# Cross-Sentence $N$-ary Relation Extraction using Lower-Arity Universal Schemas

**Kosuke Akimoto[1], Takuya Hiraoka[1], Kunihiko Sadamasa[1], Mathias Niepert[2]**
[1] Security Research Laboratories, NEC Corporation
[2] NEC Laboratories Europe
{k-akimoto@ab,t-hiraoka@ce,k-sadamasa@az}.jp.nec.com
mathias.niepert@neclab.eu

## Abstract

Most existing relation extraction approaches exclusively target binary relations, and $n$-ary relation extraction is relatively unexplored. Current state-of-the-art $n$-ary relation extraction method is based on a supervised learning approach and, therefore, may suffer from the lack of sufficient relation labels. In this paper, we propose a novel approach to cross-sentence $n$-ary relation extraction based on *universal schemas*. To alleviate the sparsity problem and to leverage inherent decomposability of $n$-ary relations, we propose to learn relation representations of lower-arity facts that result from decomposing higher-arity facts. The proposed method computes a score of a new $n$-ary fact by aggregating scores of its decomposed lower-arity facts. We conduct experiments with datasets for ternary relation extraction and empirically show that our method improves the $n$-ary relation extraction performance compared to previous methods.

## 1 Introduction

Relation extraction is a core natural language processing task which is concerned with the extraction of relations between entities from text. It has numerous applications ranging from question answering (Xu et al., 2016) to automated knowledge base construction (Dong et al., 2014).

While the vast majority of existing research focuses on extracting *binary* relations, there exists only few recent approaches to extract $n$-ary relations, that is, relations among $n \geq 2$ entities (Li et al., 2015; Ernst et al., 2018). In $n$-ary relation extraction, relation mentions tend to span multiple sentences more frequently as $n$ increases. Thus, Peng et al. (2017) recently extended the problem to cross-sentence $n$-ary relation extraction in which $n$-ary relations are extracted from multiple sentences. As a motivating example, consider the following text from Wikipedia: "*Revis*

*started off the* **2009 season** *matched up against some of football's best wide receivers. In Week 1, he helped limit Houston* **Texans** *Pro-bowler* **Andre Johnson** *to four receptions for 35 yards.*" In this example, two sentences collectively describes that Andre Johnson is a player of the football team the Texans during 2009 season, and thus we need cross-sentence information to correctly extract this ternary interaction among the three entities, i.e. Player(Andre Johnson, Texans, 2009 season).

Previous methods (Peng et al., 2017; Song et al., 2018) capture cross-sentence $n$-ary relation mentions by representing texts with a *document graph* which consists of both intra- and cross-sentence links between words. With this graphical representation, they applied graph neural networks to predict ternary relations in the medical domain. However, these methods train the neural networks in a supervised manner using distant supervision (Mintz et al., 2009) and, therefore, may suffer from the lack of sufficient positive labels when a well-populated knowledge base is not available.

On the other hand, for binary relation extraction, the problem of insufficient positive labels can be mitigated with *universal schemas* (Riedel et al., 2013). In a universal schema approach, textual representations (surface patterns) of entities and their relations are encoded into the same vector space as the canonical knowledge base relations. Thus, semantically similar surface patterns can share information of relation labels in a semi-supervised manner. This reduces the amount of required labeled training data. Applying the universal schema approach to $n$-ary ($n > 2$) relation extraction is, however, not straight-forward due to the sparsity of higher-order relation mentions among a specific set of $n > 2$ entities.[1] This is be-

---

[1] In our Wiki-90k dataset (see §4.1), only 12.5% of ternary entity tuples have at least two relations among the entities, while 77.3% of entities appear at least twice.

cause the universal schema approach (Riedel et al., 2013) and its extensions (Toutanova et al., 2015; Verga et al., 2016, 2017) utilize co-occurring patterns of relation types between specific pair of entities. Also, prior work has only addressed binary relations, and it is not trivial to define surface patterns among $n > 2$ entities and to encode these patterns into a vector representation.

To mitigate the aforementioned sparsity problem and utilize existing encoders for binary and unary surface patterns, we propose to train universal schema models on more dense lower-arity (unary and binary) facts instead of original sparse $n$-ary facts. Since most $n$-ary relations can be decomposed into a set of $k$-ary relations ($k = 1, 2$) which are implied by the $n$-ary relation,[2] we can easily acquire lower-arity facts by decomposing $n$-ary facts. Our model learns representations of these lower-arity relations using the universal schema framework, and predicts new $n$-ary facts by aggregating scores of lower-arity facts.

