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

Mohammad Javad Hosseini[†§*] **Shay B. Cohen**[†] **Mark Johnson**[‡] **Mark Steedman**[†]
[†]University of Edinburgh [§]The Alan Turing Institute, UK [‡]Oracle Digital Assistant
javad.hosseini@ed.ac.uk, scohen@inf.ed.ac.uk
mark.mj.johnson@oracle.com, steedman@inf.ed.ac.uk

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 that each triple was extracted from. 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 *context-independent* entailments between relations, in the form of entailment graphs (EG), in which the nodes are the relations. The reverse holds too: context-independent 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 known set of hand-built relations. In contrast, the relation-labels of *open-domain* KGs are obtained from text rather than fixed. Open-domain KGs can be constructed by applying parsers or open-information extraction methods to text (Hosseini et al., 2019; Broscheit et al., 2020).

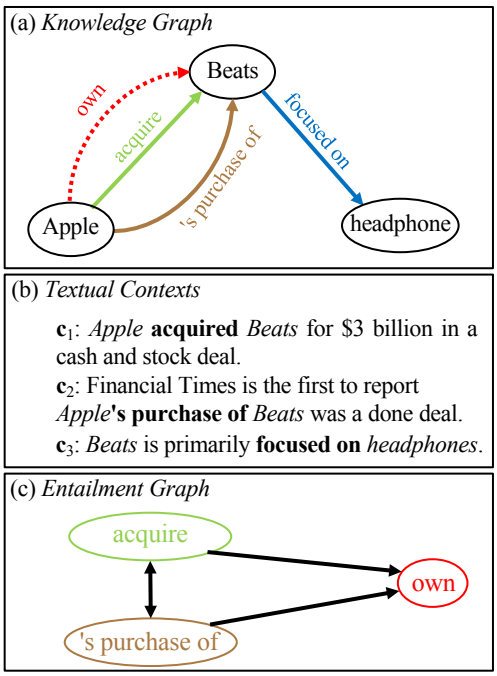

Figure 1: a) Part of an example KG. The relation *own* is missing, but can be predicted from the rest of the KG and the triple contexts using contextual link prediction. b) The contexts $\mathbf{c}_1$ and $\mathbf{c}_2$ from which we have extracted the KG triples. The relation tokens are **bold-faced** and entities are *italic*. 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 *Organization, Organization*. The contextual link prediction and EG learning tasks are complementary. For example, *acquire* → *own* from the EG can independently be used to add the 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.

[*] Now at Google Research.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2790–2802
November 7–11, 2021.

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 *context-independent* entailments 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; Hosseini, 2021). The type of each relation is determined by the types of its two entities. EGs are by definition context-independent, but they use relation types as a proxy of the context. 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 relations 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 the 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 relation entailment score that uses the extracted triples as well as predicted ones from contextual link prediction. We build state-of-the-art EGs.
- We show that EGs in turn improve contextual link prediction.
- We release a dataset containing the extracted triples grounded in context, for future research.

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

2 Related Work

Relational Entailment Graphs. Earlier attempts take a *local* approach and predict entailment relations independently from each other (Lin and Pantel, 2001; Szpektor and Dagan, 2008). Berant et al. (2011, 2015) and Hosseini et al. (2018) propose a *global* approach where the dependencies between the entailment relations 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 such as the transitivity of entailment. 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. Hosseini et al. (2019) perform standard link prediction to add more coverage to the EGs by adding missing relations before building the graphs. We instead perform contextual link prediction prior to building the graphs.

McKenna et al. (2021) extend EGs to include entailments of different valencies, e.g., a binary relation entailing a unary one. We only consider binary relations, but the method can be extended to multiple valencies in the future. Guillou et al. (2020) use temporal information for entailment graph learning by constraining the context of each relation to entity-pairs observed in a temporal window around it. Our method could be adapted similarly. Schmitt and Schütze (2021) have proposed a supervised model that fine-tunes pre-trained LMs directly on a training portion of entailment datasets. They report better results than EGs, but our focus is different. Unlike their method, our approach is unsupervised and is not capable of learning potential artifacts from datasets (Levy et al., 2015). In addition, we explicitly build EGs by doing machine-reading over large text corpora, and hence can explain the basis for the beliefs captured in them.

