Learning to Borrow – Relation Representation for Without-Mention Entity-Pairs for Knowledge Graph Completion

Anonymous NAACL submission

Abstract

Prior work on integrating text corpora with knowledge graphs (KGs) to improve Knowledge Graph Embedding (KGE) have obtained good performance for entities that co-occur in sentences in text corpora. Such sentences (textual mentions of entity-pairs) are represented as Lexicalised Dependency Paths (LDPs) between two entities. However, it is not possible to represent relations between entities that do not co-occur in a single sentence using LDPs. In this paper, we propose and evaluate several methods to address this problem, where we *borrow* LDPs from the entity pairs that co-occur in sentences in the corpus (i.e. *with mentions* entity pairs) to represent entity pairs that do *not* co-occur in any sentence in the corpus (i.e. *without mention* entity pairs). We propose a supervised borrowing method, *SuperBorrow*, that learns to score the suitability of an LDP to represent a without-mentions entity pair using pre-trained entity embeddings and contextualised LDP representations. Experimental results show that SuperBorrow improves the link prediction performance of multiple widely-used prior KGE methods such as TransE, DistMult, ComplEx and RotatE.

1 Introduction

Knowledge Graphs (KGs) are a structured form of information that underline the relationships between real-world entities (Ehrlinger and Wöß, 2016; Kroetsch and Weikum, 2016; Paulheim, 2017). A KG is represented using a set of relational tuples of the form (h, r, t), where r represents the relation between the head entity h and the tail entity t. For example, the relational tuple (*Joe Biden*, *president-of*, *US*) indicates that the *president-of* relation holds between *Joe Biden* and *US*. There exists a large number of publicly available and widely used KGs, such as Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), and YAGO ontology (Suchanek et al., 2007). KGs have been effectively applied in various NLP tasks such as, relation extraction (Riedel et al., 2013; Weston et al., 2013), question answering (Das et al., 2017; Sydorova et al., 2019), and dialogue systems (Xu et al., 2020). However, most KGs suffer from data sparseness as many relations between entities are not explicitly represented (Min et al., 2013).

To overcome the sparsity problem, Knowledge Graph Embedding (KGE) models learn representations (a.k.a. embeddings) for entities and relations in a given KG in a vector space, which can then be used to infer missing links between entities (Bordes et al., 2013; Nickel et al., 2015; Wang et al., 2017). Such models are trained to predict relations that are likely to exist between entities (known as link prediction or KG completion) according to some scoring formula. Although previously proposed KGE methods have shown good empirical performances for KG completion (Minervini et al., 2015), the KGEs are learnt from the KGs only, which might not represent all the relations that exist between the entities included in the KG. To overcome this limitation, prior work has used external text corpora in addition to the KGs (Toutanova et al., 2015; Xu et al., 2016; Long et al., 2016; An et al., 2018; Wang et al., 2019b,a; Lu et al., 2020). Compared to structured KGs, unstructured text corpora are abundantly available, up-to-date and have diverse linguistic expressions for extracting relational information.

The co-occurrences of two entities within sentences (a.k.a textual mentions) in a text corpus has shown its success for text-enhanced KGEs (Komninos and Manandhar, 2017; An et al., 2018). For example, the relational tuple in the Freebase KG (*Joe Biden*, *president-of*, *US*) is mentioned in the following sentence "Joseph Robinette Biden Jr. is an American politician who is the 46th and current president of the United States." This sentence expresses the *president-of* relation between the two

entities *Joe Biden* and *US*. As the entity-pair (*Joe Biden*,*US*) appears in a single sentence, we call it a *with-mention* entity-pair. However, even in a large text corpus, not every related entity pair co-occurs in a specified window, which are referred to as *without-mention* entity-pairs in previous studies. For instance, if we consider the widely used FB15K-237 KG (Toutanova et al., 2015) and the ClueWeb12 (Gabrilovich et al., 2013) text corpus with FB entity mention annotations,[1] 33% of entity-pairs in FB15k-237 do not have textual mentions within the same sentences. This sparseness problem limits the generalisation capabilities of using textual mentions for enhancing KGEs. Specifically, Toutanova et al. (2015); Komninos and Manandhar (2017) have shown larger improvements in link prediction for with-mention entity-pairs over without-mention pairs.

In this paper, we propose a method to augment a given KG with additional textual relations extracted from a corpus and represented as LDPs. The augmented KG can then be used to train *any* KGE learning method. This is attractive from both scalability and compatibility point of views because our proposal is agnostic to the KGE learning method that is subsequently used for learning KGEs. Our main contribution in this paper is to improve link prediction for without-mention entity-pairs by borrowing LDPs from with-mentions entity-pairs to overcome the sparseness in co-occurrences of the without-mentions entity-pairs. We show that learning a supervised borrowing method, *SuperBorrow*, that scores suitable LDPs to represent without-mention entity-pairs based on pre-trained entity embeddings and contextualised LDP embeddings boosts the performance of link prediction using a series of KGE methods, compared to what would have been possible without textual relations.

2 Related Work

KGEs from a Multi-relational Graph: Typically, KG embedding models consist of two main steps: (a) defining a scoring function for a tuple, and (b) learning entity and relation representations. Entities are usually represented as vectors, whereas relations can be represented by vectors (e.g. TransE (Bordes et al., 2013), DistMult (Yang et al., 2014) and ComplEx (Trouillon et al., 2016)) matrices (e.g. RESCAL (Nickel et al., 2011)), or by 3D tensors (e.g. Neural Tensor Network (Socher et al., 2013)).

[1] 200 million sentences in CluWeb12 annotated with FB entity mention annotations.

KGE method	Score function $f(h, R, t)$
TransE (Bordes et al., 2013)	$\|\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|\|_{\ell_{1/2}}$
DistMult (Yang et al., 2014)	$\langle \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \rangle$
ComplEx (Trouillon et al., 2016)	$\langle \boldsymbol{h}, \boldsymbol{r}, \bar{\boldsymbol{t}} \rangle$
RotatE (Sun et al., 2019)	$\|\|\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}\|\|^2$

Table 1: Score functions proposed in KGE methods. Entity embeddings $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$ are vectors in all models, except in ComplEx where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{C}^d$. Here, $\ell_{1/2}$ denotes either ℓ_1 or ℓ_2 norm of a vector. In ComplEx, $\bar{\boldsymbol{t}}$ is the element-wise complex conjugate.

