Open-Domain Contextual Link Prediction and its Complementarity with Entailment Graphs

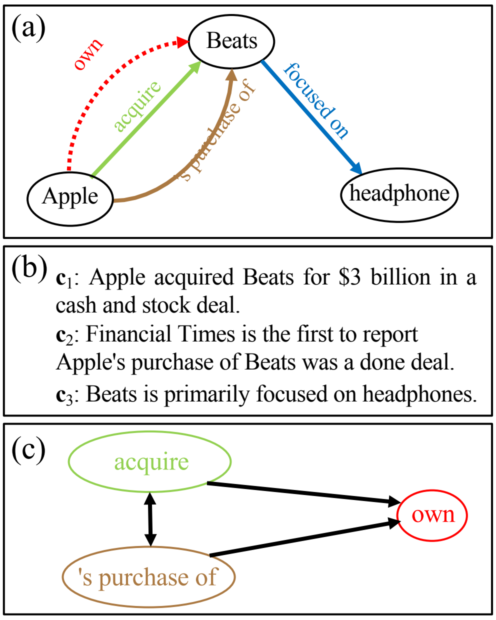
Anonymous EMNLP submission

Abstract

An open-domain knowledge graph (KG) has entities as nodes and natural language relations as edges, and is constructed by extracting (subject, relation, object) triples from text. The task of open-domain link prediction is to infer missing relations in the KG. Previous work has used standard link prediction for the task. Since triples are extracted from text, we can ground them in the larger textual context in which they were originally found. However, standard link prediction methods only rely on the KG structure and ignore the textual context of the triples. In this paper, we introduce the new task of open-domain *contextual* link prediction which has access to both the textual context and the KG structure to perform link prediction. We build a dataset for the task and propose a model for it. Our experiments show that context is crucial in predicting missing relations. We also demonstrate the utility of contextual link prediction in discovering *out-of-context* entailments between relations, in the form of entailment graphs (EG), in which the nodes are the relations. The reverse holds too: out-of-context EGs assist in predicting relations in context.

1 Introduction

A knowledge graph (KG) is constituted by a set of (subject, relation, object) triples such as (*Apple, acquire, Beats*). KGs have entities (subjects and objects) as nodes and relations as labeled edges. Manually-built KGs such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014) or DBPedia (Lehmann et al., 2015) have a fixed set of hand-built relations. In contrast, the relation-labels of *open-domain* KGs are obtained from text rather than fixed. The open-domain KGs can be constructed by applying parsers or open-information extraction methods to text (Hosseini et al., 2019; Gupta et al., 2019; Broscheit et al., 2020).



$\mathbf{c}_1$: Apple acquired Beats for $3 billion in a cash and stock deal.
$\mathbf{c}_2$: Financial Times is the first to report Apple's purchase of Beats was a done deal.
$\mathbf{c}_3$: Beats is primarily focused on headphones.

Figure 1: a) Part of an example KG. The relation *own* is missing, but can be predicted from the rest of the KG and the triple contexts using contextual link prediction. b) The contexts $\mathbf{c}_1$ and $\mathbf{c}_2$ from which we have extracted the KG triples. The contextual link prediction task predicts relations that hold between the entity pair in a grounded triple. For example, we predict that the relation *own* should be added between *Apple* and *Beats*. c) An example EG of type *company, company*. The contextual link prediction and EG learning tasks are complementary. For example, *acquire* → *own* from the EG can independently be used to add missing *own* relation to the KG.

Open-domain link prediction is the task of adding relation edges that are missing from the graph because the corresponding triple was not found in the text (Hosseini et al., 2019; Broscheit et al., 2020). Figure 1a shows part of an example open-domain KG, in which the triple (*Apple, own, Beats*) is missing, but can be inferred using link prediction over all entities in the complete KG.

Previous work has applied standard link prediction methods such as TransE (Bordes et al., 2013), ConvE (Dettmers et al., 2018), or TuckER (Balazevic et al., 2019) to open-domain triples. These methods have been shown to be effective in learning the KG structure, but they are sub-optimal for open-domain link prediction because they ignore the textual context of the triples. Since the triples are extracted from text, they can be automatically grounded back to their contexts. Hence, in addition to the KG structure, the triple contexts can be used as input to the link prediction task.

Figure 1b shows the context sentences that have given rise to the partial KG in Figure 1a.[1] There are multiple clues in the contexts such as *deal*, *$*, *cash*, *stock*, and *Financial Times*, that could be used in addition to the triples in the rest of the KG, to predict that the triple (*Apple, owns, Beats*) should be added. This is because these clues could have been seen around occurrences of other entity pairs of the same type (e.g., *Facebook* and *Whatsapp*) that are connected by *acquire*, *'s purchase of*, and *own* relations.

In this paper, we propose the new task of *contextual link prediction* for such open-domain graphs: Given a triple (e_1, r, e_2) grounded in context with the relation r holding between the entities e_1 and e_2, our goal is to predict all the other relations that hold between the two entities. We present a model that uses contextualized relation embeddings to predict new relations. We start with BERT (Devlin et al., 2019) pre-trained embeddings and fine-tune them with a novel unsupervised contextual link prediction objective function. After training the contextual link prediction model, we can add missing relations to the KG (e.g., *own in* in Figure 1a) by predicting the relations that hold between the entities of triple mentions in context (e.g., the context $\mathbf{c}_1$ in Figure 1b). Our experiments show that the proposed model for the contextual link prediction task significantly outperforms standard link prediction in open-domain KG completion.