Pre-trained LMs for Link Prediction. KG-BERT (Yao et al., 2019) uses contextual representations for KG completion. However, they form synthetic token sequences by concatenating entity descriptions and relation tokens, whereas we use the natural text associated with the triples.

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 because a) They face technical challenges in processing multi-token relations; and b) Relations can be expressed in many different ways. 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 hypothesis relations (e.g., *own*), while MTB learns a symmetric similarity score. Second, we can predict a score for any hypothesis relation as long as the relation is previously observed somewhere in the corpus with any other entities (§3.2).

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*), $\mathcal{T}$ denote the set of all entity types (e.g., *Person*; *Thing*) and $\mathcal{R}$ denote the set of all typed relations extracted from a text corpus. We consider binary relations where each relation has two entities, and hence two types, 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)*, *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 (extracted) triple. For example, $\mathcal{R}$*(Barack Obama, Hawaii)* includes *born in*[3], *visit*, etc.

Link prediction and entailment can hold between relations with the same entity order or the reverse order. When the two entity types are identical, we keep two copies of the relations one for each entity order. For example, *acquire(*Org_1*,*Org_2*)* predicts *be part of (*Org_2*,*Org_1*)*. We specify the entity order of a relation by a binary flag $o(r)$. For relations with unequal types, we do not need the flag as the order is obvious and 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.

A triple mention is a triple grounded in its textual context. We define a triple mention as a tu-

[3] For brevity, we drop the types when they are obvious.

ple $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$, 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,*$\mathbf{c}_2$*,*$[9, 11]$*)*. We denote by $\mathcal{D} = \left[(e_{i,1}, r_i, e_{i,2}, \mathbf{c}_i, \mathbf{s}_i)\right]_{i \in \{1,\ldots,N\}}$ the set of all triple mentions.

We define the contextual link prediction task as follows: Given a triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$ as a premise, the goal is to predict for any hypothesis relation $q \in \mathcal{R}(t_1, t_2)$ whether it holds between the entity-pair (e_1, e_2) or not, 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 a hypothesis relation q holds between the entity-pair (e_1, e_2) conditioned on the triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$. We propose a model, named Contextualized and Non-Contextualized Embeddings (CNCE), which is based on two different embedding spaces for relations (Figure 2):

First, the premise relation r in the triple mention m has a contextualized embedding encoding the meaning of the relation in its textual context. The contextualized embedding is 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., $\vec{m} = (\vec{h}_{s_1} + \vec{h}_{s_2})/2$. 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., $o(r) = 1$) (§3.1).

Second, each hypothesis relation $q \in \mathcal{R}$ such as *own (Organization*$_1$*,Organization*$_2$*)* has a non-contextualized embedding encoding its general context-independent meaning. While the contextualized embeddings of relations can vary depending on their textual contexts, each relation has exactly a single non-contextualized embedding. The non-contextualized embedding is taken from an embedding weight matrix that is learned from scratch from the KG (§3.3). We use the non-contextualized embedding $\vec{q} \in \mathbb{R}^d$ to encode the hypothesis 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:

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

Eq 1 estimates the probability that the relation q holds between the entity-pair. It can be applied to any relation $q \in \mathcal{R}(t_1, t_2)$[5] and predict that multiple relations are compatible with the context. Table 1 shows an example from the text corpus and the predictions by our actual trained model.