To evaluate the proposed method, we create new cross-sentence $n$-ary relation extraction datasets with multiple ternary relations.[3] The new datasets contain more entity tuples with known relational facts appeared in a knowledge base than the existing dataset (Peng et al., 2017), and, therefore, these datasets can be used to more effectively evaluate methods which predict relation labels for each individual entity tuple. We show empirically that by jointly training lower-arity models and an $n$-ary score aggregation model, the proposed method improves the performance of $n$-ary relation extraction. To the best of our knowledge, this is the first attempt to apply universal schemas to $n$-ary relation extraction, taking advantage of the compositionality of higher-arity facts.

## 2 Task Definition and Notation

The cross-sentence $n$-ary relation extraction task (Peng et al., 2017) is defined as follows. Let $\mathcal{E}$ be a set of entities, $\mathcal{R}_{\text{KB}}$ be a set of relation types of an external knowledge base $\mathcal{KB}$, and $\mathcal{O}_{\text{KB}} = \{\langle r, (e_1, ..., e_n)\rangle : r(e_1, ..., e_n) \in \mathcal{KB}, r \in$

[2]For example, the ternary relation `AwardedFor`($director, movie, award$) can be decomposed into the binary relations `DirectorOf`($director, movie$) and `WonAward`($director, award$). Note that a similar idea is introduced in (Ernst et al., 2018) as *partial facts* or *partial patterns*.

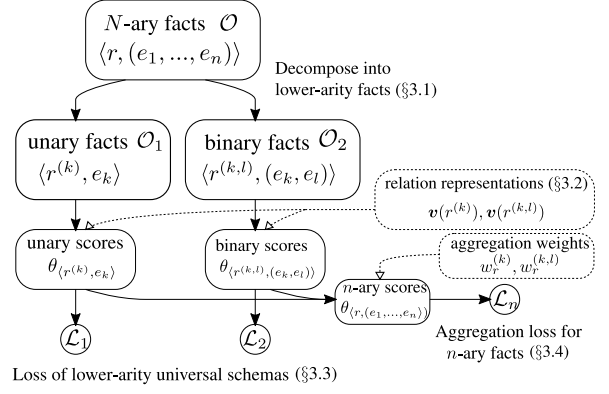[3]Our codes and datasets are available at https://github.com/aurtg/nary-relation-extraction-decomposed.
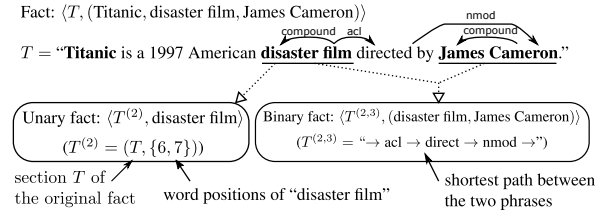
Figure 1: An overview of the proposed method.



Figure 2: An example of decomposed textual facts.

$\mathcal{R}_{\text{KB}}, e_i \in \mathcal{E}\}$ be the set of facts in $\mathcal{KB}$. We collect a set of candidate entity tuples among which KB relation $r \in \mathcal{R}_{\text{KB}}$ possibly holds.[4] Here, all entities in each candidate tuple $(e_1, ..., e_n)$ are mentioned in the same text section $T$ in a given set of documents. We define a set of these entity mentions as $\mathcal{O}_{\text{text}} = \{\langle T, (e_1, ..., e_n)\rangle : e_i \in \mathcal{E}$ is mentioned in $T\}$. Here, text section $T$ is a (short) span in a document which can describes relational facts among entities. In the cross-sentence $n$-ary relation extraction task, text section $T$ can contain multiple sentences. In this paper, following (Peng et al., 2017), we define $M$ consecutive sentences ($M \geq 1$) which contain $n$ target entities as a text section in the cross-sentence $n$-ary relation extraction task. We use the term "relation" to refer to both relations $r \in \mathcal{R}_{\text{KB}}$ and sections $T$.

The goal of the cross-sentence $n$-ary relation extraction task is to predict new facts $\langle r, (e_1, ..., e_n)\rangle \notin \mathcal{O}_{\text{KB}}$ for relation $r \in \mathcal{R}_{\text{KB}}$ given $\mathcal{O} = \mathcal{O}_{\text{KB}} \cup \mathcal{O}_{\text{text}}$, where $n \geq 2$.