In addition, we investigate the interplay between contextual link prediction and *out-of-context* entailment between relations, in the form of entailment graphs (EG). An EG has typed relations as nodes and entailment relation as directed edges (Berant et al., 2010, 2011, 2015; Hosseini et al., 2018). The type of each relation is determined by the types of its two entities. Figure 1c shows a fragment of an EG showing that for example *acquire* entails *own*. Similar to open-domain KGs, EGs are constructed based on extracted triples from text. The entailment between two relations is predicted by computing a directional entailment score between them.

It has been recently shown that the two tasks of open-domain link prediction and EG learning are complementary (Hosseini et al., 2019). EGs suffer from sparsity since many correct entailment rules are not directly supported by the extracted triples from the text. The EGs can be improved by augmenting the extractions with novel triples from standard link prediction models. Conversely, explicit entailments from EGs are shown to be useful in predicting missing links in the KG.

We show a similar relationship between contextual link prediction and the EG learning tasks. As in that previous work, we augment the set of extracted triples with novel predictions, but we use contextual link prediction instead of standard link prediction. We define a new entailment score which we use to build new state-of-the-art EGs when tested on a challenging relation entailment dataset. Our results show that contextual link prediction produces higher quality triples for augmentation than standard link prediction. Conversely, we also show that EGs in turn contain complementary information that can be combined with contextual link prediction to further improve the open-domain KG completion results.[2] Our main contributions are the following.

- We propose a new contextual link prediction task and present a model for it.
- We show that our proposed model outperforms standard link prediction models in open-domain KG completion.
- We propose a new entailment score that uses the extracted triples as well as predicted ones from contextual link prediction. We build state-of-the-art EGs as tested on a challenging entailment dataset.
- We show that EGs in turn improve contextual link prediction.
- We release a dataset containing the extracted triples grounded in context, for future research.

2 Related Work

Link Prediction. Existing link prediction models take as input a set of triples in a KG. These models are trained to assign high scores to correct triples

[1] We assume having access to an entity linked corpus. The entities consist of both proper nouns (e.g., *Apple*) and common nouns (e.g., *headphone*).

[2] We release our code and the extracted open-domain KG.

and low scores to incorrect ones (Bordes et al., 2013; Trouillon et al., 2016; Dettmers et al., 2018; Kazemi and Poole, 2018; Balazevic et al., 2019). The link prediction models have been mainly applied to existing KGs such as Freebase (Bollacker et al., 2008) or DBPedia (Lehmann et al., 2015). Recently, Hosseini et al. (2019) and Broscheit et al. (2020) have performed open-domain link prediction on triples extracted from raw text, but unlike our work, they only use the KG structure, and are not able to take advantage of the context in which the triples were found. KG-BERT (Yao et al., 2019) and the model of Kim et al. (2020) use contextual representation for KG completion. However, they form synthetic token sequences by concatenating entity descriptions and relation tokens. In our model, we use the natural text associated with the triples as the context.

Relational Entailment Graphs. Relational entailments capture meaning postulates such as *acquire* entails *own*. In order to disambiguate the context, the models learn entailment between typed relations, where the type of each relation is determined by the types of its two entities. Earlier attempts take a *local* approach and predict entailment rules independently from each other (Lin and Pantel, 2001; Szpektor et al., 2004; Szpektor and Dagan, 2008; Yates and Etzioni, 2009; Schoenmackers et al., 2010). Berant et al. (2010, 2011, 2012, 2015) and Hosseini et al. (2018) propose a *global* approach where the dependencies between the entailment rules are taken into account. They first build a local typed EG for any plausible type pair. They then build global EGs that satisfy soft or hard constraints. The constraints consider the structures both across typed EGs and inside each graph. In this work, we improve the local entailment scores, which in turn improves the global EGs.

Extracting Factual Knowledge from Pre-Trained Language Models. Recently, there has been an increasing interest in extracting factual knowledge from pre-trained Language Models. These works form a prompt where an entity is missing (e.g., *Apple acquire [MASK]*), and ask the language models to predict the masked entity (Petroni et al., 2019, 2020; Jiang et al., 2020; Bouraoui et al., 2020; Haviv et al., 2021). These models do not probe for relations for two reasons: a) The surface of relations usually span multiple tokens which poses technical challenges in probing. b) Relations can generally be expressed in many different ways. In contrast, our model predicts multi-token relations.

The matching-the-blank (MTB) model (Soares et al., 2019) learns relation embeddings by encouraging relations that share the same entity-pairs to have similar embeddings. This is similar to our training, but has two main differences: First, our contextual link prediction model outputs a directional score between relations in context (e.g., *acquire* in Figure 1) and query relations (e.g., *own*), while MTB learns a symmetric similarity score. Second, we can predict a score for any query relation as long as the relation is previously observed somewhere in the corpus with any other entities. For example in Figure 1, we can predict *own* given the context sentences $\mathbf{c}_1$ or $\mathbf{c}_2$, but MTB needs a mention for the triple (*Apple, own, Beats*) in another sentence. It might be possible to synthesize a query sentence containing the triple (e.g., *"Apple owns Beats"*), but their model is trained on real text with the triple grounded in a larger context and is not guaranteed to work well on synthesized sentences. Replacing the relation's surface form could be another possibility, but that is non-trivial because of multi-word relations with gaps (e.g., *focused on* in Figure 1) and the need to match the relation's tense and aspect.

In addition, the above models are tested on manually constructed KGs with few relations, whereas we test our contextual link prediction model on many relations extracted from open-domain text.

3 Contextual Link Prediction

In this section, we first discuss the notation and define the contextual link prediction task. We then present our model and training for the task.