We encode the hypothesis relations with learned non-contextualized embeddings rather than contextualized embeddings for two reasons: a) The model can be applied to relations $q \in \mathcal{R}(t_1, t_2) \backslash \mathcal{R}(e_1, e_2)$ where the triple (e_1, q, e_2) is unobserved. This is useful to find novel triples that are correct, but are not found in the text. *If we were modeling the hypothesis relation q with contextualized embeddings, the relation should have also been observed with the same entity-pair* (e_1, e_2) *somewhere else in the corpus, hence* (e_1, q, e_2) *would not be a novel triple.*[6] b) While the score uses the dot product between embedding vectors, it is still *asymmetric* as

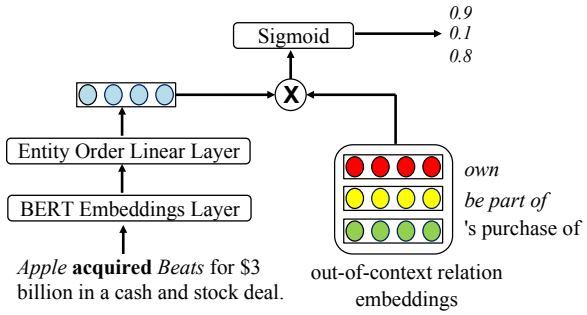

Figure 2: An example of contextual link prediction. The relation token is **boldfaced** and entities are *italic*. The output probabilities correspond to the input context-independent relations.

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

[5] We do not compute contextual link prediction score for relations with different entity types as they are irrelevant to the triple mention.

[6] One possibility would be a purely textual baseline that predicts a score for sentence-pairs like *"Apple acquires Beats for ... [SEP] Apple owns Beats."*, where [SEP] is a special token separating an actual sentence observed in the corpus and a synthesized one containing the triple. But that involves computing contextualized embeddings for every context and relation pair, instead of performing the computation for ev-

Underwood arrived **sporting** the 381 carat diamond *necklace*, with her blonde hair worn loose around her shoulders.

RELATION	CNCE SCORE	EXTRACTED
sport	1.0	1
wear	0.994	1
pose with	0.715	0
possess	0.600	0
experiment with	0.385	0
's	0.214	1

Table 1: A triple mention and its top predictions by our actual trained method. The extracted triple is *(Carrie Underwood, sport, necklace)*. Labels are 1 if the triple has been extracted from another context in the text corpus, and 0, otherwise. The relation token is **boldfaced** and entities are *italic*. Some high-scoring relations with the label 0 are likely to be correct and are useful for augmenting the set of extractions (§4). Many wrong relations get lower scores (not shown here).

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 Figure 1, we should predict *own* given *acquire*, but not *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: a) high scores to relations q that hold between the entity-pairs, i.e., $q \in \mathcal{R}(e_1, e_2)$ or equivalently (e_1, q, e_2) is a valid (extracted) triple. b) low scores to relations q' that do not hold between the entity-pairs, i.e., $q' \in \mathcal{R}(t_1, t_2) \setminus \mathcal{R}(e_1, e_2)$, or equivalently (e_1, q', e_2) is an invalid (not extracted) triple. This can be seen as a multi-label classification task. Each triple mention is an example with a total number of $|\mathcal{R}(t_1, t_2)|$ binary labels, i.e., the number of relations with types t_1 and t_2. The labels are relations that we wish to predict whether they hold between the entity-pair (positive) or not (negative). For example in Table 1, there is a binary label for any of the relations *sport*, *wear*, *pose with*, etc, given the triple mention. 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 non-contextualized 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 CNCE Markov chain (MC), similar to the ConvE MC score of Hosseini et al. (2019), but based on the contextual link prediction score, as follows.

We form a bipartite graph with relations on one side and triple mentions on the other side (Figure 3).[7] 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 $\mathcal{R}(m) = \mathcal{R}(e_1, e_2) \cup \mathcal{U}(m)$, where $\mathcal{R}(e_1, e_2)$ is the set of observed relations for the entity-pair and $\mathcal{U}(m) \subseteq \mathcal{R}(t_1, t_2) \setminus \mathcal{R}(e_1, e_2)$ contains a set of relations with high contextual link prediction scores. For a relation $u \in \mathcal{U}(m)$, the triple (e_1, u, e_2) is unobserved in the text corpus, but is likely to be correct (e.g., *pose with* in Table 1). We augment the Markov chain with such connections from mentions m to relations u. Figure 3 shows an example Markov chain, where dotted links correspond to novel triples proposed by contextual link prediction. We define the transition probabilities from relations to mentions uniformly, and from mentions to relations as normalized contextual link prediction scores:

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

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

ery context as in our method. Therefore, it does not scale to the data size for open-domain contextual link prediction (§5.2). In addition, this baseline does not take advantage of the natural textual contexts of the hypothesis relations.

