in BE models. While GloVe uses one embedding per word and a vocabulary size of 400,000, BE models use wordpiece embeddings and a vocabulary size of 30,000. This means that BE models manage to reduce the number of parameters 80% in comparison with GloVe models, while resulting in better link prediction performance.

Effect of training set size. An important question in the inductive setting we consider is: what is the effect of the number of entities seen during training, on the performance on a fixed test set? To answer this question, we use FB15k-237 and the same test set, and sample subsets for training with an increasing number of entities. We evaluate the MRR for BE-BOW and BLP-TransE. The results are shown in Figure 2. We note that a reduction of 50% in the number of entities results in approximately a 27% reduction in MRR. The mismatch between these percentages suggests that the pre-trained embeddings in BE-BOW and the architecture of BLP-TransE allow

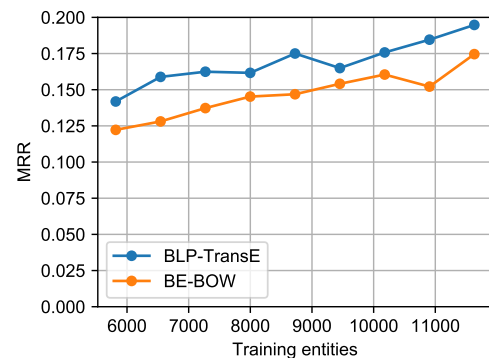


Figure 2: Inductive link prediction performance (MRR) versus number of entities used for training.

them to retain performance when reducing the size of the training set. Figure 2 also reveals in greater detail the constant gap between BE-BOW and BLP-TransE. Since both methods share the same pre-training regime and thus the data used during pre-training, we attribute the difference to the more powerful encoder used in BLP-TransE and the fact that it does not require dropping stopwords.

Transductive link prediction. Entity representations learned from entity descriptions can also be applied to the transductive scenario, where all entities at test time have been observed during training. This correspond to the setting where previous transductive methods for KG embeddings have been designed [2, 3, 22, 42, 51], although they cannot be applied to our experiments where textual descriptions and natural language are used. For reference, we include results in Appendix D, where we note that transductive methods significantly outperform description based encoders. We attribute this to the more challenging problem faced by description encoders: they must learn a complicated function from words to an entity representation, while transductive methods learn a lookup table with one embedding per entity and relation. Nonetheless, we

stress that the applicability of description based encoders is much broader, as our work demonstrates.

4.2 Entity classification

A good description encoder must learn to extract the most informative features of an entity from its description, and compress them in the entity embedding. We test this property by using the embeddings of entities trained for link prediction, as features for a logistic regression classifier. Crucially, we maintain the inductive setting, keeping the splits from the link prediction experiments. Thus, at test time the classifier is evaluated on entities that the entity encoder did not see during training.

4.2.1 Experiments.

Datasets. We evaluate entity classification using WN18RR and FB15k-237. In WN18RR we use the part of speech as the class for entities, which results in a total of 4 classes. For FB15k-237 we follow a procedure similar to Xie et al. [50] by selecting the 50 most common entity types as classes.

Experimental setup. Using entity embeddings as features, we train a multi-class logistic regression classifier with L2 regularization. The regularization coefficient is chosen from $\{1e-4, 1e-3, 1e-2, 0.1, 1, 10\}$, and we keep the coefficient resulting in the best accuracy on the validation set. We also train classifiers with features not explicitly trained for link prediction: in GloVe-avg and BE-avg we use the average of GloVe and context-insensitive BERT embeddings, respectively. SBERT [33] is a model based on BERT that is trained to learn representations of sentences, that we apply to entity descriptions. We use their publicly available trained models³ of the SBERT-NLI-base and SBERT-NLI-large variants.

Metrics. We report classification accuracy and its balanced version. The balanced accuracy weights each sample with the inverse prevalence of the true class, and allows us to identify when a classifier works better in average across classes, rather than performing better on the majority class.

