# CoRI: Collective Relation Integration with Data Augmentation for Open Information Extraction

**Zhengbao Jiang**[1]*, **Jialong Han**[2], **Bunyamin Sisman**[2], **Xin Luna Dong**[2]
Language Technologies Institute, Carnegie Mellon University[1]
Amazon[2]
zhengbaj@cs.cmu.edu
{jialongh,bunyamis,lunadong}@amazon.com

## Abstract

Integrating extracted knowledge from the Web to knowledge graphs (KGs) can facilitate tasks like question answering. We study relation integration that aims to align free-text relations in subject-relation-object extractions to relations in a target KG. To address the challenge that free-text relations are ambiguous, previous methods exploit neighbor entities and relations for additional context. However, the predictions are made independently, which can be mutually inconsistent. We propose a two-stage **Co**llective **R**elation **I**ntegration (CoRI) model, where the first stage independently makes candidate predictions, and the second stage employs a collective model that accesses all candidate predictions to make globally coherent predictions. We further improve the collective model with augmented data from the portion of the target KG that is otherwise unused. Experiment results on two datasets show that CoRI can significantly outperform the baselines, improving AUC from .677 to .748 and from .716 to .780, respectively.

## 1 Introduction

With its large volume, the Web has been a major resource for knowledge extraction. Open information extraction (open IE; Sekine 2006; Banko et al. 2007) is a prominent approach that harvests subject-relation-object *extractions* in free text without assuming a predefined set of relations. One way to empower downstream applications like question answering is to integrate those free-text extractions into a knowledge graph (KG), *e.g.,* Freebase. *Relation integration* is the first step to integrate those extractions, where their free-text relations (*i.e., source relations*) are normalized to relations in the target KG (*i.e., target relations*). Only after relation integration can entity linking proceed to resolve the free-text subjects and objects to their canonical entities in the target KG.

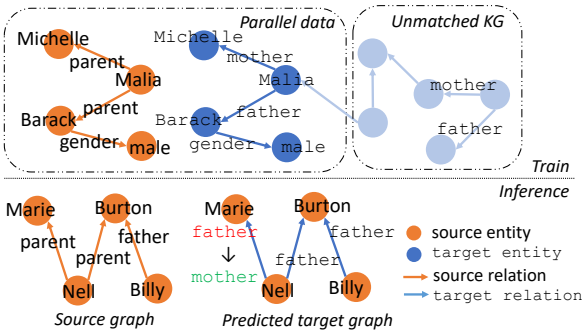



Figure 1: A motivating example. Trained on *parallel data*, a local model may suffer from sparse context for a new entity pair Nell-Marie at inference, wrongly disambiguating "parent" to `father` instead of `mother`.

**Local Approaches.** Relation integration has been studied by the natural language processing (NLP) community. With exact matching in literal form between entity names in the source graph and target KG, previous methods obtain *parallel data*, *i.e.,* common entity pairs, between the two graphs as in Fig. 1. Features of the entity pairs (*e.g.,* Malia-Barack) in the source graph and their relations in the target KG (*e.g.,* `father`) are used to train models to predict target relations for future extractions. A common challenge is the *ambiguity* of source relations, *e.g.,* "parent" may correspond to `father` or `mother` in different contexts. Previous methods exploited *contextual* features including embeddings of seen entities (*e.g.,* "Malia"; Riedel et al. 2013), middle relations between (*e.g.,* "parent"; Riedel et al. 2013; Toutanova et al. 2015; Verga et al. 2017, 2016; Weston et al. 2013), and neighbor relations around the entity pair (*e.g.,* "gender"; Zhang et al. 2019).

Assuming rich contexts to address the ambiguity challenge, previous methods may fall short under the evolving and incomplete nature of the source

*This work was performed while at Amazon.

| Methods | Middle relation | No entity param. | Neighbor relation | **Collective inference** |
|---|---|---|---|---|
| (Riedel et al., 2013) | ✓ | | | |
| (Verga et al., 2017) | ✓ | ✓ | | |
| (Zhang et al., 2019) | ✓ | ✓ | ✓ | |
| CoRI (ours) | ✓ | ✓ | ✓ | ✓ |

Table 1: Comparisons between CoRI and baselines.

graph. For example, in the lower part of Fig. 1, emerging entities may come from new extractions with sparse contextual information. For the pair Nell-Marie, a conventional model learned on the parallel data may have neither seen entities nor neighborhood information (*e.g.,* "gender") to depend on, thus failing to disambiguate "parent" and wrongly predicting `father`. Due to the *local* nature of previous approaches, *i.e.,* predictions for different entity pairs are made independently of each other, the model is unaware that "Nell" has two fathers in the final predictions. Such predictions are incoherent in common sense that a person is more likely to have one father and one mother, which is indicated by the graph structure around `Malia` in the target KG part of the parallel data.

### 1.1 Our Collective Approach

To alleviate the incoherent prediction issue of local approaches, we propose **Co**llective **R**elation **I**ntegration (CoRI) that exploits the dependency of predictions between adjacent entity pairs to enforce global coherence.

