# Duality of Link Prediction and Entailment Graph Induction

**Mohammad Javad Hosseini**$^{\star\S}$ **Shay B. Cohen**$^{\star}$ **Mark Johnson**$^{\ddagger}$ and **Mark Steedman**$^{\star}$
$^{\star}$University of Edinburgh $^{\S}$The Alan Turing Institute, UK $^{\ddagger}$Macquarie University
javad.hosseini@ed.ac.uk, scohen@inf.ed.ac.uk
mark.johnson@mq.edu.au, steedman@inf.ed.ac.uk

## Abstract

Link prediction and entailment graph induction are often treated as different problems. In this paper, we show that these two problems are actually complementary. We train a link prediction model on a knowledge graph of assertions extracted from raw text. We propose an entailment score that exploits the new facts discovered by the link prediction model, and then form entailment graphs between relations. We further use the learned entailments to predict improved link prediction scores. Our results show that the two tasks can benefit from each other. The new entailment score outperforms prior state-of-the-art results on a standard entialment dataset and the new link prediction scores show improvements over the raw link prediction scores.

## 1 Introduction

Link prediction and entailment graph induction are often treated as different problems. The former (Figure 1A) is used to infer missing relations between entities in existing knowledge graphs (Socher et al., 2013; Bordes et al., 2013; Riedel et al., 2013). The latter (Figure 1B) constructs entailment graphs with relations as nodes and entailment rules as edges between them (Berant et al., 2011, 2015; Hosseini et al., 2018) for the task of answering questions from text. In this paper, we show that these two problems are complementary by demonstrating how link prediction can help identify entailments and how discovered entailments can help predict missing links.

Methods to learn entailment graphs (Berant et al., 2011, 2015; Hosseini et al., 2018) process large text corpora to find *local* entailment scores between relations based on the Distributional Inclusion Hypothesis which states that a word (relation) $r$ entails another word (relation) $q$ if and only if in any context that $r$ can be used, $q$ can be used in its place (Dagan et al., 1999; Geffet and Dagan, 2005; Kartsaklis and Sadrzadeh, 2016). They use types such as *person*, *location* and *time*, to disambiguate polysemous relations (e.g., *person born in location* and *person born in time*). Entailment graphs are then formed by imposing *global* constraints such as transitivity of the entailments (Berant et al., 2011). The paraphrase[1] and entailment relations provide an interpretable resource that can be used to answer questions, when the answer is not explicitly stated in the text. For example, while we can find on the web the assertion *Loch Fyne lies at the foot of mountains*, we cannot find a sentence directly stating that *Loch Fyne is located near mountains* by querying Google as of 4th March 2019. Knowledge of the entailment relation between *lies at the foot of* and *is located*

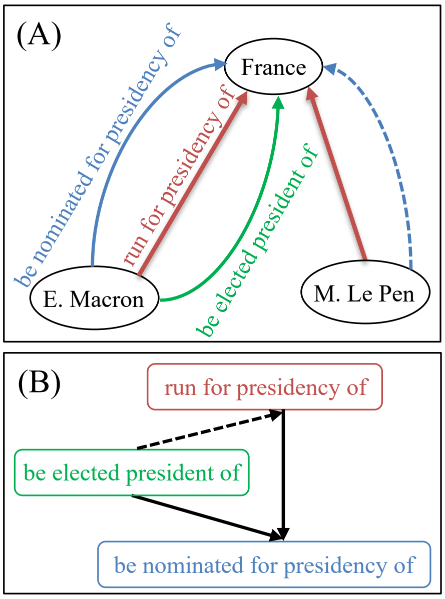

Figure 1: A link prediction knowledge graph (A) and an entailment graph (B) for entities of types *politician,country*. The solid lines are discovered correctly, but the dashed ones are missing. However, evidence from the link prediction model can be used to add the missing entailment rule in the entailment graph (B). Similarly, the entailment graph can be used to add the missing link in the knowledge graph (A).

[1] Relations that entail each other in both directions are regarded as paraphrases.

*near* can be used to answer such questions.

On the other hand, link prediction (or knowledge base completion) models are based on distributional methods and directly predict the source data. These models have received much attention in the recent years (Socher et al., 2013; Bordes et al., 2013; Riedel et al., 2013; Toutanova et al., 2016; Trouillon et al., 2016; Dettmers et al., 2018). The current methods learn embeddings for all entities and relations and a function to score any potential relation between the entities. One of the main capabilities of these models is that they implicitly exploit entailment relations such as *person born in country* entails *person be from country* (Riedel et al., 2013). However, entailment relations are not learned explicitly. For example, we cannot simply compute the cosine similarity of the vector representations of the two relations to detect the entailment between them, because cosine similarity is symmetric (§5.1). These methods are usually applied to augment existing knowledge graphs such as Freebase (Bollacker et al., 2008), DBPedia (Auer et al., 2007) and Yago (Suchanek et al., 2007), but they can also be applied to assertions extracted from raw text.