Using some form of a representation, scoring functions are then defined to evaluate the strength of a relation r between h and t entities in a triple. TransE is one of the earliest and well-known distance-based KGE method that performs a linear translation and its scoring function is given in Table 1. Alternatively, a bilinear function is used in several KGE models, such as RESCAL, DistMult and ComplEx, for which scoring functions are defined in Table 1. KGEs are learnt such that the observed facts (positive triples) are assigned higher scores compared to that of the negative triple (for example generated by perturbing a positive instance by replacing its head or tail entities by an entity randomly selected from the set of entities) by minimising a loss function, such as the logistic loss or the margin loss.

Conventional KGE models are trained using the facts in the KGs, which are often incomplete. Therefore, to overcome the sparsity of structured KGs, we propose to integrate information from a text corpus, thereby augmenting the KG. The augmented KG is then used as the input to existing KGE methods to learn accurate entity and relation embeddings. In particular, we do not modify the scoring functions nor optimisation objectives for the respective KGE methods, which makes our proposed approach applicable in many existing KGE methods without any modifications.

Text-Enhanced KGEs: Recently, a new line of research that combines textual information with relational graphs has emerged (Lu et al., 2020). Different combination methods have been proposed for this purpose. Wang et al. (2014) proposed a model to embed both entities and words (using entity names and Wikipedia anchors) into the same low-dimensional vector space to capture relational

information from a KG and the co-occurrences from the corpus. Rosso et al. (2019) control the amount of information shared between the two data sources in the joint embedding space using regularisation. This joint model is further enhanced by incorporating entity descriptions from an external corpus, which are jointly learnt with the KG (Zhong et al., 2015; Xie et al., 2016; Veira et al., 2019). In a different scenario, the text-enhanced knowledge embedding model by Wang et al. (2016) creates a co-occurrence network of words and entities from an entity-annotated corpus. The authors define point-wise and pair-wise contexts using the co-occurrence frequencies in the network. Then, entity and relation embeddings are enhanced using textual point-wise and pair-wise embeddings, respectively. Similarly, Rezayi et al. (2021) construct an augmented KG that has nodes from external text. The original and the augmented graphs are then aligned to suppress the noise and distil relevant information. In our work, we focus on adding extra edges to the KG rather than nodes as in Rezayi et al. (2021) and Wang et al. (2016).

In addition to contextual information and textual descriptions of individual words/entities, sentences where two entities co-occur have been used as contextual evidence to learn KGEs (Toutanova et al., 2015; Komninos and Manandhar, 2017; Tang et al., 2019). For example, Toutanova et al. (2015) extracted LDPs by parsing co-occurring sentences in a text corpus, which are then used as textual relations in the KG. This model can be seen as a special case of *universal schema* (Riedel et al., 2013), which combines textual and KG relations in the same entity-pair co-occurrence matrix, subsequently decomposed to obtain entity embeddings. Komninos and Manandhar (2017) proposed a novel triple scoring function where textual mentions are used as a source of supporting evidence for a triple.

Our problem setting differs from prior work on text-enhanced KGEs in two important ways. First, we do not modify the underlying structure of the KGE method, which is attractive from both scalability and compatibility of our proposal. Second, rather than considering only entity-pairs that are occurring within a specified context in the corpus (i.e. with-mention entity-pairs), we propose to borrow LDPs from with-mention entity-pairs to overcome the data sparseness in without-mention entity-pairs that never co-occur within any sentence in the corpus.

3 Method

A relational KG $\mathcal{D}$ consists of a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$. In $\mathcal{D}$, knowledge is represented by relational tuples $(h, r, t) \in \mathcal{D}$, where the head entity h is related to the tail entity t by the KG relation r. In this work, we assume relations to be asymmetric in general (if $(h, r, t) \in \mathcal{D}$ then it does not necessarily follow that $(t, r, h) \in \mathcal{D}$). The goal is to learn representations for entities and relations such that missing tuples can be accurately inferred.

As KGs $\mathcal{D}$ are often sparse with many missing edges between entities, the learnt KGEs are affected, which in return impacts the performance of KGEs on downstream tasks such as link prediction. To address this sparseness problem, we consider the availability of a text corpus $\mathcal{T}$ where relational facts are expressed using contexts in which an entity-pair co-occurs. The textual relations that are extracted from $\mathcal{T}$ can be *injected* into $\mathcal{D}$ before applying a KGE method.

To align $\mathcal{D}$ with $\mathcal{T}$, entity linking is applied to resolve ambiguous entity mentions in the text with unique entities in the KG (Gabrilovich et al., 2013; Shen et al., 2014). Then, sentences of which two entities co-occurring with are considered as *textual mentions* of relations that exist between those entities. Assuming that the corpus is annotated using the entities in $\mathcal{D}$, there are multiple possibilities to obtain relational features of sentences that mention the entities. Following previous work (Toutanova et al., 2015), we first run a dependency parser (Chen and Manning, 2014) on each sentence in the entity-annotated corpus to obtain LDPs. Then, if $\mathcal{D}$ contains the head and tail entities of an LDP l, we insert l into $\mathcal{D}$ to form a textual triple $(h, l, t) \in \mathcal{D}$. The augmented KG is then used to learn embeddings for $\mathcal{E}$ and $\mathcal{R}$ using different KGE methods. During KGE processs, we treat both original relations in the KG and the augmented LDPs equally. In principle, any existing KGE learning method can be applied on the augmented KG as we later see in our experiments.