3.1 Notation and Task Definition

Let $\mathcal{E}$ denote the set of all entities (e.g., *Barack Obama*; *message*), $\mathcal{E}$ denote the set of all entity types (e.g., *Person*; *Thing*) and $\mathcal{R}$ denote the set of all typed relations. We consider binary relations where each relation has two entities. Hence, each relation has two types, one for each entity slot, e.g., *born in(Person,Location)*. We define $\mathcal{R}(t_1, t_2)$ as the set of relations with types t_1, t_2, or t_2, t_1. For example, *$\mathcal{R}$(Person, Location)* includes *born in(Person,Location)*, *visit (Person,Location)*, *birthplace of (Location,Person)* etc. Similarly, we define $\mathcal{R}(e_1, e_2)$ as the set of relations $r \in \mathcal{R}$ such that

(e_1, r, e_2) is a valid triple. For example, *$\mathcal{R}$(Barack Obama, Hawaii)* includes *born in*[3], *visit*, etc.

The link prediction as well as entailment can hold between relations with the same entity order or the reverse order. For example, *born in(Person,Location)* predicts *birthplace of (Location,Person)*. When the two entity types are unequal, it is straightforward to see whether the order of entities in the two relations are identical or not. However, when the two entity types are identical, we keep two copies of the typed relations one for each entity order. This way, we can model link prediction and entailment when entity orders are reversed. For example, the relation *acquire(company$_1$,company$_2$)* predicts *be part of (company$_2$,company$_1$)*. We specify the entity order of a relation $r \in \mathcal{R}$ by a flag $o(r) \in \{0, 1\}$. For relations with unequal types, we do not need the flag and simply set $o(r) = 0$. For relations with identical types, we set $o(r) = 0$ if the entities are in the original order and $o(r) = 1$, otherwise. We explain the usage of the order flag in Section 3.2.

A triple mention is a triple grounded in its textual context. We define a triple mention as a tuple $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$, where $r \in \mathcal{R}$ is a relation and $e_1, e_2 \in \mathcal{E}$ are entities. The sub-word token sequence $\mathbf{c} = [c_0, \ldots, c_n]$ is the textual context of the triple including the surface form of the relation and entity-pair.[4] The pair $\mathbf{s} = (s_1, s_2)$ indicates the indices of the first and last relation tokens. An example triple mention in Figure 1b is *(Apple,acquire,Beats,*$\mathbf{c}_2$*,*$[9, 11]$*)*.[5] We denote by $\mathcal{D}{=}\big[(e_{i,1}, r_i, e_{i,2}, \mathbf{c}_i, \mathbf{s}_i)\big]_{i \in \{1,\ldots,N\}}$ the set of all triple mentions.

We define the contextual link prediction task as follows: Given a triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$, the goal is to predict all query relations $q \in \mathcal{R}(t_1, t_2)$ that hold between the entity-pair (e_1, e_2), where t_1 and t_2 are the types of the two entities.

3.2 Model

Our model computes the probability $\Pr(q|m)$ that (e_1, q, e_2) is a valid triple conditioned on the triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$. In this section, we propose our model, named Contextual and Out Of Context Embeddings (COOCE), which is based on two different embedding spaces for relations:

First, the relation r in the triple mention m has a contextualized embedding encoded by the vector $\vec{m} \in \mathbb{R}^d$, where d is the number of embedding dimensions. Let $[\vec{h}_0, \ldots, \vec{h}_n]$ be the contextualized embeddings of the context c, where $\vec{h}_i \in \mathbb{R}_d$. In our experiments, we use the contextualized embeddings of the relation's token(s) as the embedding vector of the triple mention. For multi-token relations, we use the average embedding vectors of the start and end tokens, i.e., $\vec{m} = (\vec{h}_{s_1} + \vec{h}_{s_2})/2$. As discussed in Section 3.1 we keep two copies of the relations when the entity types are the same, one for each entity order. However, these two copies should have different embeddings. We multiply the contextualized embedding with a matrix $A_0 \in \mathbb{R}^{d \times d}$, if the entities are in the original order (i.e., $o(r) = 0$), and $A_1 \in \mathbb{R}^{d \times d}$, if they are in the reverse order (i.e., $o(r) = 1$). This allows us to disentangle the embeddings of relations with original and reverse entity orders.

Second, each relation has an out-of-context embedding taken from an embedding weight matrix that is learned from scratch from the KG. We use the out-of-context embedding $\vec{q} \in \mathbb{R}^d$ to encode the query relation. We predict high link prediction score if the dot product between $\vec{m}A_{o(r)}$ and $\vec{q}$ has a high value. In particular, we define the contextual link prediction score as:

$$\Pr\big(q|m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})\big) = \sigma(\vec{m}A_{o(r)} \cdot \vec{q}) \\ = \frac{1}{1 + \exp(-\vec{m}A_{o(r)} \cdot \vec{q})}. \tag{1}$$

Equation 1 estimates the probability that the relation q holds between the entity-pair. It can be applied to any relation $q \in \mathcal{R}(t_1, t_2)$ and predict that multiple relations are compatible with the context. Figure 2 shows an example.

We encode the query relations with learned out-of-context embeddings rather than contextualized embeddings for two reasons: a) The model can be applied to relations $q' \notin \mathcal{R}(e_1, e_2)$. This is useful for KG completion because the goal is to find novel triples that are likely to be correct, but are not found in the text. *Note that if we were modeling the query relation with contextualized embeddings, that relation should have also been observed with the same entity-pair somewhere else in the corpus, hence it would not generate a novel triple.* b) While the above score uses the dot product between em-

[3] For simplicity, we drop the entity types in our examples when they are obvious.

[4] c_0=[CLS] and c_n=[SEP] are special start and end tokens.

[5] The indices of the relations could be shifted to the right if a word is divided into sub-words during tokenization.