In this paper, we explore the synergies between the two tasks. Current entailment graphs suffer from sparsity and noise in the data. The link prediction methods discover new facts that can be used to alleviate the sparsity issue. In addition, they can remove noise by filtering facts that are not consistent with the other facts. We propose a new entailment score based on link prediction (§3.1) which significantly improves over prior state-of-the-art results on a standard entailment detection dataset (5.1). For example, our method can discover that *be elected president of* entails *run for presidency of* by relying on the predicted links concerning the two relations (Figure 1).

In addition, we show that the discovered entailments can be used to predict links in knowledge graphs (§3.2). For example, knowing that *run for presidency of* entails *be nominated for presidency of* as well as the assertion *Le Pen ran for presidency of France*, we can infer that she also *was nominated for presidency of France*. In our experiments, we show improvements over a state-of-the-art link prediction model (§4.2).[2]

## 2 Background and Notation

Let $T$ denote the set of all types (e.g., *politician*), $\mathcal{E}(t)$ the set of entities with type $t$ (e.g., *E. Macron*) and $\mathcal{R}(t_1, t_2)$ the set of relations with types $(t_1, t_2)$ or $(t_2, t_1)$ (e.g., *be elected president of*). We denote by $\mathcal{E} = \bigcup_t \mathcal{E}(t)$ the set of all entities and by $\mathcal{R} = \bigcup_{t_1,t_2} \mathcal{R}(t_1, t_2)$ the set of all relations. Denote by $\mathcal{H}(t_1, t_2)$ the knowledge graph consisting of a set of correct triples $(r, e_1, e_2)$, where $r \in \mathcal{R}(t_1, t_2)$, $(e_1, e_2) \in \mathcal{E}^2(t_1, t_2)$ and $\mathcal{E}^2(t_1, t_2) = \big(\mathcal{E}(t_1) \times \mathcal{E}(t_2)\big) \cup \big(\mathcal{E}(t_2) \times \mathcal{E}(t_1)\big)$. We define $\mathcal{E}^2 = \bigcup_{t_1,t_2} \mathcal{E}^2(t_1, t_2)$ the set of all possible entity pairs. We denote by $\mathcal{H} = \bigcup_{t_1,t_2} \mathcal{H}(t_1, t_2)$ the knowledge graph consisting of all types. In practice, we have not observed all the correct triples, but instead have access to a noisy and incomplete knowledge graph. We define by $X_{r,e_1,e_2}$ a binary random variable which is 1 if $(r, e_1, e_2)$ is in the knowledge graph and 0, otherwise.

In the rest of this section, we introduce the problem of link prediction (§2.1) and finding entailment relations (§2.2).

### 2.1 Link Prediction

For each triple $(r, e_1, e_2)$, a link prediction model defines a scoring function $f(r, e_1, e_2)$ of its plausibility (Socher et al., 2013; Bordes et al., 2013; Riedel et al., 2013; Toutanova et al., 2016; Trouillon et al., 2016; Dettmers et al., 2018). We use ConvE (Dettmers et al., 2018), a state-of-the-art and efficient model, in our experiments. The models then choose $f$ such that the score $f(r, e_1, e_2)$ of a plausible triple $(r, e_1, e_2) \in \mathcal{H}$ is higher than the score $f(r', e'_1, e'_2)$ of an implausible triple $(r', e'_1, e'_2) \notin \mathcal{H}$ (Nguyen, 2017). The plausibility score $f(r, e_1, e_2)$ can optionally be mapped into a probability score $S_{r,e_1,e_2}$.[3] The probability score $S_{r,e_1,e_2}$ is an estimate of $\mathrm{P}(X_{r,e_1,e_2}{=}1)$, i.e., the probability of the triple being correct. We denote by $S \in [0, 1]^{|\mathcal{R}|\times|\mathcal{E}^2|}$ the matrix containing triple probability scores. We define $S(t_1, t_2) \in [0, 1]^{|\mathcal{R}(t_1,t_2)|\times|\mathcal{E}^2(t_1,t_2)|}$ the submatrix of $S$ with $\mathcal{R}(t_1, t_2)$ as rows and $\mathcal{E}^2(t_1, t_2)$ as columns. We apply a link prediction model to a knowledge graph of predicate-argument structures extracted from text (§4.2).

### 2.2 Entailment Prediction

The goal is to find entailment scores between all relations with the same types, where the

[2] Our code and data are available at https://github.com/mjhosseini/linkpred_entgraph.

[3] For example by applying the Sigmoid function.

entities can be in the same or opposite order (Berant et al., 2011; Lewis and Steedman, 2014b; Hosseini et al., 2018). We denote by $W(t_1,t_2) \in [0,1]^{|\mathcal{R}(t_1,t_2)|\times|\mathcal{R}(t_1,t_2)|}$ the (sparse) matrix containing all similarity scores $W_{r,q}$ between relations $r,q \in \mathcal{R}(t_1,t_2)$. We define $W$ the (block diagonal) matrix consisting of all the similarity matrices $W(t_1,t_2)$. For a $\delta > 0$, we define typed entailment graphs as $G_\delta(t_1,t_2) = \big(\mathcal{R}(t_1,t_2), E_\delta(t_1,t_2)\big)$, where $\mathcal{R}(t_1,t_2)$ are the nodes and $E(t_1,t_2) = \{(r,q)|r,q \in \mathcal{R}(t_1,t_2), W_{r,q} \geq \delta\}$ are the edges of the entailment graphs.