One obvious limitation of the above-described method is that entity-pairs that never co-occur in any contextual window (e.g. a sentence) will *not* be connected by any LDP during the augmentation process. This is fine if the two entities are truly unrelated. However, this is problematic for entities that are related but their relations were not sufficiently covered in the text corpus because of the coverage issues and small size of the corpus. As

	Relations	Entities	Triples Train/Test	w-m	w/o-m
FB	237	14,541	272,115/20,466	2,344	18,122
Text	1,100	12,930	404,009/-	-	-

Table 2: Statistics of the datasets. w-m and w/o-m denotes the number of test instances respectively in with-mention and without-mention entity-pair sets.

we later see in our evaluations (§ 5), this is indeed the case for the majority of the without-mention entity-pairs. To overcome this limitation of our proposal, next we describe a method to *borrow* LDPs from with-mention entity-pairs to without-mention entity pairs.

3.1 Learn to Borrow LDPs

Given a without-mention entity pair (h^*, t^*), we propose a supervised borrowing method **SuperBorrow** to rank LDPs that are extracted for the with mention entity-pairs from a text corpus. Given pre-trained entity representations $\boldsymbol{h}$ and $\boldsymbol{t}$, we learn an entity-pair encoder, f, parametrised by θ, to create an entity-pair representation, $f(h, t; \theta)$, for (h, t). In this work, the encoder f is implemented as a multilayer perceptron with a nonlinear activation, where the input entity-pair to the MLP is encoded as follows:

$$\boldsymbol{x} = [\boldsymbol{h} \oplus \boldsymbol{t} \oplus (\boldsymbol{h} - \boldsymbol{t}) \oplus (\boldsymbol{h} \circ \boldsymbol{t})] \tag{1}$$

Here, $\oplus$ denotes the concatenation of vectors and $\circ$ is the element-wise multiplication between two vectors. (1) considers the information in the head and tail entity embeddings independently as well as the interactions between their corresponding dimensions. The final output vector $f(h, t; \theta)$ of the MLP is treated as the representation of the entity-pair (h, t).

As an alternative to representing the relationship between two entities in an entity-pair (h, t) by $f(h, t; \theta)$ using the corresponding entity embeddings, we can use $\mathcal{L}_{(h,t)}$, the set of LDPs co-occurring between h and t (Bollegala et al., 2010). Because an LDP is a sequence of textual tokens, we can use any sentence encoder to represent an LDP by a vector. Specifically, in our experiments later we use the pretrained sentence encoder SBERT (Reimers and Gurevych, 2019) to represent an LDP, l, by a vector, $\boldsymbol{l}$.

We require LDPs that co-occur with an entity-pair (h, t) to be similar to $f(h, t; \theta)$ than LDPs that do not co-occur with (h, t). Specifically, we use the set of with-mention entity-pairs with their associated LDPs as positive training instances $\mathcal{S}_{(h,t)}$. LDPs that are associated with either h or t alone (not both) are used as negative training instances $\mathcal{S}'_{(h,t)}$ as given by (2).

$$\mathcal{S}'_{(h,t)} = \{(h, l, t) | \exists t'(h, l, t') \in \mathcal{D} \wedge t' \neq t, \\ \exists h'(h', l, t) \in \mathcal{D} \wedge h' \neq h\} \tag{2}$$

We learn the parameters of $f(h, t, \theta)$ by minimising the marginal loss over $\mathcal{S}_{(h,t)}$ and $\mathcal{S}'_{(h,t)}$ as shown in (3).

$$\sum_{(h,l,t) \in \mathcal{S}_{(h,t)}} \sum_{(h,l',t) \in \mathcal{S}'_{(h,t)}} \max\left(0, \gamma - f(h, t; \theta)^\top (\boldsymbol{l} - \boldsymbol{l'})\right) \tag{3}$$

Here, $\gamma (\geq 0)$ is the margin and is set to 1 in our experiments. To determine which LDPs to be borrowed for a particular without-mention entity pair, (h^*, t^*), we first compute its representation, $f(h^*, t^*; \theta)$ using the θ found by minimising (3) above. We then score each LDP, l, using the sentence encoder model, by the inner-product, $f(h^*, t^*; \theta)^\top \boldsymbol{l}$. We then select the top-k LDPs with the highest inner-products with $f(h^*, t^*; \theta)$ to augment the KG. The number of borrowed LDPs (k) is a hyperparameter that is tuned using the validation triples selected from the KG.

4 Experimental Setup

4.1 Dataset and Training Details

Datasets: We use FB15k237 as the KG and ClueWeb12[2] as the corpus for extracting LDPs for the entity-pairs in the FB157k237 KG. Specifically, we use the textual triples consisting of LDPs that are extracted and made available[3] by Toutanova et al. (2015). The number of extracted unique LDPs and textual triples in this dataset are respectively $2,740,176$ and $3,978,014$. To make the training of KGE methods computationally efficient, we filter out LDPs that occur in less than 100 distinct entity-pairs in the corpus. The FB15k237 test set is split into with-mention (i.e. entity-pairs that co-occur in some LDP) and without-mention (i.e entity-pairs that do not co-occur in any LDP) sets as shown in Table 2. According to Table 2, there are 88.14% without-mentions entity-pairs in the