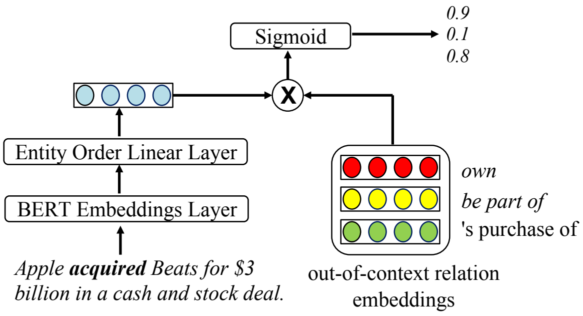

Sigmoid
0.9
0.1
0.8
X
Entity Order Linear Layer
BERT Embeddings Layer
*Apple **acquired** Beats for $3 billion in a cash and stock deal.*
own
be part of
's purchase of
out-of-context relation embeddings

Figure 2: An example of contextual link prediction. The relation token is boldfaced. The output probabilities correspond to the input out-of-context relations.

bedding vectors, it is still *asymmetric* as it uses embeddings from different spaces for the relations r and q. This is a desired property since contextual link prediction is directional, not symmetric. In the example in Figure 1a and 1b, we should predict *own* given *acquire*, but we should not necessarily predict *acquire* given *own*.

3.3 Training

Given observed triple mentions $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s}) \in \mathcal{D}$, we train a model to assign high contextual link prediction scores to relations q that hold between the entity-pairs ($q \in \mathcal{R}(e_1, e_2)$), and low scores to relations q' that do not hold between the entity-pairs ($q' \notin \mathcal{R}(e_1, e_2)$). This can be seen as a multi-label classification task. Each triple mention is an example with a total number of $|\mathcal{R}(t_1, t_2)|$ labels, where t_1 and t_2 are the types of e_1 and e_2. Each label correspond to one of the relations, e.g., there is one label for the relation *own* given the triple mention in Figure 2.. Among the $|\mathcal{R}(t_1, t_2)|$ labels (relations), $|\mathcal{R}(e_1, e_2)|$ are positive and $|\mathcal{R}(t_1, t_2)| - |\mathcal{R}(e_1, e_2)|$ are negative.

We initialize the contextualized embeddings with BERT pre-trained embeddings (Devlin et al., 2019). We initialize the out-of-context embeddings (i.e., $\vec{q}$), and the matrices A_0 and A_1 randomly. We fine-tune the contextualized embeddings and learn the other model parameters by minimizing the following binary cross entropy loss:

$$\mathcal{L} = - \sum_{m=(e_1,r,e_2,\mathbf{c},\mathbf{s}) \in \mathcal{D}} \Big[\sum_{q \in \mathcal{R}(e_1,e_2)} \log \Pr\left(q|m\right)$$

$$+ \sum_{q' \in \mathcal{R}(t_1,t_2) \setminus \mathcal{R}(e_1,e_2)} \log(1 - \Pr\left(q'|m\right)) \Big]. \quad (2)$$

4 Scoring Entailment between Relations

In this section, we describe our new entailment score. We augment the set of input triples with novel triples from the contextual link prediction model. We compute entailment scores $0 \leq w_{rq} \leq 1$ between relations r and q.

We propose an entailment score, named COOCE Augmented Markov chain (MC), between relations similar to the ConvE Augmented MC score of Hosseini et al. (2019). The previous work defines the entailment score using a standard link prediction score such as ConvE (Dettmers et al., 2018). Our entailment score is defined based on a contextual link prediction score such as our COOCE score.

We form a bipartite graph with relations on one side and triple mentions on the other side (Figure 3).[6] We define the entailment score as the probability that a random walk (with length 2) from one relation ends in another. In particular, we define a Markov chain with relation states $\langle r \rangle$ as well as triple mention states $\langle m \rangle$ as its nodes. Each relation r has directed edges to its triple mentions $m \in \mathcal{D}(r)$, where $\mathcal{D}(r)$ is defined as the set of all triple mentions of r. On the other hand, each mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$ has directed edges to a set of relations $\mathcal{R}(m) = \mathcal{R}(e_1, e_2) \cup \mathcal{R}'(m)$, where $\mathcal{R}(e_1, e_2)$ is the set of observed relations for the entity pair and $\mathcal{R}'(m) \subseteq \mathcal{R}(t_1, t_2) \setminus \mathcal{R}(e_1, e_2)$ contains a set of relations with high contextual link prediction scores. For a relation $r' \in \mathcal{R}'(m)$, the triple (e_1, r', e_2) has not been observed in the text corpus, but is likely to be valid. We augment the Markov chain with such connections from mentions m to relations r' (Section 5.3). Figure 3 shows an example Markov chain, where dotted links correspond to novel triples from contextual link prediction. We define the transition probabilities from relations to mentions uniformly, and from mentions to relations as normalized contextual link prediction scores:

$$\Pr(\langle m{=}(e_1, r, e_2, \mathbf{c}, \mathbf{s}) \rangle | \langle r \rangle) = \frac{1}{|\mathcal{D}(r)|}$$

$$\Pr(\langle q \rangle | \langle m \rangle) = \frac{\Pr\left(q|m\right)}{\sum_{r \in \mathcal{R}(m)} \Pr(r|m)},$$

where $\Pr\left(q|m\right)$ is defined in Equation 1. We define the COOCE Augmented MC entailment score as:

[6] Previous work forms a bipartite graph with relations on one side and *entity-pairs* on the other side (Hosseini et al., 2019).