Existing entailment similarity measures for relation entailment such as Weeds (Weeds and Weir, 2003), Lin (Lin, 1998), and Balanced Inclusion (BInc; Szpektor and Dagan, 2008) are typically defined on feature vectors consisting of entity-pairs (e.g., *Obama-Hawaii*), where the values are frequencies or pointwise mutual information (PMI) between the relations and the features (Berant et al., 2011, 2012, 2015). While these methods currently hold state-of-the-art results on relation entailment datasets (Hosseini et al., 2018), they suffer from low recall because the feature vectors are usually sparse and do not have high overlap with each other. The link prediction models, on the other hand, can predict the probability of any triple being in the knowledge graph. Using predicted probability scores can hugely alleviate the sparsity problem by increasing the overlap between feature vectors (§3.1).

## 3 Duality between Entailment Scores and Link Prediction

We discuss the relationship between link prediction scores $S(t_1,t_2)$ and entailment scores $W(t_1,t_2)$. We claim that while these two tasks are usually treated separately, they are complementary. We propose a method to predict entailment scores by using link prediction scores. The proposed score estimates the probability of relations given one another. It exploits the strength of the link prediction models, i.e., predicting new facts as well as removing noise from the existing ones (§3.1). We further show how we can improve link prediction scores by using predicted entailment scores. Having access to an entailment relation $r \rightarrow q$, we use the link prediction scores of $r$ to refine the scores of $q$ for any entity pairs (§3.2). All the methods in this section are applied for each type pair separately; however, in the rest of the paper, we drop $(t_1,t_2)$ for simplicity of the notation.

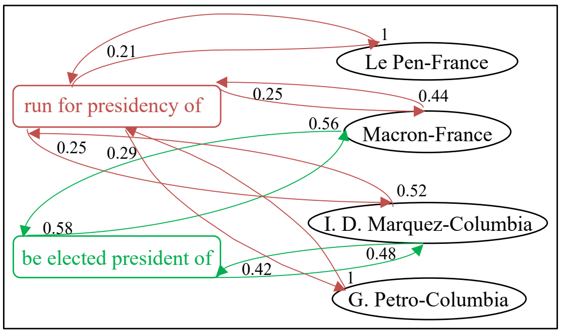

Figure 2: A small Markov chain with two relations (squares) and four entity-pairs (ovals). Directed edges connect each relation to its related entity-pairs, and vice versa. Transition probabilities are shown on each edge. The outgoing probabilities from each node sum to 1.

### 3.1 Entailment Scores From Link Prediction

In this section, we show how we can use link prediction scores to predict entailment scores. In order to compute the entailment scores, we apply a link prediction method on the knowledge graph $\mathcal{H}$. We define a new entailment score based on link prediction scores.

More specifically, We reform the knowledge graph representation into a Markov chain over a bipartite graph $M = (V_M, E_M)$, where $V_M = \mathcal{R} \cup \mathcal{E}^2$ are the nodes of the graph, and $E_M$ contains edges $(\langle r\rangle, \langle e_1,e_2\rangle)$ and $(\langle e_1,e_2\rangle, \langle r\rangle)$ iff $\mathrm{P}(X_{r,e_1,e_2}{=}1) \geq 0$. Figure 2 shows an example Markov chain with only two relations and four entity-pairs. The transition probabilities of the chain are:

$$\begin{aligned}
\mathrm{P}(\langle e_1,e_2\rangle \mid \langle r\rangle) &= \frac{\mathrm{P}(X_{r,e_1,e_2}{=}1)}{\sum_{e_1,e_2\in\mathcal{E}^2}\mathrm{P}(X_{r,e_1,e_2}{=}1)} \\
\mathrm{P}(\langle r\rangle \mid \langle e_1,e_2\rangle) &= \frac{\mathrm{P}(X_{r,e_1,e_2}{=}1)}{\sum_{r\in\mathcal{R}}\mathrm{P}(X_{r,e_1,e_2}{=}1)}
\end{aligned}$$

For relations $r$ and $q$, we define the entailment score $W_{r,q} = \mathrm{P}(\langle q\rangle|\langle r\rangle)$, where we compute the probability by considering only the paths of length 2 between $r$ and $q$ that pass through one entity-pair node.[4] We define:

[4] Longer paths did not yield better performance in our experiments while increasing the memory and running time requirements.

$$\mathrm{P}(\langle q\rangle|\langle r\rangle) = \sum_{e_1,e_2\in\mathcal{E}^2} \mathrm{P}(\langle q\rangle|\langle e_1,e_2\rangle)\,\mathrm{P}(\langle e_1,e_2\rangle|\langle r\rangle). \tag{1}$$

We use $S_{r,e_1,e_2}$ from the link prediction model as an estimate of $\mathrm{P}(X_{r,e_1,e_2}{=}1)$ to compute Equation 1. We can compute the scores for all $r, q \in \mathcal{R}$ efficiently, by normalizing each row of the matrices $S$ and $S^\top$ and multiplying them.[5] Note that building the matrix $S$ for all possible triples make the computation of the scores intractable, especially for large number of relations (§4.1). In our experiments, we consider any $(r, e_1, e_2)$ seen in the corpus. In addition, we add a subset of high scoring triples not seen in the corpus (§4.2).