## 3 Proposed Method

### 3.1 Lower-Arity Facts

To alleviate the sparsity problem of facts among $n$ entities ($n > 2$) and to utilize well-studied encoders for binary and unary surface patterns, we

[4]It is allowed that multiple KB relations hold among the same set of entities.

decompose a set of original $n$-ary facts, $\mathcal{O}$, into a set of unary facts $\mathcal{O}_1$ and a set of binary facts $\mathcal{O}_2$ (Figure 1).

**Unary Facts:** Given an $n$-ary fact $\langle r, (e_1, ..., e_k) \rangle \in \mathcal{O}$, we decompose it into a set of $n$ unary facts $\{\langle r^{(k)}, e_k \rangle : k = 1, ..., n\}$, where $r^{(k)}$ is a tentative unary relation w.r.t. the $k$-th argument of the original relation $r$. If $r$ is a KB relation, we define unary relation $r^{(k)}$ as a new canonicalized relation. If $r$ is section $T$, we define unary relation $r^{(k)}$ as a tuple $r^{(k)} = (T, pos(e_k))$, where $pos(e_k)$ is a set of word position indices of entity $e_k$ in section $T$ (Figure 2). We denote a set of all decomposed unary facts by $\mathcal{O}_1$. Intuitively, these unary relations represent semantic roles or types of corresponding arguments of the original relation $r$ (Yao et al., 2013).

**Binary Facts:** Given an $n$-ary fact $\langle r, (e_1, ..., e_k) \rangle \in \mathcal{O}$, we decompose it into a set of $n(n-1)$ binary facts $\{\langle r^{(k,l)}, (e_k, e_l) \rangle : k, l = 1, ..., n, k \neq l\}$, where $r^{(k,l)}$ is a tentative binary relation between the $k$-th and $l$-th argument of the original relation $r$. If $r$ is a KB relation, we define binary relation $r^{(k,l)}$ as a new canonicalized relation. If $r$ is a section $T$, we represent it by the shortest path between $e_k$ and $e_l$ on the *document graph* (Quirk and Poon, 2017) of $T$ (Figure 2), and denote it by $path(T; e_k, e_l)$. We denote the set of all decomposed binary facts by $\mathcal{O}_2$.

### 3.2 Lower-Arity Relation Representations

We learn a vector representation $\boldsymbol{v}(r) \in \mathbb{R}^{d_r}$ for each unary or binary relation in $\mathcal{O}_1$ or $\mathcal{O}_2$. For $r^{(k)}$ or $r^{(k,l)}$ derived from a KB relation, we represent it by a trainable parameter vector. On the other hand, for the one derived from a textual relation, we use the following encoders to compute its representations.

**Unary encoder:** For an unary textual relation $r^{(k)} = (T, pos(e_k))$, we represent each section $T$ by a sequence of word vectors and use a bidirectional LSTM (Bi-LSTM) (Schuster and Paliwal, 1997) to compute a hidden representation $\boldsymbol{h}_l \in \mathbb{R}^{d_r}$ at each word position $l$. Following recent works (Zhang et al., 2018; He et al., 2018; Lee et al., 2017), we aggregate $\boldsymbol{h}_l$ within a phrase of entity $e_k$ to compute $\boldsymbol{v}(T^{(k)})$. We use element-wise mean as aggregation function:

$$\boldsymbol{v}(r^{(k)}) = \text{mean}(\{\boldsymbol{h}_l : l \in pos(e_k)\}). \quad (1)$$

**Binary encoder:** For a binary textual relation

$r^{(k,l)} = path(T; e_k, e_l)$, we represent each token (word or edge label) in $path(T; e_k, e_l)$ by an embedding vector (Toutanova et al., 2015; Verga et al., 2016). We use a Bi-LSTM to compute a hidden representation $\boldsymbol{h}'_l \in \mathbb{R}^{d_r}$ at each token position $l$, and max-pool along the path to compute the relation representation:

$$\boldsymbol{v}(T^{(k,l)}) = \max(\{\boldsymbol{h}'_l : l = 1, ..., L\}). \quad (2)$$

### 3.3 Learning Relation Representations

We follow Verga et al. (2017) to train relation representations (§3.2). We define a score $\theta_{\langle r, p \rangle}$ for each lower-arity fact $\langle r, p \rangle \in \mathcal{O}_1 \cup \mathcal{O}_2$, and minimize the following loss (3) for each arity $i = 1, 2$. Here, placeholder $p$ refers to either an entity (if $\langle r, p \rangle \in \mathcal{O}_1$) or an entity tuple (if $\langle r, p \rangle \in \mathcal{O}_2$), and we simply refer to both as entity tuple. The loss functions contrast a score of an original fact $\langle r, p^+ \rangle \in \mathcal{O}_i$ and those of $K$ sampled negative facts $\langle r, p_k^- \rangle \notin \mathcal{O}_i$. We sample negative facts by randomly replacing entity tuple $p^+$ in the original fact by different entity tuples $p_k^-$.