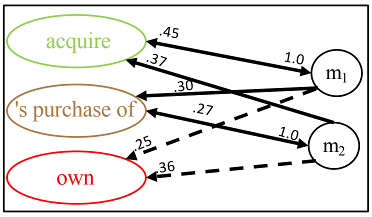


Figure 3: An example Markov chain. It has the relations *acquire*, *'s purchase of*, and *own* (r_1, r_2, and r_3) on the left, and the triple mentions $m_1 = (e_1, r_1, e_2, \mathbf{c}_1, \mathbf{s}_1)$ and $m_2 = (e_1, r_2, e_2, \mathbf{c}_2, \mathbf{s}_2)$ on the right, where e_1 and e_2 are *Apple* and *Beats*, $\mathbf{c}_1$ and $\mathbf{c}_2$ are the same contexts as in Figure 1, and $\mathbf{s}_1$ and $\mathbf{s}_2$ are the indices of relation tokens. The edge weights show transition probabilities. The dotted lines correspond to novel triples discovered by contextual link prediction that help adding the edge *acquire* entails *own* to the EG.

$$\Pr(\langle q \rangle | \langle r \rangle) = \sum_{m \in \mathcal{D}(r)} \Pr(\langle q \rangle | \langle m \rangle) \Pr(\langle m \rangle | \langle r \rangle).$$

We use our new entailment score as the local entailment scores to build global entailment graphs (Section 5.3).

5 Experimental Setup

We discuss the details of the corpus, training and EG building.

5.1 Text-Corpus with Triple Mentions

We perform our experiments on the NewsSpike corpus that contains 550K news articles from various news sources (Zhang and Weld, 2013). Hosseini et al. (2018) process the corpus with a Combinatory Categorial Grammar (CCG; Steedman, 2000) semantic parser and extract typed and entity-linked triples. The entities are linked to Freebase and assigned one of the 49 first-level FIGER types (Ling and Weld, 2012).

The parser they have used outputs triples in addition to the indices of relation tokens. We reparse the corpus and record each triple coupled with its context and the indices of its relation tokens. This yields $|\mathcal{D}| = 8.5$M triple mentions for $|\mathcal{K}| = 3.9$M unique triples. The number of relations is $|\mathcal{R}| = 304$K with a total number of 346 entity type pairs.

5.2 Training Details

We implemented our model using the Hugging Face transformers library (Wolf et al., 2019). We used Adam (Kingma and Ba, 2015) with linear decay of learning rates to minimize the loss function defined in Equation 2. We fine-tune BERT-base pre-trained embeddings.[7]

We randomly split the triple mentions into training (95%), development (2.5%) and test (2.5%) sets. We perform the split so that each entity-pair (e_1, e_2) and its reverse are present in only one of the sets. This constraint is important in evaluating the results of the contextual link prediction model since if we simply split randomly, identical triples (e_1, r, e_2) might exist in training, development, and test sets (in different contexts). Therefore, a simple model that memorizes $\mathcal{R}(e_1, e_2)$ can predict the missing links correctly as long as (e_1, e_2) is observed in a triple mention in the training set.

We use a mini-batch size of $b = 64$ triple mentions. We construct mini-batches in a way that each of them consists of triple mentions with the same entity type pairs. Recall that (Section 3.3 and Figure 2) each triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$ is considered as an example with $|\mathcal{R}(t_1, t_2)|$ possible labels (relations) for multi-label classification. Among those, $|\mathcal{R}(e_1, e_2)|$ are positive, i.e., the relations that hold between the entity-pair and $|\mathcal{R}(t_1, t_2)| - |\mathcal{R}(e_1, e_2)|$) are negative. This causes a class imbalance problem, especially for type-pairs with many relations (up to around 50K), since the number of positive relations are typically ≤ 100 that leaves almost all of the relations as negative.

To alleviate this problem, we train on all positive relations, but choose a small subset of relations as candidate negatives: a) For each triple mention m, we use the positive labels from other triple mentions in the batch as negative labels, if those are not already among the positive labels of m. b) We also choose a random subset of the other relations with the same entity types as negative candidates for the whole batch. This random subset has the size of up to the positive relations of the whole mini-batch (depending on the number of relations with types (t_1, t_2)). The training data consists of 8.1M triple mentions. It has a total of 435M positive labels (54 positive labels per triple mention on average) and a total of 7128M negative labels.

We tuned hyperparameters by maximizing the mean average precision (MAP) of contextual link prediction in the development set. To compute

[7] We also tried RoBERTa-base (Liu et al., 2019), but the results were similar. We could not use BERT-large or RoBERTa-large because of memory constraints. We performed experiments on NVIDIA P102 GPUs with 11GB of memory.

the MAP, we consider each triple mention as an example, and rank the set of positive and negative relations according to their predicted scores from highest to lowest. We compute average precision for each triple mention, and then compute their mean value. We discuss hyper-parameter tuning details in Appendix C.

5.3 Building Entailment Graphs

After training COOCE, we compute local entailment scores (Section 4). We then apply the global soft constraints of (Hosseini et al., 2018) to learn global entailment scores. We build EGs by applying a threshold on the entailment scores, either local or global scores. For a threshold $\delta > 0$, we build EGs where nodes are relations $r \in \mathcal{R}$, and edges include (r, q) with $w_{rq} \geq \delta$.[8] In our experiments, we change the threshold in the range $[0, 1]$ to build and evaluate EGs with varying degrees of confidence. In order to form the Markov chain (Figure 3), we first connect each triple mention m to its observed relations $\mathcal{R}(e_1, e_2)$. If the number of observed relations is less than $K{=}100$, we augment the Markov chain by connecting the mentions to $100{-}K$ novel highest scoring relations (i.e., $\mathcal{R}'(m)$ as defined in Section 4).