[7] Previous work forms a bipartite graph with relations on one side and *entity-pairs* on the other side.

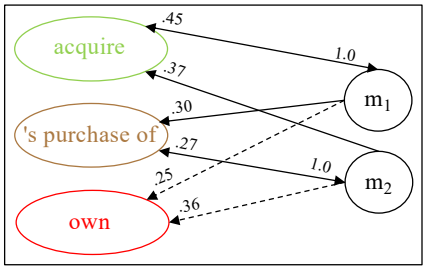

Figure 3: An example Markov chain. It has the relations *acquire*, *'s purchase of*, and *own* (r_1, r_2, and r_3) on the left, and the triple mentions $m_1 = (e_1, r_1, e_2, \mathbf{c}_1, \mathbf{s}_1)$ and $m_2 = (e_1, r_2, e_2, \mathbf{c}_2, \mathbf{s}_2)$ on the right, where e_1 and e_2 are *Apple* and *Beats*, $\mathbf{c}_1$ and $\mathbf{c}_2$ are textual 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 found by contextual link prediction that help adding the edge *acquire* entails *own* to the EG.

where $\Pr(q|m)$ is defined in Eq 1. We define the CNCE MC entailment score as:

$$w_{rq} = \Pr(\langle q \rangle | \langle r \rangle) = \sum_{m \in \mathcal{D}(r)} \Pr(\langle q \rangle | \langle m \rangle) \Pr(\langle m \rangle | \langle r \rangle). \quad (3)$$

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 $|\mathcal{R}(e_1, e_2)|$ is less than K=100[8], we augment the Markov chain by connecting the mentions to $K - |\mathcal{R}(e_1, e_2)|$ highest scoring relations (i.e., $\mathcal{U}(m)$) corresponding to unobserved triples.

We then use entailment scores as the local entailment scores and apply the global soft constraints of (Hosseini et al., 2018) to learn global entailment scores. We build EGs, whether local or global, by applying a threshold on the entailment scores,. For a threshold $\delta > 0$, we build EGs where nodes are relations $r \in \mathcal{R}$, and edges include (r, q) with $w_{rq} \geq \delta$.[9] In our experiments, we slide the threshold in the range $[0, 1]$ to build and evaluate EGs with varying degrees of confidence.

5 Experimental Setup

We discuss the details of the corpus and training.

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). We use the event-relation extraction pipeline of Hosseini et al. (2018) to extract triple mentions. They process the corpus with GraphParser (Reddy et al., 2014), a Combinatory Categorial Grammar (CCG; Steedman, 2000) semantic parser. GraphParser uses EasyCCG (Lewis and Steedman, 2014) to extract CCG dependencies for a sentence, constructs a dependency graph, and traverses the graph from each event node[10] to an entity leaf node[11]. The predicate string is formed by concatenating the words in the traversed path. The triples are normalized by lemmatizing the words. We analyze the accuracy of the parser in Appendix B.

The parser outputs triples in addition to the indices of relation tokens. We re-parse the corpus and record each triple coupled with its context and the indices of its relation tokens. We assign types to the entities following (Hosseini et al., 2018). They link each entity to Freebase using the AIDA-light (Nguyen et al., 2014) entity linker; select the most notable entity type from Freebase; and automatically map it to FIGER types (Ling and Weld, 2012). FIGER types have at most two levels of hierarchies, e.g., *person/author*. We use the 49 first-level FIGER types, e.g., *person*.

This process 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 initialized the contextualized embeddings with BERT-base.[12] We used Adam (Kingma and Ba, 2015) with linear decay of learning rates to minimize the loss function defined in Eq 2. 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 contextual link prediction since if we simply split randomly, identical triples (e_1, r, e_2) might exist in training, development, and test sets in different contexts.

[8] K is tuned on an entailment dev set (Appendix A).