4.2.2 Results. We list the results in Table 4. We observe a drastic increase in performance with all BLP models trained for link prediction, which is especially noticeable when evaluating the balanced accuracy. The marked improvements in this metric demonstrate that the embeddings are a more informative representation that allows the classifier to perform better on classes for which there is very little data, and for entities not seen during training.

Interestingly, we note that i) the baselines not trained for link prediction perform better than the BOW and DKRL baselines in most cases, and ii) SBERT models still underperform BLP models trained for link prediction. We conclude that it is the combination of a powerful BERT encoder and a link prediction fine-tuning procedure that gives rise to better entity representations.

4.3 Information retrieval

An entity can be associated with different descriptions, that can be ambiguous and not necessarily grammatical. To evaluate the robustness of an entity encoder against this variability, we test its

Table 4: Accuracy for the entity classification experiments. Raw values correspond to the regular definition of accuracy. In the balanced case (Bal.), each sample is weighted with the inverse prevalence of its true class.

Method	WN18RR		FB15k-238	
	Raw	Bal.	Raw	Bal.
GloVe-avg	90.3	55.3	82.0	35.0
BE-avg	92.7	62.1	82.4	39.4
SBERT-NLI-base	96.3	66.5	84.5	36.6
SBERT-NLI-large	96.3	67.1	83.8	35.1
GloVe-BOW	91.5	56.0	82.9	34.4
BE-BOW	93.3	60.7	83.1	28.3
GloVe-DKRL	91.2	55.5	81.1	26.6
BE-DKRL	90.0	48.8	81.6	30.9
BLP-TransE	99.1	81.5	85.4	42.5
BLP-DistMult	99.5	78.5	84.3	41.0
BLP-Complex	99.3	78.1	85.1	38.1
BLP-Simple	99.2	83.0	85.8	45.7

performance in an information retrieval task: given a query about an entity, return a list of documents (entity descriptions) ranked by relevance.

4.3.1 Experiments.

Dataset. DBpedia-Entity v2 is a dataset for the evaluation of information retrieval (IR) systems for entity-oriented search, introduced by Hasibi et al. [17]. The document index corresponds to textual descriptions of entities in DBpedia. There are 467 queries, categorized into 4 types – *SemSearch ES*: short and ambiguous queries, e.g. “john lennon, parents”; *INEX-LD*: keyword queries, e.g. “bicycle holiday nature”; *List Search*: queries seeking for lists, e.g. “Airports in Germany”; and *QALD-2*: questions in natural language, e.g. “What is the longest river?”. For each query, there is a list of documents graded by relevance by crowd workers. In average, there are 104 graded documents per query.

Experimental setup. Similarly as in previous work on embeddings for information retrieval [13], we implement a re-ranking procedure, by updating a list of document scores assigned by an existing IR system (e.g. BM25). Let q be query, d_e the text in a document, and z_{IR} the score the IR system assigned to d_e given q . We use an entity encoder f_θ to compute the similarity between the embeddings of the query and the document, via their inner product:

$$z_{new} = \alpha f_\theta(q)^\top f_\theta(d_e) + (1 - \alpha)z_{IR}. \quad (4)$$

We select the best value of α via grid search on each of 5 training folds of the data provided by Hasibi et al. [17], and report the average test fold performance. For the grid we consider 20 evenly spaced values in the interval $[0, 1]$, and for the entity encoder we use the models trained for link prediction with Wikidata5M. As in entity classification, we do not fine-tune the entity encoders. To obtain the base scores z_{IR} , we use BM25F-CA [34], as it is one of the best performing methods on the DBpedia-Entity v2 dataset reported by Zhiltsov et al. [54].

³<https://github.com/UKPLab/sentence-transformers>

Table 5: NDCG results for the information retrieval task, across different query types. We show the results for BM25F-CA, followed by the results after re-ranking with different entity encoders. Values in bold indicate that the difference between BM25F-CA and the re-ranked results is statistically significant at $p < 0.05$.