[2] https://lemurproject.org/clueweb12/

[3] https://www.microsoft.com/en-us/download/details.aspx?id=52312

	overall					with-mention					without-mention				
Model	MRR	MR	H@10	H@3	H@1	MRR	MR	H@10	H@3	H@1	MRR	MR	H@10	H@3	H@1
TransE (KG only)	0.336	113	0.523	0.368	0.243	0.314	135	0.508	0.349	0.218	0.333	111	0.519	0.364	0.241
KG+ExtractedLDPs	0.314	126	0.495	0.343	0.224	**0.433**	**43**	**0.659**	0.489	**0.319**	0.293	138	0.468	0.318	0.206
LinkAll	0.344	105	0.531	0.380	0.249	0.430	44	0.653	**0.493**	0.316	0.328	113	0.510	0.360	0.235
Co-occurrence	0.502	47	0.695	0.553	0.402	0.412	48	0.639	0.464	0.297	0.506	47	0.695	0.557	0.408
NeighbBorrow	0.491	49	0.682	0.541	0.392	0.422	46	0.646	0.475	0.308	0.493	50	0.68	0.542	0.395
SuperBorrow	**0.751**	**15**	**0.868**	**0.799**	**0.681**	0.394	49	0.629	0.445	0.277	**0.787**	**11**	**0.888**	**0.835**	**0.723**
DistMult (KG only)	0.302	133	0.489	0.333	0.209	0.257	149	0.436	0.289	0.165	0.302	131	0.489	0.333	0.209
KG+ExtractedLDPs	0.325	113	0.512	0.357	0.232	0.427	35	0.656	0.483	0.311	0.306	125	0.488	0.335	0.216
LinkAll	0.329	108	0.521	0.363	0.233	**0.437**	**33**	**0.670**	**0.496**	**0.315**	0.309	118	0.497	0.339	0.215
Co-occurrence	0.365	74	0.574	0.404	0.261	0.428	**33**	0.664	0.479	0.310	0.351	81	0.558	0.388	0.248
NeighbBorrow	0.415	54	0.639	0.465	0.302	0.412	35	0.645	0.463	0.297	0.408	57	0.631	0.458	0.295
SuperBorrow	**0.482**	**53**	**0.681**	**0.535**	**0.377**	0.415	35	0.655	0.475	0.291	**0.482**	**56**	**0.678**	**0.534**	**0.379**
ComplEx (KG only)	0.312	125	0.493	0.342	0.222	0.275	142	0.459	0.299	0.185	0.312	124	0.493	0.342	0.222
KG+ExtractedLDPs	0.321	107	0.505	0.349	0.229	0.407	36	0.637	0.458	0.291	0.304	117	0.482	0.329	0.216
LinkAll	0.328	107	0.519	0.361	0.232	0.432	34	0.665	0.493	0.311	0.309	118	0.496	0.338	0.216
Co-occurrence	0.358	76	0.570	0.399	0.252	**0.436**	**33**	**0.679**	**0.499**	**0.319**	0.342	83	0.552	0.380	0.238
NeighbBorrow	0.428	47	0.650	0.479	0.315	0.418	35	0.646	0.478	0.298	0.422	50	0.643	0.472	0.309
SuperBorrow	**0.489**	**42**	**0.687**	**0.540**	**0.385**	0.416	38	0.653	0.481	0.291	**0.491**	**43**	**0.686**	**0.541**	**0.388**
RotatE (KG only)	0.358	94	0.560	0.395	0.259	0.331	120	0.527	0.365	0.236	0.354	92	0.557	0.391	0.254
KG+ExtractedLDPs	0.359	94	0.551	0.396	0.264	**0.448**	**44**	0.672	**0.509**	**0.333**	0.341	101	0.528	0.374	0.247
LinkAll	0.363	91	0.558	0.400	0.266	0.442	**44**	0.671	0.503	0.321	0.346	98	0.536	0.378	0.251
Co-occurrence	0.435	54	0.639	0.484	0.329	0.441	46	0.663	0.499	0.327	0.426	56	0.629	0.473	0.321
NeighbBorrow	0.463	43	0.672	0.514	0.357	0.443	45	**0.675**	0.497	0.326	0.457	44	0.664	0.508	0.351
SuperBorrow	**0.682**	**19**	**0.836**	**0.739**	**0.594**	0.412	51	0.652	0.473	0.290	**0.706**	**16**	**0.851**	**0.764**	**0.621**

Table 3: Results of link prediction on FB15K237. Higher is better for all metrics except for the mean rank (MR) for which lower values indicate better performance. The best result for each metric and each KGE method is shown in bold.

test set. Note that even if we consider the complete set of LDPs from the ClueWeb12, the portion of without-mention test entity-pairs in FB15k237 is only 73.23%. This shows the significance of the problem of representing without-mention entity-pairs, which is the focus of this paper.

Training SuperBorrow: We used the with-mention entity-pairs in train split of FB15K237 to train SuperBorrow. The number of entity-pairs in the training set is 311,906, and on average we have 1.32 LDPs per entity-pair. On average, we generate 100 negative triples for each with-mention pair. We hold-out 10% of the training entity-pairs for validation purposes (we obtain 280716 and 31190 entity-pairs for training and validation, respectively). To represent each entity, we use the publicly available 100-dimensional pre-trained RelWalk embeddings[4], which are publicly available for the entities and relations in FB15k237.

According to (1), the input layer of the trained MLP has 400 features. The hyperparameters including the number of hidden layers $\{2, 3\}$, ℓ_2, regularisation coefficient $\{0, 0.01, 0.001\}$, the learning rate $\{0.01, 0.1\}$ and the non-linear activation $\{\tanh, \text{relu}, \text{sigmoid}\}$ are tuned using the above-mentioned validation set. The MLP consists of two 768-dimensional layers, and the last layer represents the entity-pair to be mapped to the LDP embedding space that has 768 dimensions encoded using the SBERT `paraphrase-distilroberta-base`[5] model, which has reported SoTA performance on various knowledge-intensive tasks (Warstadt et al., 2020). SuperBorrow is trained for 50 iterations using mini-batch Stochastic Gradient Descent with momentum and a batch size of 128.[6]

Evaluation Protocol: After augmenting FB15K237 with the borrowed k LDPs for each without-mention entity-pair, we train a KGE method to obtain embeddings for the entities in $\mathcal{E}$, relations in $\mathcal{R}$ and textual relations. The hyperparameter k is tuned on the validation set of FB15K237 for each KGE method from $\{1, 3, 10, 15, 20, 25, 30\}$.

We use Link Prediction, which has been popularly used as an evaluation task to compare the

[4] https://github.com/LivNLP/Relational-Walk-for-Knowledge-Graphs

[5] https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v2

[6] The source is uploaded to openreview and will be made publicly available upon paper acceptance.

KGEs we obtain from a KGE learning method before and after augmenting the KG with the LDPs borrowed using SuperBorrow and other baselines (Wang et al., 2021). Link prediction is the task of predicting the missing head (i.e. $(?, r, t)$) or tail (i.e. $(h, r, ?)$) entity in a given triple by ranking the entities in the KG according to the scoring function of the KGE method. Following prior work, the performance is evaluated using Mean Reciprocal Rank (MRR), Mean Rank (MR) and Hits at Rank k (H@k) under the *filtered* setting, which removes all triples appeared in training, validating or testing sets from candidate triples before obtaining the rank for the ground truth triple. We consider all entities that appear in the corresponding argument of the relation to be predicted to further filter out incorrect candidates, which is known as type-constraint setting (Chang et al., 2014; Toutanova and Chen, 2015).