### 3.2 Improving Link Prediction Scores using Entailment Scores

In the previous section, we demonstrated how we can use link prediction methods to learn entailment scores. In this section, we consider the inverse problem, i.e., we use the predicted entailment relations to improve link prediction scores. We assume the Distributional Inclusion Hypothesis (DIH) which states that a word (relation) $r$ entails another word (relation) $q$ if and only if in any context that $r$ can be used, $v$ can be used in its place (Dagan et al., 1999; Geffet and Dagan, 2005; Kartsaklis and Sadrzadeh, 2016). In particular, in a correct and complete knowledge graph, we have:

$$\begin{aligned} r{\rightarrow}q \implies & \forall(e_1,e_2)\in\mathcal{E}^2 : \\ & X_{r,e_1,e_2}=1 \rightarrow X_{q,e_1,e_2}=1 \\ & \implies X_{r,e_1,e_2} \leq X_{q,e_1,e_2}. \end{aligned} \tag{2}$$

Therefore when $r \rightarrow q$, it is reasonable to assume $\mathrm{P}(X_{r,e_1,e_2} = 1) \leq \mathrm{P}(X_{q,e_1,e_2} = 1)$ for all entity pairs $e_1, e_2$. This would suggest we can define a new link prediction score based on entailment relations:

$$S^{ent}_{q,e_1,e_2} = \max_{r\in\mathcal{R}:\, r\rightarrow q} S_{r,e_1,e_2}. \tag{3}$$

However, since we do not have access to the entailment relations and can only rely on the predictions, Equation 3 is likely to be very noisy. We smooth Equation 3 by using a weighted average of the scores of each entailment relation. We define:

$$S^{ent}_{q,e_1,e_2} = \max\Big(S_{q,e_1,e_2}, \sum_{r\in\mathcal{R}} W'_{r,q} S_{r,e_1,e_2}\Big),$$

where $W'_{r,q}$ is defined by normalizing the $q$th column of the matrix $W$.

$$W'_{r,q} = \frac{W_{r,q}}{\sum_{r':r'\rightarrow q} W_{r',q}}.$$

## 4 Experimental Set-up

In this section, we discuss the details of our experiments. We first describe the text corpus and extracted triples which are used as the input to our method (§4.1). We then describe the details of the link prediction model (§4.2), the datasets used to test the models (§4.3) and the baseline systems (§4.4).

### 4.1 Text Corpus

Link prediction models are often applied to existing knowledge graphs such as Freebase (Bollacker et al., 2008), DBPedia (Auer et al., 2007) and Yago (Suchanek et al., 2007); however, we chose to experiment on assertions extracted from raw text. This is because we can then evaluate the predicted entailments on existing entailment datasets with examples stated in natural language (§4.3).

We use the multiple-source NewsSpike corpus of Zhang and Weld (2013). The NewsSpike corpus includes 550K news articles and is well-suited for finding entailment and paraphrasing relations as it includes different articles from different sources describing identical news stories. We use the triples released by Hosseini et al. (2018)[6] who run the semantic parser of Reddy et al. (2014), GraphParser, to extract binary relations between a predicate and its arguments. GraphParser uses Combinatorial Categorial Grammer (CCG) syntactic derivations by running EasyCCG (Lewis and Steedman, 2014a). The parser converts sentences to neo-Davisonian semantics, a first order logic that uses event identifiers and extracts one binary relation for each event and pair of arguments (Parsons, 1990). The entities are typed by first linking to Freebase (Bollacker et al., 2008) and then selecting the most notable type of the entity from Freebase and mapping it to FIGER types (Ling and Weld, 2012) such as *building*, *disease* and *person*. They use the first level of the FIGER types

[5] An alternative approach would be based on sampling paths over the Markov Chain, but we compute the exact solution by performing matrix multiplication.

[6] Accessed from https://github.com/mjhosseini/entGraph.

hierarchy to assign one of the 49 types (out of 113 total types) to the entities (Hosseini et al., 2018).

Hosseini et al. (2018) extract 29M unique binary relations. We follow them by filtering any relation that is seen with less than three unique entity-pairs, and any entity-pairs that is seen with less than three unique relations. The filtered corpus has 3.9M relations covering 304K typed relations (101K untyped relations).

## 4.2 Link Prediction

We randomly split the corpus into training (95%), validation (4%) and test (1%) sets. We train the link prediction model on the training set and use the validation set for parameter tuning. We apply ConvE (Dettmers et al., 2018)[7], a state-of-the-art model for link prediction, on the training set. ConvE is an efficient multi-layer convolutional network model. Unlike most other link prediction models that take as input an entity pair and a relation as a triple $(r, e_1, e_2)$ and score it (1-1 scoring), ConvE takes one $(r, e_1)$ pair and scores it against all entities $e_2$ (1-N scoring). This improves the training time of ConvE, however more importantly, it is very fast at inference time as well. This is particularly important for our method as we apply the link prediction model exhaustively to predict new high-quality facts (§4.4).