Method	SemSearch ES		INEX-LD		ListSearch		QALD-2		All	
	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
BM25F-CA	0.628	0.720	0.439	0.530	0.425	0.511	0.369	0.461	0.460	0.551
+ GloVe-BOW	0.631	0.721	0.449	0.544	0.432	0.518	0.368	0.460	0.462	0.554
+ BE-BOW	0.629	0.721	0.458	0.546	0.431	0.522	0.377	0.469	0.460	0.552
+ GloVe-DKRL	0.624	0.719	0.440	0.529	0.424	0.516	0.368	0.468	0.459	0.550
+ BE-DKRL	0.627	0.720	0.436	0.530	0.435	0.525	0.374	0.466	0.459	0.553
+ BLP-TransE	0.631	0.723	0.446	0.546	0.442	0.540	0.401	0.482	0.472	0.562
+ BLP-DistMult	0.631	0.722	0.458	0.550	0.442	0.536	0.397	0.480	0.468	0.560
+ BLP-CompLEx	0.628	0.721	0.454	0.548	0.430	0.528	0.405	0.486	0.468	0.561
+ BLP-SimplE	0.628	0.721	0.454	0.552	0.439	0.527	0.399	0.477	0.464	0.557

Metrics. The DBpedia-Entity v2 dataset comes with a set of relevant documents for different queries. Each relevant document contains a grade of relevance, where 0 is non-relevant, 1 is relevant, and 2 is highly-relevant. This allows to compute how good the output ranking of a system is given the relevance of the documents it produces.

Assume we have a list of k documents ranked by a system, and the relevance of the document at position i is r_i . The Discounted Cumulative Gain (DCG) is defined as

$$DCG@k = \sum_{i=1}^k \frac{2^{r_i} - 1}{\log_2(1 + i)}. \quad (5)$$

Let $IDCG@k$ be the maximum $DCG@k$, produced by sorting *all* relevant documents in the corpus and computing eq. 5. The Normalized DCG (NDCG) is defined as

$$NDCG@k = \frac{DCG@k}{IDCG@k}. \quad (6)$$

4.3.2 Results. We report NDCG at 10 and 100 in Table 5. We show the results obtained with BM25F-CA, followed by the results after re-ranking with a particular encoder. BLP-TransE yields statistically significant improvements (according to a two-tailed t-test) in both NDCG at 10 and 100, when considering all query types.

Although on average the entity encoders yield higher performance across all query types, the difference with BM25F-CA is not statistically significant for SemSearch ES queries. This is expected since these queries are short and often not grammatical, differing from the entity descriptions used to train the encoders. In other query types the encoders show significant improvements, especially in QALD-2 queries, containing well formed questions.

Depending on the type of query, we observed that the optimal parameter α was always between 0.1 and 0.7, indicating that the embeddings of queries and documents alone are not enough to correctly rank documents, and a fraction of the score assigned by BM25F-CA is still required to preserve retrieval quality. However, the results of this section are encouraging in the context of representation learning of entities, as they show that the entity encoders have learned a function that maps entities and queries about them to vectors that are close in the embedding space.

Table 6: First five documents retrieved by BM25F-CA, and its reranking with BLP-TransE, for two example queries.

Query: Who founded Intel?		
Rank	BM25F-CA	+ BLP-TransE
1	Intel	Intel
2	Intel 8253	Avram Miller
3	Intel 8259	Glenford Myers
4	Intel Play	Intel Play
5	Intel Ct	Leslie L. Vadasz
Query: A list of all American inventions		
Rank	BM25F-CA	+ BLP-TransE
1	US inventions	US inventions
2	The Mothers of Invention	US inventions (after 1991)
3	American Heritage of Inv.	US inventions (1890 - 1945)
4	Invention	Timeline of inventions
5	Sandy Bull	US inventions (1946-91)

We present a sample of an original ranking by BM25F-CA, and the reranked results by BLP-TransE in Table 6. In the first example, while BM25F-CA retrieves Intel-related companies and products, BLP-TransE fixes the ranking by pushing down products and increasing the scores for persons relevant to the query. In the second query, BM25F-CA retrieves more generic topics related to “invention”, plus irrelevant documents at positions 2 and 5. BLP-TransE keeps only the first document and increases the scores of other, more relevant documents.