$$\mathcal{L}_i = \mathop{\mathbb{E}}_{\substack{\langle r, p^+ \rangle \in \mathcal{O}_i \\ \langle r, p_k^- \rangle \notin \mathcal{O}_i}} [-\log(\frac{\exp(\theta_{\langle r, p^+ \rangle})}{\exp(\theta_{\langle r, p^+ \rangle}) + \sum_k \exp(\theta_{\langle r, p_k^- \rangle})})].$$
$$(3)$$

The score of fact $\langle r, p \rangle$ is defined as $\theta_{\langle r, p \rangle} = \boldsymbol{v}(r)^T \boldsymbol{v}(p; r)$. Entity tuple representations $\boldsymbol{v}(p; r)$ are computed with a weighted average of the representations $\{\boldsymbol{v}(r') : r' \in V(p)\}$ as shown in (4) and (5) where $a(r', r; V(p))$ is the attention weight for each relation $r' \in V(p)$.[5]

$$\boldsymbol{v}(p; r) = \sum_{r' \in V(p)} a(r', r; V(p))\boldsymbol{v}(r'),$$
$$a(r', r; V(p)) = \frac{\exp(\boldsymbol{v}(r')^T \boldsymbol{v}(r))}{\sum_{r'' \in V(p)} \exp(\boldsymbol{v}(r'')^T \boldsymbol{v}(r))}. \quad (4)$$

$$V(p) = \begin{cases} \{r : \langle r, e_k \rangle \in \mathcal{O}_1\} & \text{(if } p = e_k) \\ \{r : \langle r, (e_k, e_l) \rangle \in \mathcal{O}_2\} & \text{(if } p = (e_k, e_l)) \end{cases}.$$
$$(5)$$

### 3.4 Aggregating Lower-Arity Scores

To predict $n$-ary facts of KB relation $r \in \mathcal{R}_{\text{KB}}$, we compute its score $\theta_{\langle r, (e_1, ..., e_n) \rangle}$ by aggregating lower-arity scores as in (6), where $w_r^{(\cdot)}$ is a positive scalar weight defined for each KB relation which sum to one: $\sum_k w_r^{(k)} + \sum_{k \neq l} w_r^{(k,l)} = 1$.

---

[5] During training, as in (Verga et al., 2017), we aggregate sub-sampled $M$ relations from $V(p)$ if $|V(p)| > M$. We set $M = 2$ for all experiments.

We can set all weights $w_r^{(k)}$ and $w_r^{(k,l)}$ to $1/n^2$, or train these weights to give higher scores to positive $n$-ary facts by minimizing additional loss function $\mathcal{L}_n$. Note that $\mathcal{L}_n$ directly contrasts $n$-ary scores associated with KB relations $r \in \mathcal{R}_{\text{KB}}$ in a more supervised manner than both $\mathcal{L}_1$ and $\mathcal{L}_2$.[6]

$$
\theta_{\langle r, (e_1, \ldots, e_n)\rangle} =
$$
$$
\sum_{k=1}^{n} w_r^{(k)} \theta_{\langle r^{(k)}, e_k\rangle} + \sum_{\substack{k=1,\ldots,n \\ l=1,\ldots,n \\ k \neq l}} w_r^{(k,l)} \theta_{\langle r^{(k,l)}, (e_k, e_l)\rangle}. \tag{6}
$$

$$
\mathcal{L}_n = \mathop{\mathbb{E}}_{\substack{\langle r, p^+ \rangle \in \mathcal{O}_{\text{KB}} \\ \langle r, p^- \rangle \notin \mathcal{O}_{\text{KB}}}} \left[ \max(0, 1 - \theta_{\langle r, p^+ \rangle} + \theta_{\langle r, p^- \rangle}) \right] \tag{7}
$$

The overall loss function is now $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \alpha \mathcal{L}_n$. By changing $\alpha$, we can balance the semi-supervised effect of lower-arity universal schemas ($\mathcal{L}_1, \mathcal{L}_2$) and that of the supervision with $n$-ary relation labels ($\mathcal{L}_n$).