6 Results and Discussion

We first evaluate our proposed method for the open-domain contextual link prediction task. We then evaluate our EGs on an entailment dataset.

6.1 Evaluating Contextual Link Prediction

We evaluate our proposed method, COOCE, against various baselines. We compute the MAP of predicting query relations q that hold between the entity-pairs in a triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$. We assume all models predict the trivial relation r correctly.[9] Standard link prediction is usually evaluated by predicting the entity e_2 given the first entity and the relation, i.e., $(e_1, r, ?)$. In our experiments, we predict the correct relations holding between the entity-pair, i.e., $(e_1, ?, e_2)$. We compare the following models.

COOCE is our novel model that calculates $\Pr(q|m)$ (Equation 1) as the contextual link prediction score. **Standard Link Pred (ConvE/TuckER)** are based on non-contextual link prediction. We used ConvE (Dettmers et al., 2018) and TuckER (Balazevic et al., 2019), two of the state-of-the-art link prediction models.

SINGLE MODELS	
Standard Link Pred (ConvE)	.230
Standard Link Pred (TuckER)	.263
COOCE MC EG	.317
COOCE Aug MC EG	.328
COOCE	**.333**
COMBINED MODELS	
COOCE + COOCE MC EG	.355
COOCE + COOCE Aug MC EG	**.357**
ABLATION STUDIES	
COOCE w/o Entity Order Flag	.319
COOCE w/o BERT Layers Update	.321

Table 1: MAP of relation prediction given triple mentions evaluated on the NewsSpike test set.

COOCE MC EG is the EGs based on our novel entailment score defined in Section 4, but without any augmented triples (no dotted lines in Figure 3). **COOCE Aug MC EG** is similar, but uses augmented triples. We use the entailment score w_{rq} to decide whether the relation q should be added to the KG. While textual contexts have been used to compute the entailment scores, the EG baselines are out-of-context. They only look at the entailment score between the relations r and q, but do not use the textual contexts c of the triples. In addition, we consider the combination of COOCE and the EGs: **COOCE + COOCE (Aug) MC EG**, is the linear summation $\beta \Pr(q|m) + (1 - \beta) w_{rq}$, where $\beta \in [0, 1]$ is a hyper-parameter.[10]

Table 1 shows the results. Among the single models, COOCE performs the best. It outperforms the EGs and standard link prediction models that do not use the textual context of the triples, confirming that our proposed model can effectively use the context while performing KG completion. The COOCE Aug MC EG performs better than COOCE MC EG showing that our augmentation technique improves the basic EGs.

Combining COOCE (contextual) and EGs (out-of-context) yields further improvements showing that EGs contain complementary information that further strengthen contextual link prediction.

We perform ablation of COOCE. We tested the model without the entity order flag, i.e., we only use the projection matrix A_0, but not A_1, for all relations regardless of their entity order flag. In addition, we freeze the BERT layers, which means just using BERT as a feature extractor. In both

[8] We learn a separate graph for each type-pair.

[9] Without this assumption, the results of models that do not directly use the extracted relation r will drop.

[10] We tuned $\beta = 0.05$ using the NewsSpike dev set.

	COOCE Improves EGs	
Triple	Microsoft, **is committed to**, success	
Predictions	Microsoft, **builds**, success	↓
	Microsoft, **switches to**, success	↓
	Microsoft, **'s**, success	↑
	Microsoft, **achieves**, success	↑
	Microsoft, **hopes for**, success	↑
	EGs Improve COOCE	
Triple	Apple, **is working on**, watch	
Predictions	Watch, **falls on**, Apple	↓
	Apple, **'s**, watch	↑
	Apple, **has**, watch	↑
	Apple, **launches**, watch	↑
	Apple, **tests**, watch	↑

Table 2: Extracted triples and example predictions for relations of types (*organization,thing*). Top: The context is *Microsoft is committed to the long term success of the entire PC ecosystem.* COOCE improves (downward or upward arrows) EGs. Bottom: The context is *Apple is working on a high-tech watch.* EGs improve COOCE.

cases, we observe a performance drop.

We perform qualitative analysis to check the complementarity of the two approaches. Table 2 (top) shows an example from NewsSpike where the contextual link prediction model improves the results of the EGs. The extracted triple is *Microsoft, is committed to, success*. The EGs predict high scores for wrong relations such as *Microsoft, builds, success*. This is because the typing system has assigned the general type *thing* to the entity *success* as well as many other entities such as *relationship*.[11]. The entailment signal comes from extractions such as *NATO, is committed to, relationship* and *NATO, builds, relationship*. Therefore, the EGs conflate different senses of the relation *build*. However, COOCE disambiguates the context. In addition, the scores of some correct relations (e.g., *achieves*) are increased by COOCE. On the other hand, Table 2 (bottom) shows an example where EGs perform better than contextual link prediction. For example, the embeddings of some infrequent relations such as *falls on* have not been learned well and they get a high contextual score by COOCE, but the EGs do not contain these wrong predictions.

6.2 Evaluating Entailment Graphs

We compare the EGs obtained by our proposed entailment score (Section 4) with previous state-of-the-art EGs on the Levy/Holt's entailment dataset (Levy and Dagan, 2016; Holt, 2018). The dataset contains $18,407$ examples (3,916 positive and 14,491 negative) split into development (30%) and test (70%) sets. Each example has a premise triple which either entails a hypothesis triple (positive label), or does not entail it (negative label). For instance, *Cadmium, is released into, the air* entails *Cadmium, is found in, the air*. We use the entailment score between the typed relations of each example such as *released into (Chemical_element, Thing)* and *is found in (Chemical_element, Thing)*. We predict positive if the score is greater than or equal to a threshold, and negative, otherwise. We plot precision-recall curves by changing the threshold between $[0, 1]$. Similar to Hosseini et al. (2018, 2019), we report the area under the precision-recall curves for precisions >0.5.