5 CONCLUSION

We have studied the generalization properties of entity representations learned via link prediction, by proposing the use of a pre-trained model fine-tuned with a link prediction, and an extensive evaluation framework that also considers tasks of entity classification and information retrieval. We find that when using the proposed model, the entity representations are inductive, as they

exhibit strong generalization in link prediction for entities not seen during training.

Without requiring fine-tuning, the entity representations learned via link prediction are also transferable to the tasks of node classification and information retrieval, which demonstrates that the entity embeddings act as compressed representations of the most salient features of an entity. This provides evidence that the learned entity representations generalize to tasks beyond link prediction.

This is additionally important because having generalized vector representations of knowledge graphs is useful for using them within other tasks, for example, search and automatic metadata annotation.

We consider the inductive setting where new entities appear at test time. Extending our approach to unseen relations is an interesting avenue for future research. For future work we also consider the study and improvement of the learned entity representations, for example, by enforcing independent factors of variation in the representations [25], or by learning on hyperbolic spaces better suited for hierarchical data [27].

ACKNOWLEDGMENTS

This project was funded by Elsevier's Discovery Lab.

REFERENCES

- [1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007 (Lecture Notes in Computer Science, Vol. 4825)*, Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52
- [2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.), 2787–2795. <http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data>
- [3] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7–11, 2011*, Wolfram Burgard and Dan Roth (Eds.). AAAI Press, <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659>
- [4] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4762–4779. <https://doi.org/10.18653/v1/P19-1470>
- [5] Shaosheng Cao, Wei Lu, and Qionghai Xu. 2016. Deep Neural Networks for Learning Graph Representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 1145–1152. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12423>
- [6] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast, QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 365–374. <https://doi.org/10.1145/2600428.2609628>
- [7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 1811–1818. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366>
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [9] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani (Eds.). ACM, 601–610. <https://doi.org/10.1145/2623330.2623623>
- [10] Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. 2012. Predicting RDF Triples in Incomplete Knowledge Bases with Tensor Factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (Trento, Italy) (SAC '12)*. Association for Computing Machinery, New York, NY, USA, 326–331. <https://doi.org/10.1145/2245276.2245341>
- [11] Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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)*. Association for Computational Linguistics, Hong Kong, China, 55–65. <https://doi.org/10.18653/v1/D19-1006>
- [12] Dieter Fensel, Umütcan Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. 2020. *Knowledge Graphs - Methodology, Tools and Selected Use Cases*. Springer. <https://doi.org/10.1007/978-3-030-37439-6>
- [13] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2020. Graph-Embedding Empowered Entity Retrieval. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12035)*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer, 97–110. https://doi.org/10.1007/978-3-030-45439-5_7
- [14] Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open Knowledge Graph Embeddings. 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)*. Association for Computational Linguistics, Hong Kong, China, 378–388. <https://doi.org/10.18653/v1/D19-1036>
- [15] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (Montréal, Canada) (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 2030–2041.
- [16] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 1024–1034. <http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs>
- [17] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7–11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 1265–1268. <https://doi.org/10.1145/3077136.3080751>
- [18] Pascal Hitzler, Federico Bianchi, Monireh Ebrahimi, and Md. Kamruzzaman Sarker. 2020. Neural-symbolic integration and the Semantic Web. *Semantic Web* 11, 1 (2020), 3–11. <https://doi.org/10.3233/SW-190368>
- [19] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11–15, 2019*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 105–113. <https://doi.org/10.1145/3289600.3290956>
- [20] Ganesh Jawahar, Benoit Sagot, and Djamel Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3651–3657. <https://doi.org/10.18653/v1/P19-1356>
- [21] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge Base Completion: Baselines Strike Back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Vancouver, Canada, 69–74. <https://doi.org/10.18653/v1/W17-2609>
- [22] Seyed Mehran Kazemi and David Poole. 2018. Simple Embedding for Link Prediction in Knowledge Graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*,

- NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 4289–4300. <http://papers.nips.cc/paper/7682-simple-embedding-for-link-prediction-in-knowledge-graphs>
- [23] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016). arXiv:1611.07308 <http://arxiv.org/abs/1611.07308>
- [24] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.
- [25] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 4114–4124. <http://proceedings.mlr.press/v97/locatello19a.html>
- [26] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, Suntec, Singapore, 1003–1011. <https://www.aclweb.org/anthology/P09-1113>
- [27] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 6338–6347. <http://papers.nips.cc/paper/7213-poincare-embeddings-for-learning-hierarchical-representations>
- [28] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2015), 11–33.
- [29] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, Lise Getoor and Tobias Scheffer (Eds.). Omnipress, 809–816. https://icml.cc/2011/papers/438_icmlpaper.pdf
- [30] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: Lessons and challenges. *Queue* 17, 2 (2019), 48–75.
- [31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. <https://doi.org/10.3115/v1/d14-1162>
- [32] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1499–1509. <https://doi.org/10.18653/v1/D18-1179>
- [33] Nils Reimers and Iryna Gurevych. 2019. 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)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- [34] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. <https://doi.org/10.1561/15000000019>
- [35] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. <https://openreview.net/forum?id=BkxSmlBFvr>
- [36] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10843)*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer, 593–607. https://doi.org/10.1007/978-3-319-93417-4_38
- [37] Baoxu Shi and Tim Wenginger. 2018. Open-World Knowledge Graph Completion. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, New Orleans, Louisiana, USA, February 2-7, 2018, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 1957–1964. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16055>
- [38] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 697–706. <https://doi.org/10.1145/1242572.1242667>
- [39] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.
- [40] Komal K. Teru and William L. Hamilton. 2019. Inductive Relation Prediction on Knowledge Graphs. *CoRR* abs/1911.06962 (2019). arXiv:1911.06962 <http://arxiv.org/abs/1911.06962>
- [41] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, 57–66. <https://doi.org/10.18653/v1/W15-4007>
- [42] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, 2071–2080. <http://proceedings.mlr.press/v48/trouillon16.html>
- [43] Svitlana Vakulenko, Javier David Fernandez Garcia, Axel Polleres, Maarten de Rijke, and Michael Cochez. 2019. Message Passing for Complex Question Answering over Knowledge Graphs. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1431–1440. <https://doi.org/10.1145/3357384.3358026>
- [44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 5998–6008. <http://papers.nips.cc/paper/7181-attention-is-all-you-need>
- [45] Quan Wang, Zhen-dong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
- [46] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *CoRR* abs/1911.06136 (2019). arXiv:1911.06136 <http://arxiv.org/abs/1911.06136>
- [47] Zihao Wang, Kwunping Lai, Piji Li, Lidong Bing, and Wai Lam. 2019. Tackling Long-Tailed Relations and Uncommon Entities in 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 (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 250–260. <https://doi.org/10.18653/v1/D19-1024>
- [48] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 <http://arxiv.org/abs/1910.03771>
- [49] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot Entity Linking with Dense Entity Retrieval. *CoRR* abs/1911.03814 (2019). arXiv:1911.03814 <http://arxiv.org/abs/1911.03814>
- [50] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2659–2665. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216>
- [51] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6575>
- [52] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. *CoRR* abs/1909.03193 (2019). arXiv:1909.03193 <http://arxiv.org/abs/1909.03193>
- [53] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Are Transformers universal approximators of sequence-to-sequence functions?. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. <https://openreview.net/forum?id=ByxRMOntvr>
- [54] Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. 2015. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *Proceedings*

of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (Santiago, Chile) (SIGIR '15). Association for Computing Machinery, New York, NY, USA, 253–262. <https://doi.org/10.1145/2766462.2767756>

A TECHNICAL DETAILS

Dataset generation. The original splits of WN18RR and FB15k-237 are intended to be used in the transductive setting, where all entities in the validation and test sets are also present in the training set. We thus generate new splits for these datasets for the inductive setting. The algorithm for generating the datasets is as follows:

- (1) Sample a node from the graph.
- (2) Check that removing the node does not leave other nodes with zero neighbors.
- (3) Check that if the node is removed, the number of edges for any relation type is not lower than 100.
- (4) If conditions 2 and 3 are met, remove the node and add it to the set of nodes removed from the training set.