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

[10] Verb or preposition identified by the POS tags VB*, IN, TO, POS.

[11] Noun, proper noun, or pronoun.

[12] 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.

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 (§3.3) each triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$ is considered as an example with $|\mathcal{R}(t_1, t_2)|$ 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, which leaves almost all of the relations as negative.

To resolve this problem, we train on all positive relations (first term of Eq 2 unchanged), but choose a small subset of relations as candidate negatives (second term of Eq 2 applied to fewer relations): a) For each triple mention m, we use the positive relations from other triple mentions in the mini-batch as negative relations, if those are not already among the positive relations of m. For example, while processing the triple mention in Table 1, if another triple mention in the mini-batch has the positive relations *makes* and *propose*, they will be used as negative relations for the triple mention. b) We also choose a random subset of all the relations with the same entity types as negative candidates for the whole mini-batch. This random subset has the size of (up to) the positive relations of the whole mini-batch. The training data consists of 8.1M triple mentions. It has a total of 435M positive labels and a total of 7128M negative labels. This corresponds to around 54 positive labels (880 negative labels) per triple mention on average.

We tuned hyperparameters by maximizing the mean average precision (MAP) of contextual link prediction in the development set. To compute the MAP, we rank the predictions of each dev/test triple mention from highest to lowest. We compute average precision by computing the precision value of one relation at a time. For the example in Table 1, the average precision is: $(1/1+2/2+3/6+\ldots)/n$, where n is the number of relations that hold for *(Carrie Underwood, necklace)*. We then compute the mean of average precisions.

For each contextualized embedding $\vec{m}$, we have many (typically $\geq$100) non-contextualized embeddings. Therefore, each contextualized embedding will be usually updated $\geq$100 times the number of updates for non-contextualized embeddings. In practice, this causes the contextualized embeddings to be significantly affected by the randomly initialized non-contextualized embeddings. In order to resolve this issue, we tuned initial learning rates separately: 10^{-6} for contextualized embeddings, and 10^{-4} for non-contextualized embeddings and the matrices A_0 and A_1. We found that 40 tokens are sufficient for the context, where the context can cross multiple sentences. We optimized the model for 10 epochs. We discuss the hyper-parameter range and tuning details in Appendix A.

ConvE	.230
TuckER	.263
CNCE	**.333**
ABLATION STUDIES	
CNCE w/o Entity Order Flag	.319
CNCE w/o BERT Layers Update	.321

Table 2: MAP of relation prediction given triple mentions evaluated on the NewsSpike test set.

6 Results and Discussion

We first evaluate our proposed method for the open-domain contextual link prediction task (§6.1). We then investigate the complementarity of contextual link prediction and EGs (§6.2 and §6.3).

6.1 Evaluating Contextual Link Prediction

We evaluate our proposed method, CNCE, against standard link prediction baselines. We compute the MAP of predicting hypothesis 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.[13] 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.

CNCE is our novel model that calculates $\Pr(q|m)$ (Eq 1) as the contextual link prediction score. We used two standard non-contextual link prediction methods, **ConvE** (Dettmers et al., 2018) and **TuckER** (Balazevic et al., 2019), which are among the state-of-the-art link prediction methods.

Table 2 shows the results. CNCE outperforms the 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.

[13]Without this assumption, the results of all models drop.

We perform ablation of CNCE. We test the model without the entity order flag, i.e., $o(r) = 0$ for all relations. In addition, we freeze the BERT layers, which means just using BERT as a feature extractor. In both cases, the MAP decreases.

6.2 Contextual Link Prediction Assists EGs

We compare the EGs obtained by our proposed entailment score (§4) with previous state-of-the-art EGs on the Levy/Holt's entailment dataset (Levy and Dagan, 2016; Holt, 2018). For the experiments of this section, we compute the entailment scores using all the NewsSpike triple mentions, not only the training, since we evaluate on an external entailment dataset. 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. **CNCE MC** is our novel entailment score (Eq 3) that uses CNCE to compute transition probabilities in the Markov chain. **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 side (as opposed to our Markov chain with triple mentions on the second side). The transition probabilities are computed by a standard link prediction method such as ConvE or TuckER.[14] Balanced Inclusion (**BInc**) is a Sparse Bag-of-Word model (Szpektor and Dagan, 2008) used by Hosseini et al. (2018).