We evaluate the EGs obtained by the following entailment scores. **COOCE MC** is our novel entailment score, but without augmented triples. **COOCE Aug MC** is our novel entailment score with augmented triples. These are the same models that we tested in Section 6.1. **ConvE/TuckER MC** is the model of Hosseini et al. (2019), where entailment scores are computed based on a Markov chain between relations on one side and entity-pairs on the other (as opposed to triple mentions in our proposed model). The transition probabilities are computed based on a standard link prediction method such as ConvE or TuckER. The previous work had only reported results with ConvE, but we also repeated their experiments with TuckER. **ConvE/TuckER Aug MC** is similar, but it augments the triples using standard link prediction. Balanced Inclusion (**BInc**) is a Sparse Bag-of-Word model (Szpektor and Dagan, 2008) used in Hosseini et al. (2018).

Table 3 shows the results of EGs in local and global settings. The plots are shown in Appendix A. ConvE/TuckER MC and COOCE MC models only use the extractions from the text-corpus, but use different link prediction scores to compute transition probabilities in the MCs. COOCE MC gets better results than the other two models. The ConvE/TuckER Aug MC, and our novel model, COOCE Aug MC, augment the KG with additional triples. They all improve the results compared to the models without augmentation. They alleviate the sparsity of EGs by adding more connections between the relations (e.g., *acquire* → *own* in Fig-

[11] The type *thing* is assigned to entities that are not linked to any entity in Freebase or their Freebase types do not have a mapping to FIGER types.

	Local	Global
BInc	.076	.165
ConvE MC	.079	.174
ConvE Aug MC	.085	.187
TuckER MC	.071	.162
TuckER Aug MC	.082	.184
COOCE MC	.084	.176
COOCE Aug MC	**.096**	**.195**

Table 3: Area under the precision-recall curves of EGs on the Levy/Holt's dataset (precision >0.5).

ure 3). However, COOCE Aug MC outperforms ConvE/TuckER Aug MC, i.e., the previous state-of-the-art EGs, in both local and global settings. This confirms that contextual link prediction is more effective than standard link prediction in finding new high-quality triples to augment the original ones.

7 Conclusions

We have introduced the contextual link prediction problem and proposed a model (COOCE) for it. We trained COOCE on a corpus of triple mentions. We have shown that our model outperforms standard link prediction models in completing an open-domain KG. We used the model to assign scores to both observed and novel triples. We defined entailment scores between relations based on a KG containing both. Our empirical evaluation shows that the resulting entailment graph is stronger than one built on observed triples alone. We have also shown that the learned EGs further improve the contextual link prediction task.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5188–5197, Hong Kong, China.

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, Ido Dagan, and Jacob Goldberger. 2010. Global Learning of Focused Entailment Graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden.

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.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can We Predict New Facts with Open Knowledge Graph Embeddings? A Benchmark for Open Link Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, Online.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388, Hong Kong, China. Association for Computational Linguistics.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. Bertese: Learning to speak to bert. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623.

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.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of Link Prediction and Entailment Graph Induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 4284–4295, Montreal, Canada.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia–a Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

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.

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.

Dekang Lin and Patrick Pantel. 2001. Discovery of Inference Rules for Question-Answering. *Natural Language Engineering*, pages 343–360.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China.

Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. 2010. Learning First-Order Horn Clauses From Web Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, Massachusetts, USA.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

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.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling Web-based Acquisition of Entailment Relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Barcelona, Spain.

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.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, pages arXiv–1910.

Liang Yao, Chengsheng Mao, and Y. Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 2901–2908, New York City, New York, USA.

Alexander Yates and Oren Etzioni. 2009. Unsupervised Methods for Determining Object and Relation Synonyms on the Web. *Journal of Artificial Intelligence Research*, 34:255–296.

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.

A Entailment Graph Precision-Recall Curves

Figure 4 shows the precision-recall curves of evaluating the EGs on the Levy/Holt's dataset in (A) local and (B) global settings. We have not shown the MC model for more clarity.

B Evaluating Directionality of Entailment Graphs

We evaluate all models on the directional portion of the Levy/Holt's dataset. This portion is a subset of the main dataset and contains 2414 examples (630 in dev and 1784 in test). For any triple pair in this portion, the reverse of the pair is also present. The entailment is correct in one direction and incorrect in the other. For example, *Printing press was invented by Gutenberg* entails *Gutenberg developed the printing press*; however, the entailment is not correct in the opposite direction.[12] This makes the task much harder than the one of the original dataset. Even a perfect paraphrasing model (two-way entailment) gets the precision of exactly 0.50. Therefore, the model needs to specifically score the entailment in one direction above the other direction. For the original dataset, a symmetric score such as Lin score (Lin, 1998), that is only aware of relatedness between relations but cannot distinguish the directions, can still solve many examples correctly and yield high precision values (Hosseini et al., 2018).

Figure 5 shows the precision-recall curves for global models. We report the area under the curves in Table 3 for recall ≤ 0.33 that are covered by all models. In order to have a fair comparison between the Aug Contextual MC and the Aug MC models, we also computed the area under the curve for recall ≤ 0.48 that is covered by both models: the area under the curves are 0.251 and 0.250, respectively. The results show that defining the entailment scores on a Markov chain as the probability that a path (of length 2) from one relation ends in another relation is an effective way to predict directional entailments. Augmenting the Markov chains with additional links further improves the results. Note that while the two models with augmentation get better overall results, the precisions for all models are still relatively low (≤ 0.60). In addition, the precision is not high even for low recalls meaning

[12] During the data annotation, one of the arguments is masked with its type so that world knowledge does not bias the data (Levy and Dagan, 2016).