We learn 200-dimensional vectors for each entity and relation. We use the default parameter settings of the ConvE model as those parameters yielded good results on the validation set.[8] We run the model for 80 epochs where the model has converged (less than $10^{-5}$ change in training loss). We learn embeddings for each predicate and its reverse to handle examples where the argument order of the two predicates are different.

For evaluating on the entailment task, we calculate entailment scores by using the predictions of the link prediction model on the triples in train, development and test sets. This is because the other baselines have also access to the whole set of triples (§4.4). However, for evaluating the link prediction model, we compute entailment scores by only considering the predictions in the training set. This is essential as the entailment scores will be used to predict improved link prediction scores on the test set. Therefore, the comparison will not be valid if the method has access to the test triples while computing entailment scores.

## 4.3 Evaluation Datasets

We discuss the datasets that we use to test the proposed methods for the entailment detection and the link prediction tasks.

**Entailment Detection Evaluation.** For the entailment detection task, we evaluate on Levy/Holt's dataset (Levy and Dagan, 2016; Holt, 2018). Each example in the dataset contains a pair of triples where the entities are the same (possibly in the reverse order), but the relations are different. The label of the examples are either positive or negative, meaning that the first triple entails or does not entail the second triple. For example *Bartlett was interviewed on television*, entails *Bartlett appeared on television*, but the latter does not entail the former. The dataset contains $18,407$ examples (3,916 positive and 14,491 negative). We use the split of the dataset into development (30%) and test sets (70%) chosen by Hosseini et al. (2018) in our experiments.

**Link Prediction Evaluation.** For the link prediction task, we evaluate the models on the test set of the NewsSpike corpus (§4.2) that has 40K triples. For each triple, we compare the link prediction score with the score of a corrupted triple by changing one of the entities in the triple.

## 4.4 Comparison

We compare the following entailment scores for evaluating on the entailment detection dataset.

**MC** is the entailment score based on the Markov chain (3.1), when the link prediction scores are computed only for the predicates we have seen in the corpus. While the link prediction method can assign scores to any possible triple, we report this results to check how the Markov chain model performs compared to the other scores that are directly computed for the triples in the corpus.

**Aug MC** is our novel entailment score that is based on the Markov chain, but augments the matrix $S$ of the MC model with new entries. We use the link prediction method to compute scores on the original set of triples as well as new predicted triples. For each triple $(r, e_1, e_2)$, we compute the score $S_{r,e_1,e'_2}$ for all candidate entities $e'_2$ that have been seen with $e_1$ for any other relation $r'$. We augment the matrix $S$ with the $K$ highest scores. We similarly score $S_{r,e'_1,e_2}$ for all candidate entities $e'_1$ and augment the matrix $S$ with the $K$ high-

[7] Accessed from https://github.com/TimDettmers/ConvE.

[8] We experimented with chaning the learning and dropout rates, but the results did not improve on the validation set.

est scores, accordingly. In our experiments, we used $K = 50$.[9]

**Cos** is the cosine similarity of the embeddings of the relations if the cosine is positive, and 0 otherwise. We also compare to three Sparse Bag-of-Word (SBOW) methods: **Weeds** (Weeds and Weir, 2003), **Lin** (Lin, 1998), and Balanced Inclusion (**BInc**; Szpektor and Dagan, 2008). These similarities check the set of entity-pairs for each relation pair and compute how much one set is included in the other, and/or how much they overlap. Following previous work, we have computed these scores based on the Pointwise Mutual Information (PMI) between the relations and the entity pairs.

**Berant's ILP** is the method of Berant et al. (2011). It computes local similarities and then learns global entailment graphs satisfying transitivity constraints by solving an Integer Linear Programming. We downloaded Berant et al. (2011)'s entailment graphs and tested it on the Levy/Holt's dataset.[10]

For all the above similarities, we report results both in the **local** setting, where the similarities are computed for each relation pair independent of the others and the **global** setting, where we apply the global soft constraints of Hosseini et al. (2018). We apply two sets of global soft constraints: a) Cross Graph which transfers similarities between relations in different, but related typed graphs; and b) Paraphrase Resolution which encourages paraphrase relations to have the same patterns of entailment. We tune the parameters of the global soft constraints on the development set of the Levy/Holt's dataset.

For the link prediction task, we compare the ConvE model with our proposed link prediction score. We test how MC and Aug MC entailment scores can improve the link prediction scores in both local and global settings.

## 5 Results and Discussion

We first compare our proposed entailment score with the previous state-of-the-art results (§5.1) and then show that we can use entailment decisions to improve the link prediction task (§5.2).

### 5.1 Entailment Scores based on Link Prediction

In this section, we compare the variants of our method to the previous state-of-the-art results on the Levy/Holt's dataset. We compute similarity scores and report precision-recall curve by changing the threshold for entailment between 0 and 1. In order to have a fair comparison with Berant's ILP method, we first test a set of rule-based constraints proposed by them (Berant et al., 2011). We also apply the lemma baseline heuristic process of Levy and Dagan (2016) before testing the methods.