For the link prediction methods we report results in two settings. **No Aug** means that we only use the triples extracted from the text-corpus and do not augment them with novel triples from standard or contextual link prediction. In this setting, the link prediction is only used to compute transition probabilities in the Markov chains. **Aug** means that we add the novel triples (the dotted links in Figure 3) before computing the entailment scores.

	LOCAL		GLOBAL	
	NO AUG	AUG	NO AUG	AUG
BInc	.076	-	.165	-
ConvE MC	.079	.085	.174	.187
TuckER MC	.071	.082	.162	.184
CNCE MC	**.084**	**.096**	**.176**	**.195**

Table 3: Area under the precision-recall curves of EGs on the Levy/Holt's dataset (precision >0.5). We compare our CNCE MC model with BInc (Szpektor and Dagan, 2008; Hosseini et al., 2018), and ConvE/TuckER MC (Hosseini et al., 2019).

Table 3 shows the results of EGs in local and global settings. The plots are shown in Appendix C. Our proposed score outperforms the previous scores across all settings. We can also see that augmenting the triples improves the results for all the link prediction based entailment scores. The augmentation alleviates the sparsity of EGs by adding more connections between the relations (e.g., *acquire* $\rightarrow$ *own* in Figure 3). The comparison between CNCE MC and ConvE/TuckER MC, i.e., the previous state-of-the-art EGs, confirms that contextual link prediction is more effective than standard link prediction in finding new high-quality triples to augment the extracted triples. We also observe similar results on the strictly directional portion of the dataset (Appendix D).

6.3 EGs Assist Contextual Link Prediction

We test whether we can use context-independent EGs to improve the contextual link prediction task. We evaluate the following additional models.

EG is based on CNCE MC scores, without any augmented triples. **Aug EG** is based on the CNCE MC scores, computed with augmented triples.[15] Given a triple mention $m = (e_1, r, e_2, \mathbf{c}, \mathbf{s})$, we use the entailment score w_{rq} to decide whether the triple (e_1, q, e_2) should be added to the KG. Note that while textual contexts have been used to compute the entailment scores w_{rq} (Eq 3), the EG baselines are context-independent: 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

[14] The previous work has only reported results with ConvE, but we also repeated their experiments with TuckER.

[15] We could use any entailment scores, but we tried the best performing ones on the Levy/Holt's dev set.

CNCE	.333
EG	.317
Aug EG	.328
CNCE + EG	.355
CNCE + Aug EG	**.357**

Table 4: MAP of relation prediction given triple mentions evaluated on the NewsSpike test set.

CNCE and the EGs: **CNCE + (Aug) EG**, is the linear summation $\beta \Pr(q|m) + (1-\beta)w_{rq}$, where $\beta \in [0, 1]$ is a hyper-parameter.[16]

Table 4 shows the results. Our contextual CNCE (.333) model outperforms the context-independent EGs. Between the EG baselines, the augmented one (.328) gets better results than the basic one (.317) showing that the augmentation technique is effective, similar to our findings in Section 6.2. Combining CNCE and EGs yields further improvements (.355 and .357) showing that EGs contain complementary information that further strengthen contextual link prediction.

We perform qualitative analysis of the best combined model (CNCE + Aug EG) to check the complementarity of the two approaches. Table 5 (top) shows an example from NewsSpike where EGs perform better than CNCE. 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 CNCE, but the EGs do not contain these wrong predictions. On the other hand, Table 5 (bottom) shows an example where CNCE 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*.[17]. 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, CNCE disambiguates the context. In addition, the scores of some correct relations (e.g., *achieves*) are increased by CNCE.

7 Conclusions

We have introduced the contextual link prediction problem and proposed a model (CNCE) for it. We trained CNCE on a corpus of triple mentions.

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

[17]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.