BInc	.155
MC	.159
Aug MC	.163
Context MC	.159
Aug Context MC	**.165**

Table 4: Area under the precision-recall curve on the directional subset of the Levy/Holt's dataset (recall $\leq$0.33).

that the models cannot separate the directionality of the entailments well even if the entailment scores are very high. This calls for more research on finding the direction of the relational entailments.

C Hyperparameter Details

We tuned the hyperparameters using grid search or manually as specified below. We tuned the hyperparameters for training (Section 5.2) as well as evaluating contextual link prediction (Section 6.1) using the MAP of the NewsSpike development portion. We tuned the hyperparameters of the inference (Section 5.3) on the Levy/Holt's development dataset.

The hyper-parameters for training are tuned as:

- **Initial learning rate for contextualized embeddings**: 10^{-6} selected from $\{10^{-4}, \ldots, 10^{-8}\}$
- **Initial learning rate for out-of-context embeddings**: 10^{-4} selected from $\{10^{-2}, \ldots, 10^{-6}\}$
- **Batch size**: 64 which was the highest possible size.
- **Number of training epochs**: We used 10. The results stayed similar after 3 epochs. Training takes around 5 days to complete.
- **Number of context tokens**: 40 tokens (up to 20 tokens at each side of the relations). Small windows (e.g., 4 tokens) yielded worse results and more tokens were not feasible. The results were not very sensitive to the number of tokens.

The hyper-parameters for evaluating contextual link prediction are tuned as:

- β: We tuned $\beta = 0.95$ from $\{0.3, 0.5, 0.7, 0.9, 0.93, 0.95, 0.97, 0.99\}$.

The hyper-parameters for inference are tuned as:

- K **(number of connections for relation as specified in Section 5.3)**: We used $K = 100$. $K = 40$ had worse results and $K = 300$ yielded similar results, but larger graphs.
- **Batch size**: The new relations for adding to the Markov chain (i.e., data augmentation) are selected from a candidate set containing relations in the current batch at the inference time. We used a batch size of 512 triple mentions at the inference time that gives us a relatively high number of candidate relations. 512 was the highest possible size to give reasonable entailment graph building time (around 10 days). Smaller batch sizes (256 and 128) yielded slightly worse results.
- α: In the augmented contextual MC model, we multiplied the contextual link prediction scores of the new connected relations by a factor $\alpha \in [0, 1]$ before computing the chain probabilities and the entailment scores. This guides the entailment scores to rely more on the original connections and is useful to improve the precision of the graphs. We tuned $\alpha = 0.5$ based on the development set of the Levy/Holt's entailment dataset. We tuned $\alpha = 0.5$ selected from $\alpha = \{0.3, 0.5, 0.7\}$

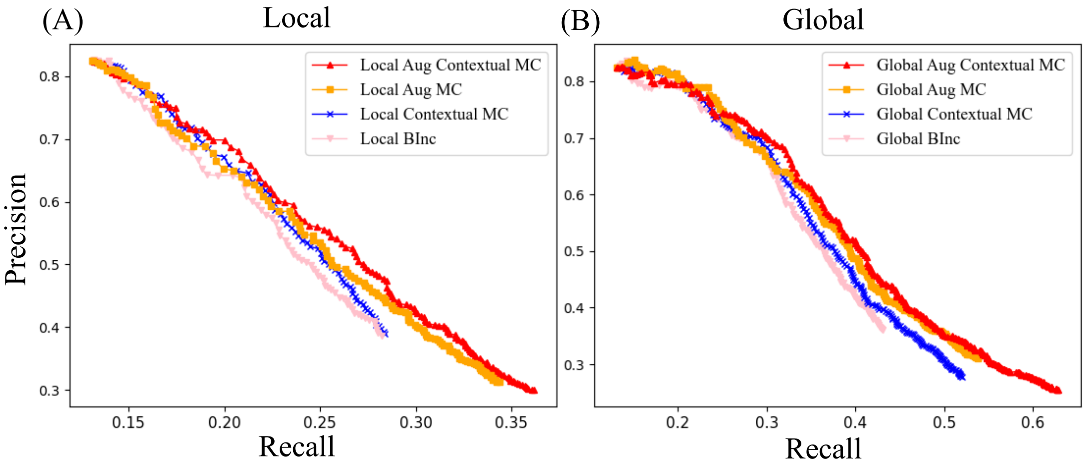


Figure 4: Comparison of the Aug Context MC and Context MC models with models without any context (BInc and Aug MC) on the Levy/Holt's dataset in (A) local and (B) global settings.

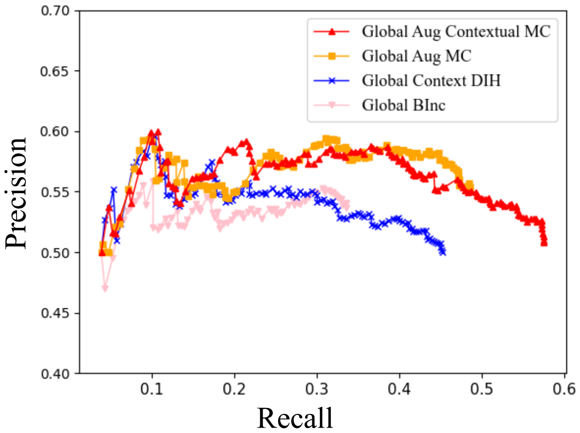


Figure 5: Comparison of models on the directional portion of the Levy/Holt's dataset.