Figure 3 shows the precision-recall curve of all the methods in both local (A) and global (B) settings. From the SBOW methods, we only show the BInc score in the graphs as it got the best results on the development set. For Berant's ILP method, we only have one point of precision and recall, as we had access to their entailment graphs for only one sparsity level. In both settings, Aug MC works better than all the other methods. This confirms that the link prediction method is indeed useful for finding entailment relations. Aug MC consistently outperforms MC suggesting that adding the missing entries before forming the Markov chain alleviates the sparsity problem inherent to the entailment task.

Interestingly, while the MC model has access to the same set of entity-pairs as the BInc score, it outperforms it in most of the recall range (especially in the high recall range). Note that the link prediction method might still assign a low score to a triple $(r, e_1, e_2)$ in the corpus if it is not consistent with the other facts. This is especially important when the input triples are noisy. For triples extracted directly from text, the noise might come from various sources such as the relation extraction components (e.g, parsing and named entity linking) or fake or inconsistent news. The MC model seems to be successful in removing the noise from the input triples.[11] The cos similarity is worse than the other methods. This is mainly attributed to the fact that cos is symmetric, while the entailment relation is directional.

We also report area under the precision recall curve. Because the different methods cover different ranges of precision and recall values, we com-

[9] Higher values of $K$ was not feasible on our machines. We performed our experiments on a 32-core 2.3 GHz machine with 256GB of RAM.

[10] Berant et al. (2015)'s entailment graphs yield similar results.

[11] We tested the MC model when the transition probabilities were computed by frequency information or PMI of entity pairs and relations, and the results were worse than the MC model based on the link prediction scores.

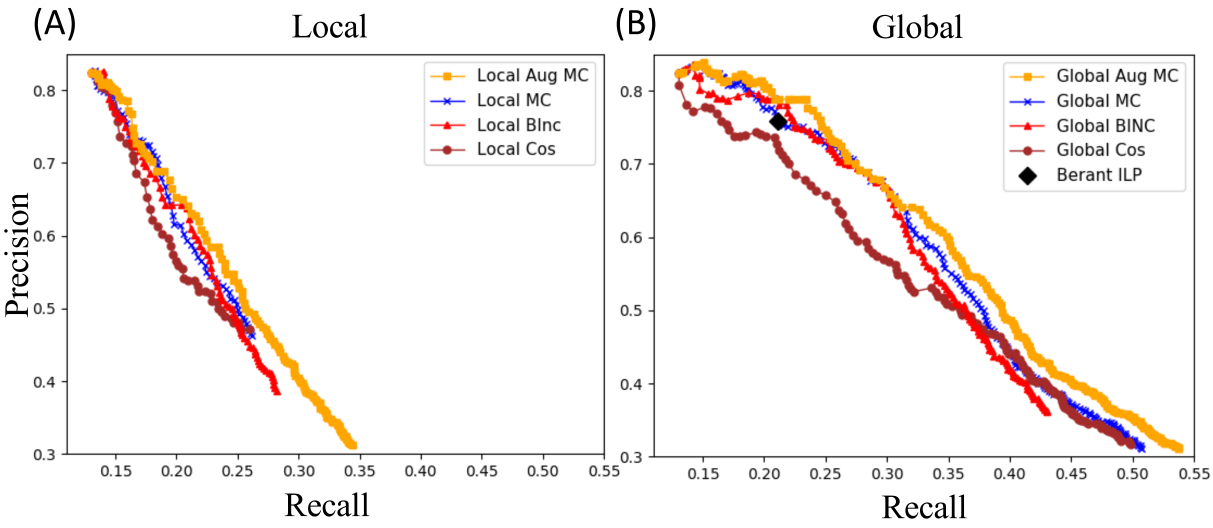

Figure 3: Comparison of the Markov chain (MC) and Augmented Markov chain (Aug MC) models to the BInc score (SBOW model) on Levy/Holt's dataset in local (A) and global (B) settings.

pute area under the precision recall curve for the precision range $[0.5, 1]$, as it is covered by all the baselines and the precision values higher than random are more important for end applications such as semantic parsing or summarization. Table 1 shows the area under precision-recall curves for all the methods. In the global setting, Aug MC shows about 13% improvement relative to the best result of the methods based on SBOW vectors (.187 vs .165). In addition, it is 25% higher relative to the cos score (15%). Similar patterns can be seen in the local setting.

### 5.2 Effect of Entailment Scores for Improving Link Prediction

We now test the proposed method for improving the link prediction score. Each triple $(r, e_1, e_2)$ in the test set is corrupted by either replacing its first or second entity by any possible entities. The candidate entities are then ranked in descending order based on their plausibility score. The original entity is then ranked among all the other entities. We report results using a filtered setting, i.e., we rank test triples against all other triples not appearing in the training, validation or test sets (Bordes et al., 2013). We report Hits@1 (the proportion of the test triples for which the correct entity was ranked as the first prediction), Hits@10, Mean Rank (MR) and Mean Reciprocal Rank (MRR).

Table 2 shows the results of link prediction. We report the results for all entities as well as infrequent entities, where in the latter case we have removed any triple with an entity in the top 20 most frequent entities. In each setting, the first row is the plain ConvE model. We then test how the different variants of our entailment scores change the results. We observe that adding the entailment scores improve the rankings of the correct triples. The value of MRR, Hits@1 and Hits@10 have increased after applying any of the methods for learning entailment scores.