Negative sampling. In all link prediction experiments, negative samples are obtained by randomly replacing the head or tail with a random entity. We sample entities from the set of entities involved in a mini-batch of triples. This strategy allows to increase the number of negative samples while preserving efficiency, as this requires encoding less distinct entity descriptions per mini-batch, than if the negative samples were sampled from the complete set of entities in the graph.

B COMPUTING RESOURCES

For the experiments with WN18RR and FB15k-237 we used a workstation with an Intel Xeon processor, 1 NVIDIA GeForce GTX 1080 Ti GPU with 11GB of memory, and 60GB of RAM. For Wikidata5M we used 4 NVIDIA TITAN RTX GPUs with 24GB of memory each.

For all the BLP models we use the BERT-base configuration [8]. In particular, we use the bert-base-cased pre-trained model from the Transformers library [48]. This model has 110 million parameters. For link prediction, we additionally have parameters for relation embeddings, which have a dimension of 128. The number of parameters depends on the number of relations in the dataset. This results in 1,400 relation parameters for WN18RR, 30,300 for FB15k-237, and 105,216 for Wikidata5M.

The time required for training was 7 hours for WN18RR, 14 hours for FB15k-237, and 2 days for Wikidata5M.

C EFFECT OF DESCRIPTION LENGTH

To determine the influence of the length of the descriptions on performance, we trained several variants of BLP-TransE with FB15k-237, varying the maximum length of the entity descriptions from 16 up to 128 tokens. The resulting link prediction performance (MRR) in the validation set, and training time are shown in Figure 3. We note that using more than 32 tokens does not bring significant improvements, while monotonically increasing the time required for training.

D TRANSDUCTIVE LINK PREDICTION

Entity representations learned via link prediction can also be applied to the transductive setting, where all entities at test time have been observed during training. For completeness, we compare the

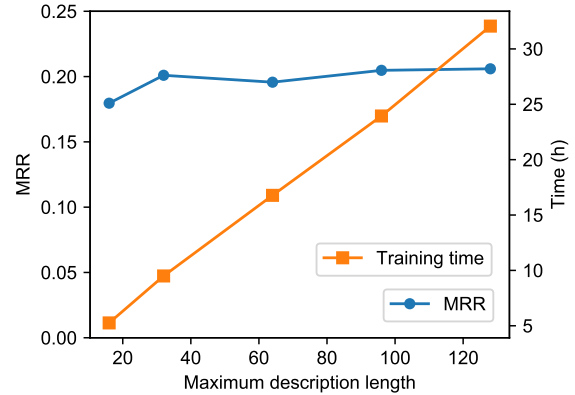


Figure 3: Inductive link prediction performance in the validation set and training time vs. maximum length of entity descriptions.

Table 7: Results of filtered metrics for link prediction in the transductive setting. *Results taken from Sun et al. [39]. The first group contains transductive methods, and the second corresponds to inductive methods that use entity descriptions.

Method	WN18RR				FB15k-237			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE*	0.226	–	–	0.501	0.294	–	–	0.465
DistMult*	0.430	0.390	0.440	0.490	0.241	0.155	0.263	0.419
ComplEx*	0.440	0.410	0.460	0.510	0.247	0.158	0.275	0.428
RotatE*	0.476	0.428	0.492	0.571	0.338	0.241	0.375	0.533
KG-BERT [52]	0.242	0.110	0.280	0.524	0.236	0.145	0.258	0.420
BE-BOW	0.193	0.039	0.265	0.503	0.232	0.152	0.250	0.394
BE-DKRL	0.222	0.047	0.323	0.558	0.215	0.135	0.231	0.379
BERT-TransE	0.325	0.144	0.431	0.679	0.235	0.150	0.253	0.411
BERT-DistMult	0.314	0.182	0.370	0.581	0.210	0.130	0.222	0.377

methods studied in our work, with previous works designed for this setting. The results are shown in Table 7.