Specifically, we follow two stages, *i.e., candidate generation* and *collective inference*. In candidate generation, we simply use a local model to make independent predictions as candidates, *e.g.,* `father` for all the three pairs in the lower part of Fig. 1. In collective inference, we employ a *collective* model that is aware of the common substructures of the target graph, *e.g.,* `Malia`. The collective model makes predictions by not only taking as input all contextual features to the local model but also the candidate predictions of the current and all neighbor pairs. For the pair Nell-Marie, the collective model will have access to the candidate prediction `father` of Nell-Burton, which helps flip its final prediction to the correct `mother`. Tab. 1 summarizes CoRI and representative previous work from four aspects. To the best of our knowledge, CoRI is the first to collectively perform relation integration rather than locally.

Being responsible to make globally consistent predictions, the collective model needs to be trained to encode common structures of the target KG, *e.g.,* `Malia` having only one father/mother in the parallel data of Fig. 1. To this end, we train the collective model in a *stacked* manner (Wolpert, 1992). We first train the first-stage local model on the parallel data, then train the second-stage collective model by conditioning on the candidate predictions of neighbor entity pairs from the first stage (*e.g.,* `father` for Malia-Barrack) to make globally consistent predictions (*e.g.,* `mother` for Malia-Michelle).

**Parallel Data Augmentation.** The parallel data may be bounded by the low recall of exact name matching or the limited extractions generated by open IE systems. We observe that, even without counterpart extractions, the *unmatched* part of the target graph (as in Fig. 1) may also have rich common structures to guide the training of the collective model. To this end, we propose *augmenting* the parallel data by sampling subgraphs from the unmatched KG and creating pseudo parallel data by synthesizing their extractions, so the collective model can benefit from additional training data characterizing the desired global coherence.

To summarize, our contributions are three-fold: **(1)** We propose CoRI, a two-stage framework that improves state-of-the-art methods by making collective predictions with global coherence. **(2)** We propose using the unmatched target KG to augment the training data. **(3)** Experimental results on two datasets demonstrate the superiority of our approaches, improving AUC from .677 to .748 and from .716 to .780, respectively.

## 2 Preliminaries

In this section, we first formulate the task of relation integration, then describe local methods by exemplifying with the state-of-the-art approach OpenKI (Zhang et al., 2019).

### 2.1 Relation Integration

We treat subject-relation-object extractions from open IE systems as a *source graph* $\mathcal{K}(\mathcal{E}, \mathcal{R}) = \{(s, r, o) \mid s, o \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ denotes extracted textual entities, *e.g.,* "Barack Obama", and $\mathcal{R}$ denotes extracted source relations, *e.g.,* "parent". We denote by $(s, o)$ a source entity pair. For $(s, o)$, $\mathcal{K}_{s,o} = \{r \mid (s, r, o) \in \mathcal{K}\}$ denotes all source relations between them. Similarly, $\mathcal{K}_r = \{s, o \mid (s, r, o) \in \mathcal{K}\}$ denotes all entity pairs with relation

$r$ in between. We use the union $\mathcal{K}_{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \mathcal{K}_r$ to refer to all extracted entity pairs.

**Definition 1 (Relation Integration).** Given a source graph $\mathcal{K}$ and a *target KG* $\mathcal{K}'(\mathcal{E}', \mathcal{R}')$ with target entities $\mathcal{E}'$ and target relations $\mathcal{R}'$, the task of relation integration is to predict all applicable target relations for each extracted entity pair in $\mathcal{K}_{\mathcal{R}}$:

$$\Gamma \subseteq \mathcal{K}_{\mathcal{R}} \times \mathcal{R}',$$

where $(s, r', o) \in \Gamma$ is an *integrated extraction* indicating that a target relation $r'$ holds for $(s, o)$.

To train relation integration models, all methods employ *parallel data* formalized as follow:

**Definition 2 (Parallel Data).** Parallel data are common entity pairs shared between $\mathcal{K}_{\mathcal{R}}$ and $\mathcal{K}'_{\mathcal{R}'}$ and their ground truth target relations in $\mathcal{K}'$: $\mathcal{T} = \{\langle (s, o), \mathcal{K}'_{s,o} \rangle \mid (s, o) \in \mathcal{K}_{\mathcal{R}} \cap \mathcal{K}'_{\mathcal{R}'}\}$. For example, $\langle (\text{Malia}, \text{Barack}), \{\texttt{father}\} \rangle$ is an instance of parallel data in Fig. 1.

To obtain parallel data, a widely used approach is to find entities shared by $\mathcal{E}$ and $\mathcal{E}'$ by exact name matching, then generate common entity pairs and their ground truth.

### 2.2 Local Approaches

Previous local methods score potential integrated extractions by assuming their independence:

$$P(\Gamma \mid \mathcal{K}) = \prod_{(s,r',o) \in \mathcal{K}_{\mathcal{R}} \times \mathcal{R}'} P_\theta(s, r', o \mid \mathcal{K}), \quad (1)$$

where $\theta$ is the parameters of the local model. One representative local model achieving state-of-the-art performance is OpenKI (Zhang et al., 2019). It encodes the neighborhood of $(s, o)$ in $\mathcal{K}$ by grouping and averaging embeddings of source relations in three parts. Let $\mathcal{K}_{s,\cdot}$ be the set of source relations between $s$ and neighbor entities other than $o$, and similarly for $\mathcal{K}_{\cdot,o}$. OpenKI represents $(s, o)$ by concatenating the three averaged embeddings into a *local representation* $\mathbf{t}_l$:

$$\mathbf{t}_l = [A(\mathcal{K}_{s,o}); A(\mathcal{K}_{s,\cdot}); A(\mathcal{K}_{\cdot,o})], \quad (2)$$

where $l$ stands for local, and $A(.)$ takes a set of relations and outputs the average of their embeddings. Then each integrated extraction is scored by:

$$P_\theta(s, r', o \mid \mathcal{K}) = \sigma(\mathbf{MLP}_l(\mathbf{t}_l))_{r'}, \quad (3)$$

where $\mathbf{MLP}_l$ is a multi-layer perceptron and $\sigma$ the sigmoid function. Given a parallel data $\mathcal{T} = \{\langle (s, o), \mathcal{K}'_{s,o} \rangle\}$, the loss function per training example trades between maximizing the probabilities of positive target relations and minimizing those of negative target relations:

$$L\big((s, o), \mathcal{K}'_{s,o}\big) = -\frac{1}{|\mathcal{K}'_{s,o}|} \sum_{r' \in \mathcal{K}'_{s,o}} \log P_\theta(s, r', o \mid \mathcal{K}) + \frac{\gamma}{|\mathcal{R}' \setminus \mathcal{K}'_{s,o}|} \sum_{r' \in \mathcal{R}' \setminus \mathcal{K}'_{s,o}} \log P_\theta(s, r', o \mid \mathcal{K}), \quad (4)$$

where $\gamma$ is a hyperparameter to account for the imbanlance between positive and negative relations, because the latter often outnumber the former. The final loss is the sum over all examples.

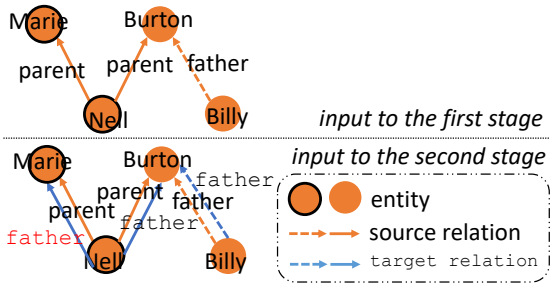



Figure 2: Input of both stages on the Nell-Marie case. Solid edges are features for Nell-Marie. Additional edges in the lower part are predicted candidate target relations $\Gamma^l$.

## 3 Collective Relation Integration

As discussed in § 1, the drawback of local methods is that predictions of different entity pairs are independently made. Neglecting their dependency may lead to predictions inconsistent with each other.

To address the issue, we propose a collective approach CoRI, which achieves collective relation integration via two stages: candidate generation and collective inference. In this section, we demonstrate the input and output of the two stages, as well as our current implementations.

### 3.1 Candidate Generation

As mentioned in § 1.1, candidate generation's responsibility is to provide candidate predictions to the collective inference stage. Formally, candidate predictions $\Gamma^l$ ($l$ means local) are generated by executing a local model on the source graph $\mathcal{K}$:

$$\Gamma^l = \operatorname*{argmax}_{\Gamma} P_\theta(\Gamma \mid \mathcal{K}). \quad (5)$$

The candidate predictions in $\Gamma^l$ may be partially wrong, but the other correct ones can help adjust

wrong predictions of their adjacent entity pairs in the collective inference stage, under the guidance of the collective model.

For example, in the upper part of Fig. 2, we have a source graph $\mathcal{K}$ with three entity pairs. The input to candidate generation is the entire $\mathcal{K}$. After applying the local model (OpenKI in our case), we have three additional edges as the output $\Gamma^l$ in the lower part of Fig. 2. Note that the candidate prediction `father` for Nell-Marie (denoted by black outline) is incorrect due to insufficient information in its neighborhood in $\mathcal{K}$, *i.e.,* both the relations in between of and around the entity pair (denoted by solid edges) are ambiguous "parent"s.

Fortunately, the entity pair Nell-Burton is relatively easy for the local model to predict as `father` because it can leverage the neighbor relation "father" between Billy-Burton. Such correct candidate predictions are included in $\Gamma^l$, provided to the collective inference stage as additional signals for later correction of the wrong predictions such as `father` for Nell-Marie.

### 3.2 Collective Inference

Collective inference's responsibility is to encode the structures of the target graph and use such information to refine the candidate predictions $\Gamma^l$ by enforcing coherence among them. To this end, a collective model $P_\beta$ (with parameters $\beta$) takes both the source graph $\mathcal{K}$ and the candidate predictions $\Gamma^l$ as input, and outputs the final predictions $\Gamma$:

$$P(\Gamma \mid \mathcal{K}) = P_\beta(\Gamma \mid \mathcal{K}, \Gamma^l). \tag{6}$$

In the Nell-Marie case of Fig. 2, when making the final prediction, its own candidate predictions and those of the neighbor entity pairs (solid edges in $\Gamma^l$ of the lower part in Fig. 2) are used to leverage the dependency among them. We concatenate the embeddings of candidate predictions to the local representation $\mathbf{t}_l$ obtained in the first stage, and represent each entity pair as follow:

$$\mathbf{t}_c = [\mathbf{t}_l; A(\Gamma^l_{s,o}); A(\Gamma^l_{s,\cdot}); A(\Gamma^l_{\cdot,o})], \tag{7}$$

where $c$ means collective. $\Gamma^l_{s,o}$ includes candidate target relations between $s$ and $o$, and similarly for $\Gamma^l_{s,\cdot}$ and $\Gamma^l_{\cdot,o}$. Then we use another multi-layer perceptron $\mathbf{MLP}_c$ to convert $\mathbf{t}_c$ to probabilities

$$P_\beta(s, r', o \mid \mathcal{K}, \Gamma^l) = \sigma(\mathbf{MLP}_c(\mathbf{t}_c))_{r'}, \tag{8}$$

and minimize the loss function for $P_\beta$ similar to that of the local model $P_\theta$ in Eq. 4.