It is interesting to see that the improvements obtained by the different entailment scores are generally consistent with the results on the entailment detection task, i.e., the scores with better results on the Levy/Holt's dataset, show more improvements on this task as well. The change of the mean rank (MR) is more apparent. For example, MR has decreased about 50% when we apply our best method (Global Aug MC) to re-rank the link prediction scores. This means using entailment relations is more useful to improve the link prediction for harder examples. The results of all methods for infrequent entities are worse than the results on all entities; however, we observe the same trends among the different methods.

Note that the amount of the data that is used for all the methods is the same. In particular, we have only used the triples from the NewsSpike corpus for both link prediction and entailment detection tasks and the gain in performance of the both tasks is merely because the two tasks learn complementary information.

## 6 Related Work

**Link Prediction.** In recent years, many link prediction models have been proposed that learn vec-

| | SBOW | | | Link Prediction | | |
|---|---|---|---|---|---|---|
| | Weeds | Lin | BInc | Cos | MC | Aug MC |
| Local | .073 | .074 | .076 | .067 | .079 | **.085** |
| Global | .147 | .149 | .165 | .150 | .174 | **.187** |

Table 1: Area under precision-recall curve (for precision $>$ 0.5) on Levy/Holt's dataset.

| | Hits@1 | Hits@10 | MR | MRR |
|---|---|---|---|---|
| | ALL entities | | | |
| ConvE | 20.36 | 47.93 | 1999.29 | 29.58 |
| + Local MC | 20.66 | 48.64 | 1157.33 | 30.03 |
| + Local Aug MC | **20.68** | 48.90 | 1018.37 | 30.12 |
| + Global MC | **20.68** | 49.13 | 1012.54 | **30.19** |
| + Global Aug MC | 20.64 | **49.16** | **987.13** | **30.19** |
| | INFREQUENT entities | | | |
| ConvE | 19.05 | 45.59 | 2124.71 | 27.94 |
| + Local MC | 19.26 | 46.10 | 1303.56 | 28.25 |
| + Local Aug MC | **19.30** | 46.36 | 1154.06 | 28.33 |
| + Global MC | 19.29 | 46.60 | 1154.28 | 28.41 |
| + Global Aug MC | 19.28 | **46.66** | **1118.09** | **28.43** |

Table 2: Link prediction results on the test set of NewsSpike for all entities (top) and infrequent entities (below). We test the effect of refining ConvE scores with entailment relations.

tor or matrix representations for relations and entities (Socher et al., 2013; Bordes et al., 2013; Riedel et al., 2013; Wang et al., 2014; Lin et al., 2015; Toutanova et al., 2016; Nguyen et al., 2016; Trouillon et al., 2016; Dettmers et al., 2018; Schlichtkrull et al., 2018; Nguyen et al., 2018). These models are trained by assigning higher plausibility scores to correct facts than incorrect ones. For example, the well-known TransE model (Bordes et al., 2013) captures relational similarity between entity pairs by considering a translation vector for the relations connecting them. In particular, it learns embeddings for entities and relations such that $\vec{e_2} - \vec{e_1} \approx \vec{r}$ for any correct triple $(r, e_1, e_2)$. In our experiments we have used ConvE (Dettmers et al., 2018), however, our proposed score can be computed based on any link prediction model and the discovered entailment relations might be useful for improving any link prediction model.

**Entailment Graph Induction.** Entailment graphs are learned by imposing global constraints on local entailment decisions. Berant et al. (2011, 2012, 2015) have used transitivity constraints and applied Integer Linear Programming (ILP) or approximation methods to learn entailment graphs. Hosseini et al. (2018) have used two sets of global soft constraints to: (a) transfer similarities between different but related typed entailment graphs; and (b) encouraging paraphrase relations to have the same patterns of entailments. Our method, in contrast, learns a new entailment score to improve local decisions, which in turn improves the entailment graphs.

**Entailment Rule Injection for link prediction.** There are some attempts in recent years to improve link prediction by injecting entailment rules. Wang et al. (2015) incorporate various set of heuristic rules, including entailment rules, into embedding models for knowledge base completion. They formulate inference as an ILP problem, with the objective function generated from embeddings models and the constraints translated from the rules. Guo et al. (2016) extend the TransE model by defining plausibility scores for grounded logical rules as well as triples and learning entity and relation embeddings that score positive examples higher than negative ones. Guo et al. (2018) take an iterative approach where in each iteration a set of unseen triples are scored according to the current link prediction model and a small set of precomputed logical rules. The new triples and their scores are then used to update the current link prediction model.

The above models need grounding of logical rules. A few recent works do not need grounding and are more space and time efficient (Demeester et al., 2016; Ding et al., 2018). They incorporate logical rules into distributed representations of relations. These models constrain entity or entity-pair vector representations to be nonnegative. They encourage partial ordering over relation embeddings based on implication rules; however, their methods can be only applied to (multi-)linear link prediction models such as ComplEx (Trouillon et al., 2016). In contrast, our method can be applied to any type of link prediction model.