EGs Improve CNCE		
Triple	Apple, **is working on**, watch	
Predictions	Watch, **falls on**, Apple (0)	21→23
	Apple, **'s**, watch (1)	5→2
	Apple, **has**, watch (1)	12→7
	Apple, **launches**, watch (1)	20→15
	Apple, **tests**, watch (1)	98→41
CNCE Improves EGs		
Triple	Microsoft, **is committed to**, success	
Predictions	Microsoft, **builds**, success (0)	3→5
	Microsoft, **switches to**, success (0)	18→81
	Microsoft, **'s**, success (1)	4→2
	Microsoft, **achieves**, success (1)	108→6
	Microsoft, **hopes for**, success (1)	116→37

Table 5: Extracted triples and example predictions for relations of types (*Organization,Thing*). The true label for each prediction is written in brackets, where (1) means the triple is part of the unseen development set KG, and (0) means otherwise. The ranking of each predicted relation among all relations is written both for the individual model (left-hand side of the arrow) and the combined model (right-hand side of the arrow). Top) Context: *Apple is working on a high-tech watch.* EGs improve CNCE. Bottom) Context: *Microsoft is committed to the long term success of the entire PC ecosystem.* CNCE improves EGs.

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 by using both sets of triples. 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.

Acknowledgements

The authors would like to thank Liane Guillou, Nick McKenna, Sander Bijl de Vroe, and Ian Wood for helpful discussions and the 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 and a University of Edinburgh/Huawei Technologies award (Steedman).

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, 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.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. 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 6262–6267.

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.

Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2020. Incorporating Temporal Information in Entailment Graph Mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71, Barcelona, Spain (Online). 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. 2021. *Unsupervised Learning of Relational Entailment Graphs from Text*. Ph.D. thesis, University of Edinburgh.

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.

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.

William Lechelle, Fabrizio Gotti, and Philippe Langlais. 2019. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15.

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.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976.

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*.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, and Mark Steedman. 2021. Multivalent entailment graphs for question answering. *arXiv preprint arXiv:2104.07846*.

Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-Throughput Named-Entity Disambiguation. In *Workshop on Linked Data on the Web*, pages 1–10, Seoul, Korea.

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.

Martin Schmitt and Hinrich Schütze. 2021. Language Models for Lexical Inference in Context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1267–1280, Online. Association for Computational Linguistics.

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.

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.

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 Hyperparameter Details

We tuned the hyperparameters using grid search or manually as specified below. We tuned the hyperparameters for training (§5.2) as well as evaluating contextual link prediction (§6.1) using the MAP of the NewsSpike development portion. We tuned the hyperparameters of the inference (§4) 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 $\{10^{-4}, 10^{-5}, \ldots, 10^{-8}\}$
- **Initial learning rate for non-contextualized embeddings**: 10^{-4} selected from $\{10^{-2}, 10^{-3}, \ldots, 10^{-6}\}$
- **Mini-batch size**: 64 which was the highest possible size. Also tried 32.
- **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 β=0.05, for combining CNCE and EGs, from $\{0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7\}$.

The hyper-parameters for inference when building the EGs are tuned as:

- K **(number of connections for relation as specified in Section 4)**: We used $K = 100$. $K = 40$ had worse results and $K = 300$ yielded similar results, but larger graphs.
- **Mini-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 mini-batch at the inference time. We used a mini-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 mini-batch sizes (256 and 128) yielded slightly worse results.
- α: In the augmented CNCE 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 $\alpha = \{0.3, 0.5, 0.7, 1.0\}$

B Analyzing the semantic parser output

In this section, we report the result of our analysis of the semantic parser that we used to extract the triple mentions (§5.1). We analyzed 100 randomly extracted triples. The exact tuple match precision is 73.8% and the token-based precision is 87.5%, which is relatively high compared to existing openIE systems (Bhardwaj et al., 2019; Lechelle et al., 2019). Compared to OpenIE, CCG extractions generate better triples for sentences with long-range dependencies as well as those involving coordination.

C 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 ConvE MC model and TuckER (Aug) MC models for more clarity.