# 4 Experiments

## 4.1 Dataset

The cross-sentence $n$-ary relation extraction dataset from Peng et al. (2017) contains only 59 distinct ternary KB facts including the train and test set. Since our proposed method and universal schemas baselines predict KB relations for each *entity tuple* instead of each *surface pattern*, the number of known facts of KB relations is crucial to reliably evaluate and compare these methods. Thus, we created two new $n$-ary cross-sentence relation extraction datasets (dubbed with **Wiki-90k** and **WF-20k**) that contain more known facts retrieved from public knowledge bases.

To create the Wiki-90k and WF-20k datasets, we used Wikidata and Freebase respectively as external knowledge bases. Since these knowledge bases store only binary relational facts, we defined multiple ternary relations by combining a few binary relations.[7,8] For both datasets, we collected paragraphs from the English Wikipedia, and used Stanford CoreNLP (Manning et al., 2014) to extract dependency and co-reference links. Entity mentions are detected using DBpedia Spotlight (Daiber et al., 2013). We followed (Peng et al., 2017) to extract co-occurring entity tuples and their surface patterns, that is, we selected tuples which occurred in a *minimal span* within at most $M \leq 3$ consecutive sentences. Entity tuples without a known KB relation are subsampled, since the number of such tuples are too large. We randomly partitioned all entity tuples into train, development (dev), and test sets.

## 4.2 Baselines

**(Song et al., 2018):** The state-of-the-art cross-sentence $n$-ary relation extraction method proposed by Song et al. (2018) represents each surface pattern by the concatenation of entity vectors from the last layer of a Graph State LSTM, a variant of a graph neural network. The concatenated vector is then fed into a classifier to predict the relation label. Since their method directly predicts a relation label for each surface pattern, it is more robust to the sparsity of surface patterns among a specific higher arity entity tuple. However, due to their purely supervised training objective, its performance may degrade if the number of available training labels is small.

**Universal schemas:** We compared our method with semi-supervised methods based on universal schemas (Toutanova et al., 2015; Verga et al., 2017). In our experiments, we used the same encoder as (Song et al., 2018) to encode each surface pattern.[9] We tested two types of scoring functions, Model F and Model E, as in (Toutanova et al., 2015).[10,11]

## 4.3 Evaluation

We compared the methods in the held-out evaluation as in (Mintz et al., 2009) and report (weighted) mean average precision (MAP) (Riedel et al., 2013). Unless otherwise noted, reported values are average values over six experiments, in which network parameters are randomly initialized. All reported $p$-values are calculated based on Wilcoxon rank sum test (Wilcoxon, 1945) with

---

[6] The loss function (7) performs better than a log-likelihood based loss, $-\log\sigma(\theta_{\langle r, p^+ \rangle} - \theta_{\langle r, p^- \rangle})$.

[7] See the appendix A for details about defined ternary relations.

[8] For about half of the defined ternary relations, combined original binary relations in KB are different from decomposed binary relations of the ternary relations in the proposed method (§3.1).

[9] Using linear projection instead of simple concatenation did not improve performance in our preliminary experiments.

[10] It is not trivial to apply Model E scoring function with Verga et al. (2017) method, since their aggregation method calculates a representation for each row, i.e. entity tuple.

[11] DistMult scoring function in (Toutanova et al., 2015) showed poor performance compared to the other two scoring functions in our preliminary experiments.

| Method | Wiki-90k | | WF-20k | |
|---|---|---|---|---|
| | average | weighted | average | weighted |
| Proposed | **0.584** | **0.634** | **0.821** | **0.842** |
| (Song et al., 2018) | 0.471 | 0.536 | 0.639 | 0.680 |
| (Toutanova et al., 2015) with Graph State LSTM (Song et al., 2018) encoder | | | | |
| Model F | 0.240 | 0.262 | 0.341 | 0.380 |
| Model E | 0.399 | 0.414 | 0.725 | 0.752 |
| (Verga et al., 2017) with Graph State LSTM (Song et al., 2018) encoder | | | | |
| Model F | 0.443 | 0.482 | 0.610 | 0.653 |

**bold**: $p \leq 0.01$

Table 1: Mean average precisions (MAPs) on **test** data.

## 4.4 Results

Table 1 illustrates the performance of each method.[12] Compared to the baseline methods, our proposed method achieves higher weighted MAP for both datasets. Interestingly, Model F performs well in Verga et al. (2017) baseline, while it shows low performance in Toutanova et al. (2015) baseline.