We also evaluate the learnt KGEs using a relation prediction task, which predicts the relation between two given entities (i.e., $(h, ?, t)$) from the set of relations in the KG. This task assumes that we are given entity-pairs with candidate relations. The performance is measured using the same evaluation metrics as used in the link prediction task under the filtered setting. We use the publicly available OpenKE tool to conduct experiments with different KGE methods (Han et al., 2018).[7]

4.2 Baselines

We compare the proposed LDP borrowing method against multiple baselines as follows.

LinkAll: In this baseline we connect the two entities in each without-mention entity-pair using a distinct relation specific to that entity-pair and augment the KG with the without-mention entity-pairs. This baseline enables us to simply incorporate all without-mention entity-pairs into the KG without requiring to borrow any LDPs. It will demonstrate the importance, if any, of sharing LDPs between with- vs. without-mention entity-pairs as opposed to simply connecting all without-mention entity-pairs with distinct relations.

Co-occurrence: This baseline connects all entity-pairs that co-occurs in any sentence in the corpus ($\mathcal{T}$) with a generic relation (i.e. co-occurrence relation) in the augmented KG and does not distinguish between different textual relations. This baseline is designed to highlight the importance of the context of entity-pair co-occurrences in the corpus beyond simply treating all co-occurrences equally during the augmenting process.

NeighbBorrow: Given a without-mention entity-pair (h^*, t^*), we can borrow the LDPs from the first nearest neighbouring (1NN) with-mention entity-pair (h, t). The similarity between entity-pairs can be computed using (4) in an unsupervised manner using pretrained entity embeddings such as RelWalk embeddings (Bollegala et al., 2021).

$$\text{sim}((h, t), (h^*, t^*)) = \cos(\boldsymbol{h}, \boldsymbol{h^*}) \cos(\boldsymbol{t}, \boldsymbol{t^*}) \quad (4)$$

Here, cos is the cosine similarity between two vectors converted to nonnegative values (i.e. $[0, 1]$) using the linear transformation $(x + 1)/2$. On average, when considering 1NN, we borrow 1.3 LDPs for each without-mention pair of entities. In contrast to the proposed SuperBorrow, NeighbBorrow is unsupervised and decouples entities in each pair when computing their similarity.

5 Results

Link Prediction: Table 3 shows the results of link prediction for different settings on FB15K237 under different KGE methods. Two translational distance-based KGE methods (i.e. **TransE** and **RotatE**) and two semantic matching-based models (i.e. **DistMult** and **ComplEx**) are used as the KGE learning methods (Rossi et al., 2021; Wang et al., 2021). We emphasize that our purpose here is *not* to compare the absolute performance among those KGE methods, but to evaluate the effect of using LDPs for augmenting the KG and representing the without-mention entity-pairs via different borrowing methods. For SuperBorrow, the optimal numbers of borrowed LDPs (k) determined using the validation set for TransE, DistMult, ComplEx and RotatE respectively are 30, 20, 15 and 25.

As shown in Table 3, augmenting the KG with the extracted LDPs (i.e., KG+ExtractedLDPs) significantly improves the performance for with-mention entity-pairs for all KGE methods. However, the performance when predicting links for without-mention entity-pairs decreases slightly for all KGE methods, except for DistMult in the KG+ExtractedLDPs setting. For the borrowing models, even though the co-occurrence baseline improves the prediction for without-mention set, borrowing relevant LDPs from the 1NN entity-pairs (NeighbBorrow) or the proposed supervised borrowing (SuperBorrow) reports superior results.

[7] https://github.com/thunlp/OpenKE

	overall			without-mention		
Model	MR	H@3	H@1	MR	H@3	H@1
DistMult (KG only)	4.1	0.938	0.856	4.0	0.942	0.865
KG+ExtractedLDPs	2.6	0.955	**0.876**	2.7	0.957	**0.883**
LinkAll	7.2	0.887	0.752	7.8	0.880	0.744
Co-occurrence	2.4	0.954	0.875	2.4	0.956	0.882
NeighbBorrow	3.0	0.955	0.874	3.0	0.956	0.881
SuperBorrow	**2.2**	**0.960**	0.875	**2.2**	**0.962**	0.882
ComplEx (KG only)	3.1	0.954	0.900	2.8	0.957	0.908
KG+ExtractedLDPs	1.9	0.967	0.917	1.9	0.967	0.922
LinkAll	4.0	0.909	0.812	4.3	0.902	0.808
Co-occurrence	1.8	0.967	0.916	1.8	0.967	0.920
NeighbBorrow	**1.7**	**0.973**	**0.921**	**1.7**	**0.974**	**0.925**
SuperBorrow	**1.7**	0.972	0.917	**1.7**	0.973	0.922

Table 4: Results of relation prediction on FB15K237.

We can see that the best performance for the *overall* and *without-mention* sets are achieved with the augmented KG using SuperBorrow, followed by NeighbBorrow.

Relation Prediction: Table 4 shows the accuracies for the relation prediction task. Experimentally, the best results for this task is obtained when corrupting r, in addition to h and t corruptions, is applied to generate negative triples to train the KGE method. This negative sampling schedule follows the evaluation procedure of relation prediction. As shown in the table, SuperBorrow reports the best MR and Hits@3 for DistMult KGEs, while NeibhBorrow baseline performs better than SuperBorrow with ComplEx method. Further results for relation prediction are in the Supplementary Appendix A.

Comparisons against Prior Work: We compare our proposed method against prior work, namely Feature Rich Network (**FRN**) (Komninos and Manandhar, 2017) and **Conv (E+DistMult)** (Toutanova et al., 2015). In FRN, an MLP is trained to predict the probability of a given triple being true using different types of features such as the entity types and features extracted from textual relation mentions. Conv(E+DistMult) represents LDPs by vectors using a convolutional neural network, and combines DistMult scoring function with that of the Entity model (E) proposed by Riedel et al. (2013). E model learns a vector for each entity and two vectors for each relation corresponding to the two arguments $\boldsymbol{r}_h$ and $\boldsymbol{r}_t$ of a relation r. The scoring function of a triple in E model is defined as $\boldsymbol{h}^\top \boldsymbol{r}_h + \boldsymbol{t}^\top \boldsymbol{r}_t$. The combined model (E+DistMult) is trained on a linearly weighted combination of KG triples and textual triples. For a fair comparison, we consider the task of predicting missing tail entities and we avoid the type-constraint setting.