D 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

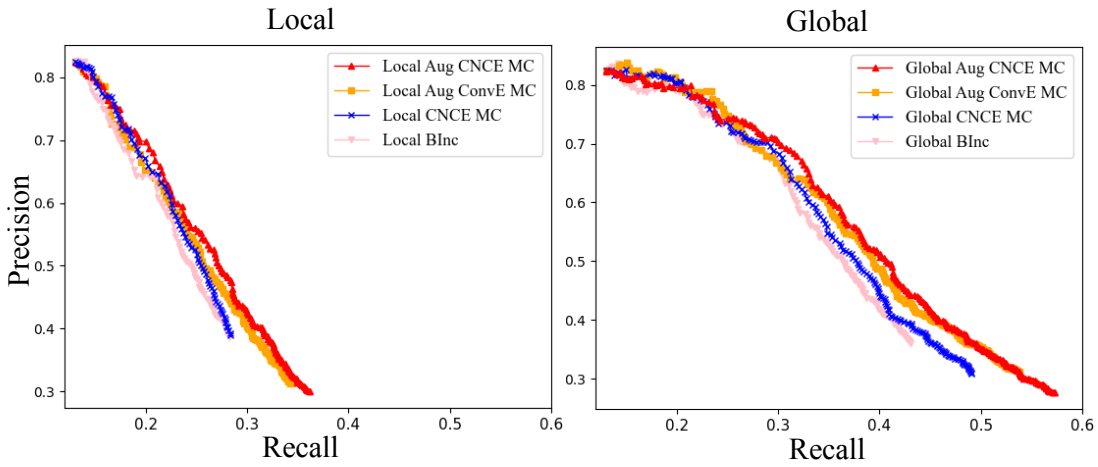

Figure 4: Comparison of the EGs based on the new entailment scores CNCE Aug MC and CNCE MC with previous state-of-the-art EGs on the Levy/Holt's dataset in (A) local and (B) global settings.

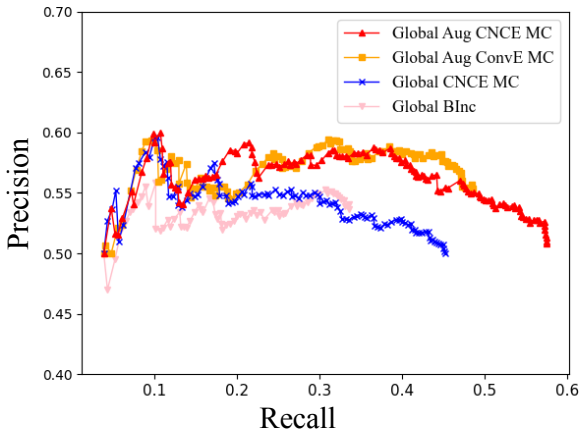

Figure 5: Comparison of models on the directional portion of the Levy/Holt's dataset (global setting).

BInc	.155
ConvE MC	.159
Aug ConvE MC	.163
TuckER MC	.156
Aug TuckER MC	.161
CNCE MC	.159
Aug CNCE MC	**.165**

Table 6: Area under the precision-recall curve on the directional portion of the Levy/Holt's dataset (recall $\leq$0.33) (global setting).

correct in the opposite direction.[18] 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).

We report the area under the curves in the global setting in Table 3 for recall $\leq$ 0.33 that are covered by all models. Figure 5 shows the precision-recall curves for global models. We have not shown the ConvE MC model and TuckER (Aug) MC models for more clarity. In order to have a fair comparison between Aug ConvE MC, Aug TuckER MC, and Aug CNCE MC models, we also computed the area under the curve for recall $\leq$ 0.48 that is covered by the three models with augmented triples: the area under the curves are .250, .246 and .251, 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 a relatively effective way to predict directional entailments. Augmenting the Markov chains with additional links further improves the results. Note that while Aug ConvE MC and Aug CNCE MC models get better overall results, the precisions for all models are still relatively low ($\leq$ 0.60). In addition, the precision is not high even for low recalls meaning 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.

[18]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).