**Algorithm 1:** Training collective model.

**Result:** Collective model $\beta$.
$\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(T)} \leftarrow$ Split training data $\mathcal{T}$ into $T$ folds;
**for** *fold* $i = 1, \ldots, T$ **do**
  $\theta^{(i)} \leftarrow$ train local model on data folds $1, \ldots, i-1, i+1, \ldots, T$;
  $\Gamma^l_i \leftarrow$ local predictions on $\mathcal{T}^{(i)}$ using $\theta^{(i)}$;
**end**
$\Gamma^l \leftarrow \cup_i \Gamma^l_i$;
$\beta \leftarrow$ train collective model on $\mathcal{T}$ with input $\mathcal{K}$ and $\Gamma^l$;

### 3.3 Training Collective Model

According to Eq. 6, we need $\Gamma^l$ as features to train the collective model $P_\beta$. This is to ensure that $P_\beta$ captures the dependencies among target relations. One may ask why we do not directly use ground truth $\mathcal{K}'$ instead of predictions $\Gamma^l$. At test time, we can only use target relations predicted by $P_\theta$ as input to $P_\beta$ because the ground truth target relations of neighbor entity pairs might not be available. If we train $P_\beta$ using the ground truth, there will be a discrepancy between training and testing, potentially hurting the performance.

Specifically, we split the training set $\mathcal{T}$ into $T$ folds. We generate $\Gamma^l$ by rotating and unioning a temporary local model's predictions on a held-out fold, where the temporary model is trained on the other folds. Then we train $P_\beta$ on the parallel data $\mathcal{T}$ with $\Gamma^l$. In this manner, we can use the full dataset to optimize the collective model while avoiding generating candidates on the training data of the local model, which leads to overfitting. The detailed training procedure is given in Alg. 1.

## 4 Data Augmentation w/ Unmatched KG

As in Def. 2, the volume of parallel data is limited by the number of shared entity pairs $\mathcal{K}_\mathcal{R} \cap \mathcal{K}'_{\mathcal{R}'}$ of the two graphs. In Fig. 1, the *unmatched* part of the target KG, containing entity pairs without extraction counterparts (*i.e.,* $\mathcal{K}'_{\mathcal{R}'} \setminus \mathcal{K}_\mathcal{R}$) and their target relations, can also indicate common substructures of the target KG, and guide the training of the collective model. To this end, we propose leveraging unmatched KG to generate *pseudo parallel data* to augment the limited training data.

**Synthesizing Pseudo Extractions.** To leverage the unmatched KG, we need to synthesize *pseudo extractions* for the target entities and relations to add to $\mathcal{K}$ as features. Since we do not use entity-specific parameters, we only synthesize source relations like "parent", and keep the target entities

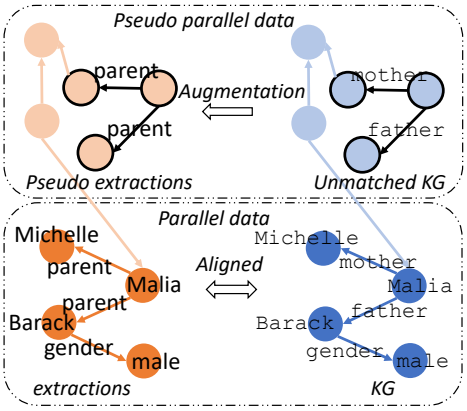



Figure 3: Illustration of parallel data augmentation. We first generate pseudo extractions for the unmatched KG, then select a subset of entity pairs that are similar to the parallel data (with black outline) to augment training.

unchanged, as illustrated in Fig. 3. Specifically, for each subject-relation-object tuple $(s', r', o')$ in the unmatched KG, we keep $s'$ and $o'$ unchanged, and synthesize source relations $r$ by sampling from:

$$P(r \mid r') = \frac{|\mathcal{K}_r \cap \mathcal{K}'_{r'}|}{|\mathcal{K}'_{r'}|}, \tag{9}$$

*i.e.,* the conditional probability of observing $r$ given $r'$ based on co-occurrences in the parallel data. $|\mathcal{K}_r \cap \mathcal{K}'_{r'}|$ is the number of entity pairs with both $r$ and $r'$ in between, and $|\mathcal{K}'_{r'}|$ is the number of entity pairs with $r'$ in between. In this way, we obtain a pseudo extraction $(s, r, o)$, as detailed in Alg. 2

**Pseudo Data Selection.** We regard all pseudo extractions as a graph $\mathcal{K}^p$. Similar to Def. 2, we define *pseudo parallel data* as below.

**Definition 3** (**Pseudo Parallel Data**). Pseudo parallel data $\mathcal{T}^p$ includes common entity pairs between pseudo extractions $\mathcal{K}^p$ and the target KG $\mathcal{K}'$, associated with their ground truth target relations, *i.e.,* $\mathcal{T}^p = \{\langle (s, o), \mathcal{K}'_{s,o} \rangle\} \mid (s, o) \in \mathcal{K}^p_{\mathcal{R}} \cap \mathcal{K}'_{\mathcal{R}'}\}$.

To make use of pseudo parallel data $\mathcal{T}^p$, the most straightforward way is to use them together with parallel data $\mathcal{T}$ to train the collective model $P_\beta$. However, not all substructures in the target graph $\mathcal{K}'$ are useful for $P_\beta$. For example, when $\mathcal{K}'$ has other domains irrelevant to the source extraction graph, substructures in those domains may distract $P_\beta$ from concentrating on the domains of the source graph. To mitigate this issue, we only use a subset of $\mathcal{T}^p$ similar to $\mathcal{T}$, as shown by the black-outlined parts in Fig. 3. Specifically, we represent each entity pair $(s, o)$ as a *virtual document* with surrounding target relations $\mathcal{K}'_{s,o} \cup \mathcal{K}'_{s,\cdot} \cup \mathcal{K}'_{\cdot,o}$ as "tokens". For each entity pair from the parallel data $\mathcal{T}$, we use BM25 (Robertson and Zaragoza, 2009) to retrieve its top $K$ most similar entity pairs from $\mathcal{T}^p$, and add them to the selected pseudo parallel data $\overline{\mathcal{T}^p}$ for training, as detailed in Alg. 2.

**Algorithm 2:** Our augmentation approach.

**Result:** Collective model $\beta$ with data augmentation.
**(1) Synthesizing Pseudo Extractions** $\mathcal{K}^p$
$\mathcal{K}^p \leftarrow \emptyset$; $\overline{\mathcal{T}^p} \leftarrow \emptyset$;
**for** $(s', r', o') \in \mathcal{K}'$, *where* $(s', o') \in \mathcal{K}'_{\mathcal{R}'} \setminus \mathcal{K}_{\mathcal{R}}$ **do**
  $s \leftarrow s'$ and $o \leftarrow o'$;
  Sample $r \sim P(r|r')$;
  $\mathcal{K}^p \leftarrow \mathcal{K}^p \cup \{(s, r, o)\}$;
**end**
**(2) Pseudo Data Selection**
**for** *entity pair* $\in \mathcal{K}_{\mathcal{R}} \cap \mathcal{K}'_{\mathcal{R}'}$ **do**
  $S \leftarrow$ its top $K$ similar entity pairs in $\mathcal{K}^p_{\mathcal{R}} \cap \mathcal{K}'_{\mathcal{R}'}$;
  $\overline{\mathcal{T}^p} \leftarrow \overline{\mathcal{T}^p} \cup \{\langle (s, o), \mathcal{K}'_{s,o} \rangle \mid (s, o) \in S\}$;
**end**
$\beta \leftarrow$ Train on $\mathcal{T} \cup \overline{\mathcal{T}^p}$ with Alg. 1;

## 5 Experimental Settings

### 5.1 Datasets and Evaluation

We use the **ReVerb** dataset (Fader et al., 2011) as the source graph, and **Freebase**[1] and **Wikidata**[2] as the target KGs, respectively. We follow the same name matching approach in Zhang et al. (2019) to obtain parallel data. To simulate real scenarios where models are trained on limited labeled data but applied to a large testing set, we use 20% of entity pairs in the parallel data for training and the other 80% for testing, and there is *no* overlap. We also compare the performance under other ratios in § 6.3. Dataset statistics are listed in Tab. 2.

| Datasets | #Train | #Test | $\lvert\mathcal{R}\rvert$ |
|---|---|---|---|
| ReVerb + Freebase | 12,344 | 49,629 | 97,196 |
| ReVerb + Wikidata | 8,447 | 33,849 | 182,407 |

Table 2: Dataset statistics. We follow Zhang et al. (2019) to use the most frequent 250 target relations.

We evaluate by ranking all integrated extractions based on their probabilities, and report area under the curve (**AUC**). Considering real scenarios where we want to integrate as many extractions as possible while keeping a high precision, we also report **Recall** and $\mathbf{F}_1$ when precision is 0.8, 0.9, or 0.95.

[1] https://developers.google.com/freebase

[2] https://www.wikidata.org

### 5.2 Compared Methods

We compare the following methods in experiments.

**Relation Translation** is a simple method that maps source relations to target relations with conditional probability $P(r' \mid r)$ similar to Eq. 9. For an entity pair $(s, o)$, the predicted target relations are $\{\arg\max_{r'} P(r'|r) \mid r \in \mathcal{K}_{s,o}\}$.

**Universal Schema (E-model)** (Riedel et al., 2013) learns entity and relation embeddings through matrix factorization, which cannot generalize to unseen entities. It is a local model that scores each integrated extraction independently.

**Rowless Universal Schema** (Verga et al., 2017) is a local model which improves over the E-model by eliminating entity-specific parameters, thus generalizing to unseen entities.

**OpenKI** (Zhang et al., 2019) is a local model that addresses the ambiguity of source relations by using neighbor relations for more context.

**CoRI** is our collective two-stage relation integration model trained with Alg. 1.

**CoRI + DA** is our model where the training data is augmented by pseudo parallel data with Alg. 2. To verify the necessity of **retrieval**-based pseudo data selection, we also compare with a **random** DA baseline where we select $K$ random entity pairs.

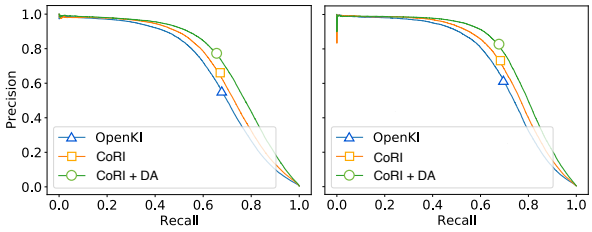
**CoRI + KGE** is another approach to exploit the unmatched KG with KG embeddings (**KGE**) trained on the entire target KG in an unsupervised manner. We initialize the embeddings of target relations averaged by $A(.)$ in Eq. 7 with TransE (Bordes et al., 2013) embeddings trained on the target graph.

### 5.3 Implementation Details

We uniformly use 32-dimension embeddings for all relations, and AdamW (Loshchilov and Hutter, 2019) optimizer with learning rate 0.01 and epsilon $10^{-8}$. The ratio $\gamma$ in Eq. 4 is set to 10. We sample at most 30 neighbor source relations to handle entity pairs with too many neighbor relations. We use $T = 5$ folds in Alg. 1 to train our collective model. We retrieve top $K = 5$ entity pairs in pseudo data selection, adding about 20K and 12K entity pairs to the two datasets in Tab. 2, respectively. We use BM25 (Robertson and Zaragoza, 2009) implementation in ElasticSearch[3] in pseudo data selection. We use the pretrained KGE released by OpenKE.[4] Our model is trained with 32 CPU cores and a single 2080Ti GPU, and it takes 1-2 hours to converge.

[3] https://www.elastic.co/

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




(a) ReVerb + Freebase (b) ReVerb + Wikidata

Figure 4: Precision-recall curves of best three methods.

## 6 Experimental Results

We aim to answer the following questions: **(1)** Is CoRI superior to local models? **(2)** Is CoRI robust *w.r.t.* varying size of training and testing data? **(3)** Is unmatched KG useful for CoRI? Is our parallel data augmentation approach the best choice?

### 6.1 Main Results

In Tab. 3, we show results comparing all methods on both datasets. Our observations are as follows.

**Collective inference is beneficial.** Among the baselines, OpenKI generally performs best because it leverages neighbor relations besides middle relations between entity pairs, without relying on entity parameters. Even without data augmentation, CoRI outperforms OpenKI by a large margin, improving AUC from .677 to .708 and from .716 to .746 on the two datasets, respectively, which demonstrates the effectiveness of collective inference.

**Data augmentation further improves the performance.** By comparing CoRI with CoRI + DA (retrieval), we observe that data augmentation further improves AUC from .708 to .748 and from .746 to .780, respectively, which indicates that using unmatched KG can effectively augment the training of the collective model. We plot the precision-recall curves of the best three approaches in Fig. 4. It demonstrates the superiority of our methods across the whole spectrum.

**Generalization on unseen entities is necessary.** Among the baselines, the E-model uses entity-specific parameters, hindering it from generalizing to unseen entities and making it less competitive.

### 6.2 Effectiveness of Pseudo Data Selection

As shown in Tab. 3, both KGE, random, and retrieval-based data augmentation approaches perform better than CoRI (without DA), indicating the effectiveness of using the unmatched KG. Our retrieval-based DA outperforms the random coun-

| Datasets | ReVerb + Freebase | | | | | | | ReVerb + Wikidata | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Metrics** | AUC | Prec = 0.8 | | Prec = 0.9 | | Prec = 0.95 | | AUC | Prec = 0.8 | | Prec = 0.9 | | Prec = 0.95 | |
| | | Rec | $F_1$ | Rec | $F_1$ | Rec | $F_1$ | | Rec | $F_1$ | Rec | $F_1$ | Rec | $F_1$ |
| Translation | .571 | .590 | .679 | .100 | .180 | .067 | .125 | .604 | .595 | .683 | .088 | .160 | .042 | .080 |
| E-model | .205 | .014 | .027 | .010 | .020 | .005 | .010 | .214 | - | - | - | - | - | - |
| Rowless | .593 | .473 | .594 | .372 | .526 | .186 | .310 | .647 | .511 | .624 | .381 | .536 | .266 | .416 |
| OpenKI | .677 | .553 | .654 | .449 | .599 | .314 | .472 | .716 | .605 | .689 | .511 | .652 | .407 | .570 |
| CoRI | .708 | .590 | .679 | .494 | .638 | .381 | .544 | .746 | .641 | .712 | .558 | .689 | .461 | .621 |
| + KGE | .711 | .597 | .684 | .514 | .654 | .418 | .581 | .763 | .662 | .725 | .596 | .717 | .520 | .672 |
| + DA (random) | .734 | .616 | .696 | .518 | .658 | .395 | .558 | .774 | .678 | .734 | .606 | .724 | .521 | .673 |
| + DA (retrieval) | **.748** | **.636** | **.708** | **.539** | **.674** | **.421** | **.583** | **.780** | **.685** | **.738** | **.613** | **.729** | **.529** | **.680** |

Table 3: Main experimental results. The best results are **in bold**, and the best external baselines are underlined. CoRI outperforms the best baseline OpenKI by a large margin, and parallel data augmentation (DA) further improves its performance. "-" indicates that the precision was not achieved.

terpart, which confirms the superiority of similarity-based data augmentation in choosing substructures that cover domains relevant to the original parallel data. Our DA approach outperforms KGE, demonstrating the necessity of selectively using the unused KG to avoid discrepancy with the parallel data.

**Different Numbers of Pseudo Data Entity Pairs.** In Fig. 5, we compare the performance of DA *w.r.t.* different numbers of retrieved entity pairs $K$. We observe that $K$=5 yields better performance than $K$=1. However, further increasing $K$ hurts the performance, which is probably due to pseudo entity pairs with lower similarity to the parallel data causing a domain shift. This validates the necessity of selectively using the pseudo parallel data.

### 6.3 Impacts of Data Size on CoRI

Due to its collective nature, one may wonder about CoRI's performance *w.r.t.* other training and testing data sizes. We analyze these factors in this section. Our observations are similar on both datasets, so we only report the results on ReVerb + Freebase.

**Varying Size of Training Data.** In Fig. 6a, we compare CoRI (without DA) with OpenKI by varying the portion of the parallel data for training from 20% (used in our main results in Tab. 3) to 80%. We observe that using more training data improves the performance, as shown by the increasing trends *w.r.t.* all metrics. Our method outperforms OpenKI in all settings, demonstrating that our method is effective in both high- and low-resource settings.

**Varying % of Accessible Neighbor Entity Pairs.** Our collective framework is special in its collective inference stage, where the collective model refines the candidate prediction of an entity pair by considering its neighbor entity pairs' candidates. We hypothesize that the more neighbor entity pairs the collective model has access to, the better performance it should achieve. For example, if we use a portion of 50%, candidate predictions for only half of the neighbor entity pairs rather than the entire $\Gamma^l$ will be used in Eq. 7. We vary the portion from 25% to 100% (used in our main experiments in Tab. 3). As shown in Fig. 6b, even accessing 25% can make CoRI outperform OpenKI. As the percentage increases, CoRI continues to improve, while OpenKI remains the same because it is local, *i.e.,* not using candidate predictions.

### 6.4 Case Study

In Fig. 7, we show two cases from ReVerb + Freebase where CoRI corrects the mistakes of OpenKI in the collective inference stage. In the first case, the source relation "is in" between "Iowa" and "Mahaska County" is extracted but in the wrong direction. OpenKI just straightforwardly predicts `containedby` based on the surface form, but fails to leverage the neighbor relations to infer that Iowa is a larger geographical area. With the collective model, CoRI is able to use the other two candidate predictions of `containedby` to flip the wrong prediction to `contains`.

In the second case, a prediction is needed between "Bily Joel" and "Columbia". Here the source relation "was in" and the object entity "Columbia" are both ambiguous, which can refer to geographical containment with a place or membership to a company. OpenKI makes no prediction due to the ambiguity, while CoRI makes the right prediction `music_label` by collectively working on the other entity pairs, where all predictions coherently indicate that "Columbia" is a music company.

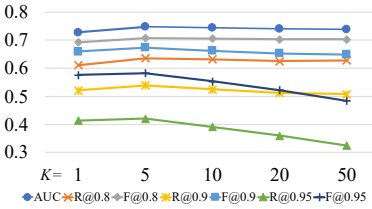
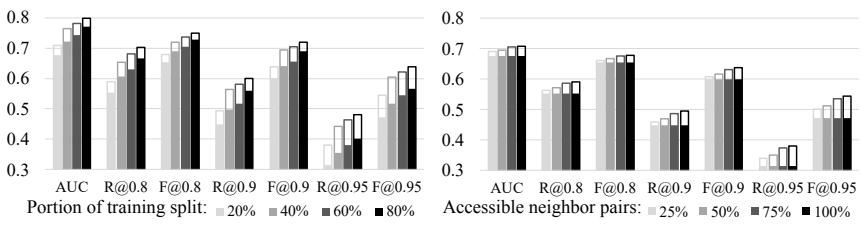



Figure 5: Performance of data augmentation with different numbers of retrieved pairs $K$.




(a) Varying size of training data.

(b) Varying % of accessible neighbor pairs

Figure 6: CoRI (bars without filling) vs. OpenKI (solid bars) on ReVerb + Freebase. CoRI consistently outperforms OpenKI by a large margin. Larger improvements are achieved when candidates of more neighbor entity pairs are accessed.

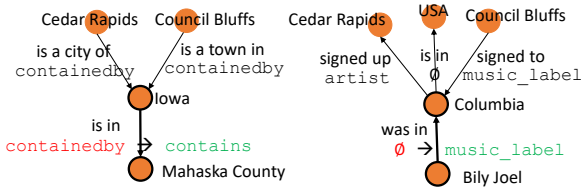



Figure 7: Two cases from ReVerb + Freebase with predictions in this `font`. The `wrong` predictions of OpenKI is `corrected` by our collective model.

## 7 Related Work

Relation integration has been studied by both the database (DB) and the NLP communities. The DB community formulates it as schema matching that aligns the schemas of two tables, *e.g.,* matching columns of an `is_in` table to those of another `subarea_of` table (Rahm and Bernstein, 2001; Cafarella et al., 2008; Kimmig et al., 2017). Such table-level alignment is valid since all rows in an `is_in` table should have the same semantics, *i.e.,* being geographical containment or not. However, in open IE, predictions should be made at the entity pair level because of the ambiguous nature of source relations. Putting all extracted "is in" entity pairs into one table to conduct schema matching is problematic from the first step since the entity pairs may have different ground truths.

The NLP community, on the other hand, investigates the problem at the entity pair level. Besides manually designed rules (Soderland et al., 2013), most works leverage the link structure between entities and relations. Universal schema (Riedel et al., 2013) learns embeddings of entities and middle relations between entity pairs through decomposing their co-occurrence matrix. However, the entity embeddings make it not generalize to unseen entities. Other methods (Toutanova et al., 2015; Verga et al., 2016, 2017; Gupta et al., 2019) also exploit middle relations, but eliminate entity parameters. Zhang et al. (2019) moves one step further by explicitly considering neighbor relations, leveraging more context from the local link structure. Some works (Weston et al., 2013; Angeli et al., 2015) directly minimize the distance between embeddings of relations sharing the same entity pairs. Yu et al. (2017) further leverage compositional representations of entity names instead of using free parameters to deal with unseen entities at test time.

There are also works on Open IE canonicalization that cluster source relations. Some use entity pairs as clustering signals (Yates and Etzioni, 2009; Nakashole et al., 2012; Galárraga et al., 2014), while others use lexical features or side information (Min et al., 2012; Vashishth et al., 2018). However, the clusters are not finally aligned to relations in target KGs, different from our problem.

The two-stage collective inference framework has been explored in other problems like entity linking (Cucerzan, 2007; Guo et al., 2013; Shen et al., 2012), where candidate entities are generated for each mention independently, and collectively ranked based on their compatibility in the second stage. In machine translation, an effective approach to leverage monolingual corpus in the target language is to back-translate it to the source language to augment the limited parallel corpus (Sennrich et al., 2016). The above works inspired us to use collective inference for relation integration and leverage the unmatched KG for data augmentation. Another approach to perform collective inference is to solve learning problem with constraints, such as integer linear programming (Roth and Yih, 2004), posterior regularization (Ganchev et al., 2010), and conditional random fields (Lafferty et al., 2001). Comparing to our approach, these methods usually involve heavy computation, or are hard to optimize. Examining the perfor-

mance of these methods is an interesting future direction. Besides, we also adopted ideas of selecting samples from out-domain data similar to in-domain samples (Xu et al., 2020; Du et al., 2020) to select our pseudo parallel data.

## 8 Conclusion

In this paper, we proposed CoRI, a collective inference approach to relation integration. To the best of our knowledge, this is the first work exploring this idea. We devised a two-stage framework, where the candidate generation stage employs existing local models to make candidate predictions, and the collective inference stage refines the candidate predictions by enforcing global coherence. Observing that the target KG is rich in substructures indicating the desired global coherence, we further proposed exploiting the unmatched KG by selectively synthesizing pseudo parallel data to augment the training of our collective model. Our solution significantly outperforms all baselines on two datasets, indicating the effectiveness of our approaches.

## Acknowledgments

We would like to thank Prashant Shiralkar, Hao Wei, Colin Lockard, Binxuan Huang, and all the reviewers for their insightful comments and suggestions.

## References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 708–716.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding. *CoRR*, abs/2010.02194.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1535–1545. ACL.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1679–1688. ACM.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 378–388. Association for Computational Linguistics.

Angelika Kimmig, Alex Memory, Renée J. Miller, and Lise Getoor. 2017. A collective, probabilistic approach to schema mapping. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 921–932. IEEE Computer Society.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1027–1037. ACL.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1135–1145. ACL.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 74–84. The Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 1–8. ACL.

Satoshi Sekine. 2006. On-demand information extraction. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on World Wide Web*, pages 449–458.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. 2013. Open information extraction to KBP relations in 3 hours. In *Proceedings of the Sixth Text Analysis Conference, TAC 2013, Gaithersburg, Maryland, USA, November 18-19, 2013*. NIST.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509. The Association for Computational Linguistics.

Shikhar Vashishth, Prince Jain, and Partha P. Talukdar. 2018. CESI: canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1317–1327. ACM.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 886–896. The Association for Computational Linguistics.

Patrick Verga, Arvind Neelakantan, and Andrew McCallum. 2017. Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 613–622. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1366–1371. ACL.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Frank F. Xu, Zhengbao Jiang, Pengcheng Yin, Bogdan Vasilescu, and Graham Neubig. 2020. Incorporating external knowledge through pre-training for natural language to code generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6045–6052. Association for Computational Linguistics.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Intell. Res.*, 34:255–296.

Dian Yu, Lifu Huang, and Heng Ji. 2017. Open relation extraction and grounding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 854–864. Asian Federation of Natural Language Processing.

Dongxu Zhang, Subhabrata Mukherjee, Colin Lockard, Xin Luna Dong, and Andrew McCallum. 2019. Openki: Integrating open information extraction and knowledge bases with relation inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 762–772. Association for Computational Linguistics.