As shown in Table 5, for the overall test set of FB15K237 our models outperform both FRN and Conv models according to MRR and H@10. For with-mention entity-pairs, our models report higher scores compared to Conv(E+DistMult), while FRN performs best. For with-mention entity-pairs FRN can extract rich features from the contexts of co-occurrences, which helps it to obtain superior performances. However, both FRN and Conv models perform poorly on without-mention entity-pairs, where such contextual features are unavailable. On the other hand, by using the proposed SuperBorrow to augment LDPs for KGs we can overcome this limitation successfully.

	overall		with-mention		without-mention	
Model	MRR	H@10	MRR	H@10	MRR	H@10
Conv (E+DistMult)	0.401	0.581	0.339	0.499	0.424	0.611
FRN	0.403	0.620	**0.441**	**0.683**	0.387	0.595
ours (DistMult)	0.460	**0.714**	0.378	0.649	0.468	**0.720**
ours (RotatE)	**0.499**	0.712	0.439	0.674	**0.504**	0.715

Table 5: Comparisons against prior work on link prediction on FB15K237. The results for prior work are taken from the original papers. The best results are in bold, while the second best results are underlined.

6 Analysis

Borrowed LDPs: To provide examples of LDPs injected into FB15K237, Table 6 shows the borrowed LDPs by NeighbBorrow and SuperBorrow for some selected entity-pairs. We can see that representative LDPs of various relation types are ranked at the top by SuperBorrow. For example, for the *film-distributor* relation, NeighbBorrow selects LDPs containing specific tokens such as *movie* or *film*, whereas SuperBorrow retrieves LDPs that better express the target relation such as *20th Century Fox*:⟨-dobj⟩:released:⟨nsubj⟩:*Lincoln*.

Relation Categories: To better analyse the effect of the proposed SuperBorrow for KGEs, we evaluate the link prediction task on different relation categories including 1to1, 1toN, Nto1 and NtoN as defined in Bordes et al. (2013).

Table 7 presents the results of predicting head entities for all KGE methods considering KG only and SuperBorrow. We can see that SuperBorrow achieves higher performance over the original graph on all relation categories. In particular, our proposal significantly boosts the performance of predicting head entities for the Nto1 relation type where all KGE methods report the lowest H@10 for

Entity-pairs (h, r, t)	Borrowed LDPs NeighbBorrow	 SuperBorrow
h= Woodrow Wilson t= League of Nations r= organizations-founded	h:⟨-nsubj⟩:joined:⟨dobj⟩:t h:⟨-nsubj⟩:left:⟨dobj⟩:t h:⟨-poss⟩:t	h:⟨-poss⟩:t h:⟨-nsubj⟩:president:⟨prep⟩:of:⟨pobj⟩:t h:⟨-nsubj⟩:joined:⟨dobj⟩:t h:⟨-poss⟩:ambassador:⟨prep⟩:to:⟨pobj⟩:t h:⟨-nsubj⟩:member:⟨prep⟩:of:⟨pobj⟩:t
h= 20th Century Fox t= Lincoln r= film-distributor	h:⟨-nn⟩:movie:⟨appos⟩:t h:⟨-nn⟩:film:⟨nsubj⟩:t h:⟨-nn⟩:movie:⟨dep⟩:t h:⟨-poss⟩:t	h:⟨-dobj⟩:released:⟨nsubj⟩:t h:⟨-dobj⟩:release:⟨nsubj⟩:t h:⟨-nsubj⟩:released:⟨dobj⟩:t h:⟨-appos⟩:grant:⟨appos⟩:t
h= Deep Impact t= Leslie Dilley r= film-production-design-by	h:⟨-pobj⟩:in:⟨-prep⟩:work:⟨poss⟩:t h:⟨-nn⟩:fame:⟨-pobj⟩:of:⟨-prep⟩:t h:⟨poss⟩:t	h:⟨-dep⟩:film:⟨poss⟩:t h:⟨vmod⟩:produced:⟨prep⟩:by:⟨pobj⟩:t h:⟨vmod⟩:written:⟨prep⟩:by:⟨pobj⟩:t h:⟨-dep⟩:tagged:⟨appos⟩:t h:⟨-nn⟩:film:⟨nsubj⟩:t
h= Idaho t= Christianity r= religion	h:⟨-nsubjpass⟩:located:⟨prep⟩:in:⟨pobj⟩:t h:⟨-appos⟩:usa:⟨-appos⟩:t h:⟨-poss⟩:t	h:⟨-poss⟩:t h:⟨-amod⟩:state:⟨prep⟩:of:⟨pobj⟩:t h:⟨rcmod⟩:plays:⟨dobj⟩:t h:⟨-dobj⟩:entered:⟨nsubj⟩:t h:⟨-nn⟩:date:⟨nn⟩:t

Table 6: Borrowed LDPs of selected entity-pairs. Top 5 LDPs with our borrowing method and LDPs borrowed from 3NN entity-pairs are shown.

Method		1to1	1toN	Nto1	NtoN
	# Tuples	192	1293	4289	14,696
TransE	KG only	0.536	0.597	0.124	0.418
	SuperBorrow	0.947	0.984	0.377	0.829
DistMult	KG only	0.500	0.433	0.064	0.371
	SuperBorrow	0.922	0.856	0.338	0.547
ComplEx	KG only	0.495	0.434	0.045	0.368
	SuperBorrow	0.917	0.913	0.277	0.601
RotatE	KG only	0.568	0.631	0.118	0.388
	SuperBorrow	0.932	0.969	0.404	0.722

Table 7: Hits@10 of tail prediction for different relation categories.

(a) KG only (b) KG with LDPs

Figure 1: t-SNE plots for DistMult entity embeddings comparing (a) KG-only and (b) KG with LDPs.

the KG only setting. Similar results are obtained for predicting the tail entities as in Appendix B. Overall, these results show that incorporating information from text corpora into KGs enables us to learn KGEs that encode diverse relation types.

Visualisation of Entity Embeddings: In Figure 1, we visualise the entity embeddings of KGonly and KG with LDPs using t-distributed stochastic neighbour embeddings (t-SNE) (Hinton and Roweis, 2002) method. Relations in FB15k237 are labelled as domain/type/property where domain/type represents the type of a head entity in the relation. Thus, for each entity in the KG, we extract its types from all training triples where the entity acts as the head. We label entities that belong to the two most frequent entity types, which are people/person (4,538 entities) and film/film (1,923 entities). From Figure 1, we see that the embeddings learnt from the augmented graph results in distinct clusters of the same type, compared to the clusters obtained from the KG alone. This emphasizes the importance of using textual mentions in KGE learning.

7 Conclusion

We considered the problem of representing without-mention entity-pairs in KGE learning. Specifically, we proposed a method (SuperBorrow) to determine which LDPs to borrow from with-mention entity-pairs to augment a KG using a corpus. Our proposed method improves the performance of several KGE learning methods in link and relation prediction tasks.

References

Bo An, Bo Chen, Xianpei Han, and Le Sun. 2018. Accurate text-enhanced knowledge graph representation learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pages 745–755.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, Springer, pages 722–735.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. pages 1247–1250.

Danushka Bollegala, Huda Hakami, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2021. RelWalk - a latent variable model approach to knowledge graph embedding. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, pages 1551–1565. https://doi.org/10.18653/v1/2021.eacl-main.133.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *WWW 2010*. pages 151 – 160.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*. pages 1–9.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1568–1579.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 740–750.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. *arXiv preprint arXiv:1704.08384* .

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12(7).

Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* 48(1-4):2.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora.

Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.

Geoffrey Hinton and Sam T Roweis. 2002. Stochastic neighbor embedding. In *NIPS*. Citeseer, volume 15, pages 833–840.

Alexandros Komninos and Suresh Manandhar. 2017. Feature-rich networks for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 324–329.

M Kroetsch and G Weikum. 2016. Special issue on knowledge graphs. *Journal of Web Semantics* 37(38):53–54.

Teng Long, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2016. Leveraging lexical resources for learning entity embeddings in multi-relational data. *arXiv preprint arXiv:1605.05416* .

Fengyuan Lu, Peijin Cong, and Xinli Huang. 2020. Utilizing textual information in knowledge graph embedding: A survey of methods and applications. *IEEE Access* 8:92072–92088.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 777–782.

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2015. Efficient learning of entity and predicate embeddings for link prediction in knowledge graphs. In *URSW@ ISWC*. pages 26–37.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8(3):489–508.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* .

Saed Rezayi, Handong Zhao, Sungchul Kim, Ryan Rossi, Nedim Lipka, and Sheng Li. 2021. Edge: Enriching knowledge graph embeddings with external text. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, pages 2767–2776. https://doi.org/10.18653/v1/2021.naacl-main.221.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 74–84.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15(2):1–49.

Paolo Rosso, Dingqi Yang, and Philippe Cudre-Mauroux. 2019. Revisiting text and knowledge graph joint embeddings: The amount of shared information matters! In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, pages 2465–2473.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27(2):443–460.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*. pages 926–934.

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*. pages 697–706.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* .

Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019. Interpretable question answering on knowledge bases and text. *arXiv preprint arXiv:1906.10924* .

Xing Tang, Ling Chen, Jun Cui, and Baogang Wei. 2019. Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. *Information Processing & Management* 56(3):809–822.

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*. pages 57–66.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*. PMLR, pages 2071–2080.

Neil Veira, Brian Keng, Kanchana Padmanabhan, and Andreas G Veneris. 2019. Unsupervised embedding enhancements of knowledge graphs using textual associations. In *IJCAI*. pages 5218–5225.

Meihong Wang, Linling Qiu, and Xiaoli Wang. 2021. A survey on knowledge graph embeddings for link prediction. *Symmetry* 13(3):485.

Quan Wang, Zhendong 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):2724–2743.

Shirui Wang, Chao Jiang, et al. 2019a. Knowledge graph embedding with interactive guidance from entity descriptions. *IEEE Access* 7:156686–156693.

Yashen Wang, Huanhuan Zhang, and Haiyong Xie. 2019b. A model of text-enhanced knowledge graph representation learning with collaborative attention. In *Asian Conference on Machine Learning*. PMLR, pages 220–235.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1591–1601.

Zhigang Wang, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of International Joint Conference on Artificial Intelligent (IJCAI)*. pages 4–17.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R Bowman. 2020. Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually). *arXiv preprint arXiv:2010.05358* .

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973* .

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 30.

Hongcai Xu, Junpeng Bao, and Junqing Wang. 2020. Knowledge-graph based proactive dialogue generation with improved meta-learning. In *2020 2nd International Conference on Image Processing and Machine Vision*. pages 40–46.

Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Knowledge graph representation with jointly structural and textual encoding. *arXiv preprint arXiv:1611.08661* .

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* .

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 267–272.

Supplementary Material

A Relation Prediction

Relation prediction results for all the KGE methods are shown in Table 8. As we see, unlike semantic matching-based KGE models, incorporating LDPs into the KG do not improve relation prediction for translational distance-based KGE methods (TransE and RotatE). For KG+ExtractedLDPs embeddings, the performance for with-mention set decreases by 0.045 and 0.012 on average for MRR and H@{10,3,1}, for TransE and RotatE respectively. In-depth analysis for this observation can be conducted in future research.

B Tail Prediction for Relation Categories

Table 9 presents Hits@10 for tail prediction considering 1to1, 1toN, Nto1 and NtoN relation categories. As we see, SuperBorrow embeddings obtain the best results for all KGE methods and all the relation categories.

Method		1to1	1toN	Nto1	NtoN
	# Tuples	192	1293	4289	14,696
TransE	KG only	0.547	0.097	0.851	0.574
	SuperBorrow	0.943	0.647	0.980	0.907
DistMult	KG only	0.521	0.055	0.774	0.507
	SuperBorrow	0.880	0.424	0.898	0.657
ComplEx	KG only	0.500	0.034	0.787	0.518
	SuperBorrow	0.869	0.456	0.964	0.753
RotatE	KG only	0.536	0.107	0.855	0.561
	SuperBorrow	0.927	0.731	0.983	0.853

Table 9: Hits@10 of tail prediction for different relation categories.

C Training KGE Methods

For reproducability, we list the hyperparameter setting to train KGE methods in Table 10. AdaGrad (Duchi et al., 2011) with 100 batches is used to learn KGEs. Table 11 shows the training time (in hours) to train KGE methods for KG only and SuperBorrow using OpenKE-Pytorch tool (Han et al., 2018).

Method		#Train tuples	Time (h)
TransE	KG only	272,115	0.42
	SuperBorrow	1,217,294	1.58
DistMult	KG only	272,115	0.78
	SuperBorrow	1,036,904	2.67
ComplEx	KG only	272,115	0.69
	SuperBorrow	946,709	2.11
RotatE	KG only	272,115	1.11
	SuperBorrow	1,127,099	4.13

Table 11: Training time on FB15K237 in hours.

	overall					with-mention					without-mention				
Model	MRR	MR	H@10	H@3	H@1	MRR	MR	H@10	H@3	H@1	MRR	MR	H@10	H@3	H@1
TransE (KG only)	0.961	1.6	0.992	0.980	0.940	0.919	1.9	0.988	0.958	0.875	0.967	1.5	0.993	0.983	0.949
KG+ExtractedLDPs	0.934	1.6	0.994	0.967	0.899	0.860	1.7	0.991	0.926	0.789	0.944	1.5	0.994	0.973	0.914
LinkAll	0.932	1.5	0.993	0.955	0.899	0.845	1.9	0.985	0.887	0.778	0.944	1.4	0.994	0.964	0.916
Co-occurrence	0.925	1.6	0.993	0.962	0.887	0.863	1.8	0.990	0.931	0.793	0.933	1.5	0.993	0.967	0.899
NeighbBorrow	0.927	1.5	0.994	0.964	0.888	0.862	1.7	0.993	0.929	0.791	0.936	1.5	0.994	0.969	0.901
SuperBorrow	0.925	1.5	0.993	0.963	0.886	0.868	1.8	0.990	0.926	0.802	0.933	1.5	0.994	0.968	0.897
DistMult (KG only)	0.901	4.1	0.968	0.938	0.856	0.855	4.5	0.959	0.914	0.789	0.907	4.0	0.969	0.942	0.865
KG+ExtractedLDPs	0.918	2.6	0.980	0.955	0.876	0.887	2.4	0.982	0.940	0.826	0.922	2.7	0.980	0.957	0.883
LinkAll	0.825	7.2	0.940	0.887	0.752	0.883	2.1	0.986	0.944	0.813	0.818	7.8	0.934	0.880	0.744
Co-occurrence	0.918	2.4	0.979	0.954	0.875	0.89	2.0	0.980	0.942	0.831	0.921	2.4	0.979	0.956	0.882
NeighbBorrow	0.917	3.0	0.979	0.955	0.874	0.883	2.7	0.976	0.942	0.819	0.921	3.0	0.979	0.956	0.881
SuperBorrow	0.920	2.2	0.984	0.960	0.875	0.885	2.2	0.980	0.943	0.822	0.924	2.2	0.985	0.962	0.882
ComplEx (KG only)	0.929	3.1	0.977	0.954	0.900	0.884	4.8	0.962	0.925	0.835	0.935	2.8	0.980	0.957	0.908
KG+ExtractedLDPs	0.944	1.9	0.987	0.967	0.917	0.921	1.7	0.986	0.962	0.877	0.947	1.9	0.987	0.967	0.922
LinkAll	0.867	4.0	0.955	0.909	0.812	0.906	1.8	0.988	0.960	0.848	0.861	4.3	0.951	0.902	0.808
Co-occurrence	0.944	1.8	0.987	0.967	0.916	0.930	1.7	0.989	0.965	0.892	0.946	1.8	0.987	0.967	0.920
NeighbBorrow	0.948	1.7	0.989	0.973	0.921	0.925	1.9	0.987	0.965	0.884	0.951	1.7	0.989	0.974	0.925
SuperBorrow	0.946	1.7	0.990	0.972	0.917	0.922	1.8	0.987	0.962	0.879	0.949	1.7	0.990	0.973	0.922
RotatE (KG only)	0.972	1.4	0.996	0.990	0.954	0.945	1.3	0.993	0.981	0.910	0.976	1.4	0.997	0.991	0.960
KG+ExtractedLDPs	0.967	1.3	0.995	0.988	0.945	0.933	1.5	0.983	0.974	0.892	0.971	1.2	0.996	0.990	0.952
LinkAll	0.958	1.3	0.995	0.984	0.931	0.923	1.6	0.983	0.964	0.879	0.963	1.3	0.996	0.987	0.938
Co-occurrence	0.964	1.3	0.994	0.985	0.943	0.931	1.4	0.987	0.970	0.892	0.969	1.3	0.995	0.987	0.949
NeighbBorrow	0.964	1.3	0.995	0.985	0.941	0.933	1.5	0.985	0.971	0.894	0.968	1.2	0.996	0.987	0.948
SuperBorrow	0.964	1.2	0.995	0.986	0.941	0.931	1.5	0.985	0.972	0.892	0.968	1.2	0.996	0.988	0.947

Table 8: Relation predictino on FB15K237.

KGE Method	learning rate	embedding dimension	negative samples	loss function	margin	epochs
TransE	1.0	300	25	Margin loss	5.0	1000
DistMult	0.5	300	25	SoftPlus loss	-	1000
ComplEx	0.5	100	25	SoftPlus loss	-	1000
RotatE	2e-5	300	25	SigmoidLoss	9.0	1000

Table 10: The hyperparameter setting for KGE methods on link prediction task.