All these methods require entailment rules as their input. In most cases (Wang et al., 2015; Demeester et al., 2016; Guo et al., 2016), the entailment rules are constructed manually, or selected from lexical resources such as WordNet (Miller, 1995). Therefore, the improvement of such methods come from out-of-domain knowledge (manually built lexical resources or expert knowledge), while our entailment rules come from in-domain knowledge, i.e., the same data which is used for link prediction. The number of entailment rules in all the previous models is very small because of scalability issues (at most a few hundred rules in Ding et al. (2018)). In contrast, our method can incorporate millions of automatically discovered entailment rules.

## 7 Conclusion

We have shown that link prediction and entailment graph induction are complementary tasks. We have introduced a new score for entailment detection by performing link prediction on predicate-argument structures extracted from text. We reform the normal knowledge graph representation into a Markov chain with relations and entity-pairs as its states. The score is computed by estimating transition probabilities between the relation states. Our experiments show that the entailment graphs built by our proposed score outperform previous state-of-the-art results because link prediction is effective in filtering noise and adding new facts. We have additionally considered the reverse problem, i.e., using the learned entailment graphs to improve link prediction. Our results show that the two tasks can benefit from each other.

## Acknowledgements

We thank Nathanael Chambers for many useful discussions. The authors would also like to thank the three anonymous reviewers for their valuable feedback. This work was supported in part by the Alan Turing Institute under the EPSRC grant EP/N510129/1. The experiments were made possible by Microsoft's donation of Azure credits to The Alan Turing Institute. The research was supported in part by ERC Advanced Fellowship GA 742137 SEMANTAX, a Google faculty award, a Bloomberg L.P. Gift award, and a University of Edinburgh/Huawei Technologies award to Steedman. Steedman and Johnson were supported by the Australian Research Council's Discovery Projects funding scheme (project number DP160102156).

## References

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*, pages 722–735. Springer.

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient Global Learning of Entailment Graphs. *Computational Linguistics*, 42:221–263.

Jonathan Berant, Ido Dagan, Meni Adler, and Jacob Goldberger. 2012. Efficient Tree-Based Approximation for Entailment Graph Learning. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 117–125, Jeju, Korea.

Jonathan Berant, Jacob Goldberger, and Ido Dagan. 2011. Global Learning of Typed Entailment Rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 610–619, Edinburgh, Scotland, UK.

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 ACM SIGMOD international conference on Management of data*, pages 1247–1250, Vancouver, Canada.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Advances in neural information processing systems*, pages 2787–2795, Lake Tahoe, Nevada, USA.

Ido Dagan, Lillian Lee, and Fernando C.N. Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine learning*, 34(1-3):43–69.

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1389–1399.

Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pages 1811–1818, Honolulu, Hawaii, USA.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. *arXiv preprint arXiv:1805.02408*.

Maayan Geffet and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 107–114, Ann Arbor, Michigan, USA.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 192–202.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4816–4823.

Xavier R. Holt. 2018. Probabilistic Models of Relational Implication. Master's thesis, Macquarie University.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier Holt, Shay Cohen, Mark Johnson, and Mark Steedman. 2018. Learning Typed Entailment Graphs with Global Soft Constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2016. Distributional Inclusion Hypothesis for Tensor-based Composition. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2849–2860, Osaka, Japan.

Omer Levy and Ido Dagan. 2016. Annotating Relation Inference in Context via Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 249–255, Berlin, Germany.

Mike Lewis and Mark Steedman. 2014a. $A^*$ CCG Parsing with a Supertag-factored Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 990–1000, Doha, Qatar.

Mike Lewis and Mark Steedman. 2014b. Combining Formal and Distributional Models of Temporal and Intensional Semantics. In *Proceedings of the ACL Workshop on Semantic Parsing*, pages 28–32, Baltimore, Maryland, USA.

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 768–774, Montreal, Canada.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187. AAAI Press.

Xiao Ling and Daniel S. Weld. 2012. Fine-Grained Entity Recognition. In *Proceedings of the National Conference of the Association for Advancement of Artificial Intelligence*, pages 94–100, Toronto, Canada.

George A Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A capsule network-based embedding model for knowledge graph completion and search personalization. *arXiv preprint arXiv:1808.04122*.

Dat Quoc Nguyen. 2017. An Overview of Embedding Models of Entities and Relationships for Knowledge Base Completion. *arXiv preprint arXiv:1703.08098*.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of NAACL-HLT*, pages 460–466.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, Cambridge, MA.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-Scale Semantic Parsing without Question-Answer Pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation Extraction with Matrix Factorization and Universal Schemas. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, USA.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

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, Lake Tahoe, Nevada, USA.

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. ACM.

Idan Szpektor and Ido Dagan. 2008. Learning Entailment Rules for Unary Templates. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 849–856, Manchester, UK.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1434–1444, Berlin, Germany.

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 International Conference on Machine Learning*, pages 2071–2080, New York City, New York, USA.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the*

*Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. AAAI Press.

Julie Weeds and David Weir. 2003. A General Framework for Distributional Similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Sapporo, Japan.

Congle Zhang and Daniel S. Weld. 2013. Harvesting Parallel News Streams to Generate Paraphrases of Event Relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786, Seattle, Washington, USA.