**Ablation Study:** Table 2 illustrates the performance of various settings of our proposed method. **U**,**B**, and **N** stand for using the loss functions $\mathcal{L}_1$, $\mathcal{L}_2$, and $\alpha\mathcal{L}_n$ respectively. In the result, **U+B** performs significantly better ($p < 0.005$) than **U** and **B**, and this shows effectiveness of combining scores of both binary facts and unary facts. On the other hand, there was no significant difference between **U+B+N** and **N** ($p > 0.9$). Note that we used all positive labels in this experiment, that is, sufficient amount of positive labels are used for calculating the loss **N**.

**Data efficiency:** Furthermore, we also investigated the influence of the training data size (the number of positive labels) of our proposed method and baseline methods.[13] Here, $\alpha = \infty$ stands for optimizing $\mathcal{L}_n$ instead of $\mathcal{L}_1 + \mathcal{L}_2 + \alpha\mathcal{L}_n$. As shown in Figure 3, $\alpha = 1$ achieved higher performance than $\alpha = \infty$, showing that introducing lower-arity semi-supervised loss ($\mathcal{L}_1 + \mathcal{L}_2$) improves the performance for dataset with few positive labels. On the other hand, the lower performance of $\alpha = 0$ compared to $\alpha = 0.1, 1$ suggests that information of higher-arity facts introduced from $\mathcal{L}_n$ is beneficial for $n$-ary relation extraction.

## 5   Conclusion and Future Works

We proposed a new method for cross-sentence $n$-ary relation extraction that decomposes sparse $n$-

---

[12]For the proposed method, we set $\alpha = 10$.

[13]In this experiment, we conducted four experiments per each setting and set $K = 10$.

| Setting | average | weighted |
|---|---|---|
| U | 0.363 | 0.404 |
| B | 0.411 | 0.452 |
| N | 0.646 | 0.685 |
| U+B | 0.521 | 0.552 |
| U+B+N | 0.646 | 0.682 |
| U+B+N ($K = 10$) | **0.650** | **0.689** |
| U+B+N ($K = 5$) | 0.592 | 0.640 |
| U+B+N (fix $w_r = 1$) | 0.645 | 0.679 |

Default hyperparameter: $K = 20$

Table 2: Ablation study: mean average precisions (MAPs) on **dev** data (Wiki-90k).
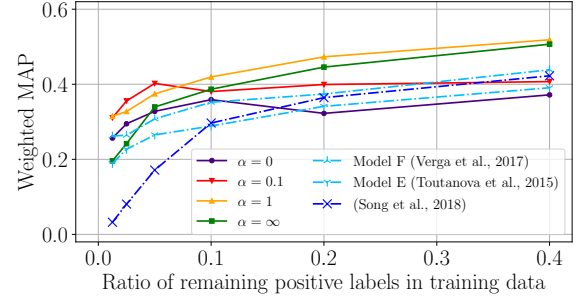


Figure 3: Weighted MAP on test data with missing labels (Wiki-90k).

ary facts into dense unary and binary facts. Experiments on two datasets with multiple ternary relations show that our proposed method can statistically significantly improve over previous works, which suggests the effectiveness of using unary and binary interaction among entities in surface patterns.

However, as Fatemi et al. (2019) suggests, there exists cases in which reconstructing $n$-ary facts from decomposed binary facts induces false positives. Tackling this issue is one important future research direction.

## References

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124. ACM.

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM*

*SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610.

Patrick Ernst, Amy Siu, and Gerhard Weikum. 2018. Highlife: Higher-arity fact harvesting. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1013–1022. International World Wide Web Conferences Steering Committee.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2019. Knowledge hypergraphs: Extending knowledge graphs beyond binary relations. *arXiv preprint arXiv:1906.00137*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369. Association for Computational Linguistics.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics.

Hong Li, Sebastian Krause, Feiyu Xu, Andrea Moro, Hans Uszkoreit, and Roberto Navigli. 2015. Improvement of n-ary relation extraction by adding lexical semantics to distant-supervision rule learning. In *Proceedings of the International Conference on Agents and Artificial Intelligence - Volume 2*, ICAART 2015, pages 317–324. SCITEPRESS - Science and Technology Publications, Lda.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 1003–1011. Association for Computational Linguistics.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182. Association for Computational Linguistics.

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

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state lstm. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235. Association for Computational Linguistics.

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

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 886–896. 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: Volume 1, Long Papers*.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated*

*Knowledge Base Construction*, AKBC '13, pages 79–84, New York, NY, USA. ACM.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics.