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

Kosuke Akimoto¹, Takuya Hiraoka¹, Kunihiko Sadamasa¹, Mathias Niepert²
² Security Research Laboratories, NEC Corporation
¹ 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 $n \ge 2$ 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 straightforward due to the sparsity of higher-order relation mentions among a specific set of $n > 2$ entities.¹ This is be-

¹In 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.

\section{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, \( \mathcal{R}_{KB} \) be a set of relation types of an external knowledge base \( KB \), and \( \mathcal{O}_{KB} = \{(r,e_1,\ldots,e_n) : r(e_1,\ldots,e_n) \in KB, r \in \mathcal{R}_{KB}\} \) be a set of known facts in \( KB \).

\(^2\)For example, the ternary relation \textit{AwardedFor}(director, movie, award) can be decomposed into the binary relations \textit{DirectorOf}(director, movie) and \textit{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/autrg/nary-relation-extraction-decomposed.
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 $\langle r, (e_1, ..., e_n) \rangle \in \mathcal{O}$, 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 $\langle r, (e_1, ..., e_n) \rangle \in \mathcal{O}$, 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 $v(r) \in \mathbb{R}^{d_r}$ for each unary or binary relation in $\mathcal{O}_1 \cup \mathcal{O}_2$. 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 ones 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 $h^r_l \in \mathbb{R}^{d_r}$ at each token position $l$. Following recent works (Zhang et al., 2018; He et al., 2018; Lee et al., 2017), we aggregate $h^r_l$ within a phrase of entity $e_k$ to compute $v(T^{(k)})$. We use elementwise mean as aggregation function:

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

**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 $h^r_l \in \mathbb{R}^{d_r}$ at each token position $l$, and max-pool along the path to compute the relation representation:

$$v(T^{(k,l)}) = \max\{(h^r_l : l = 1, ..., L)\}. \quad (2)$$

3.3 Learning Relation Representations

We follow Verga et al. (2017) to train relation representations ($\S 3.2$). We define a score $\theta_{(r,p)}$ 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 $\langle r, p_k^\prime \rangle \notin \mathcal{O}_i$. We sample negative facts by randomly replacing entity tuple $p^*$ in the original fact by different entity tuples $p_k^\prime$.

$$L_i = \mathbb{E}_{\langle r, p^\prime \rangle \in \mathcal{O}_i \setminus \langle r, p^* \rangle \in \mathcal{O}_i} \left[ \log\left( \frac{\exp(\theta_{(r,p^*)})}{\sum_k \exp(\theta_{(r,p_k^\prime)})} \right) \right]. \quad (3)$$

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

$$v(p; r) = \sum_{r' \in V(p)} a(r', r; V(p)) v(r'), \quad (4)$$

$$a(r', r; V(p)) = \frac{\exp(v(r')^T v(r))}{\sum_{r'' \in V(p)} \exp(v(r'')^T v(r))}. \quad (5)$$

3.4 Aggregating Lower-Arity Scores

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

\[\]
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 \( r \in \mathcal{R}_{KB} \) in a more supervised manner than both \( \mathcal{L}_1 \) and \( \mathcal{L}_2 \).  

\[
\theta_{(r,e_1,\ldots,e_n)} = \sum_{k=1}^{n} w_r^{(k)} \theta_{(r^{(k)},e_k)} + \sum_{k=1}^{n} w_r^{(k,l)} \theta_{(r^{(k,l)},e_k,e_l)}. 
\]

\[
\mathcal{L}_n = E_{(r,p^+) \in \mathcal{O}_{KB}} \left[ \max(0, 1 - \theta_{(r,p^+)} + \theta_{(r,p^-)}) \right] 
\]

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.  

#### 4.2 Baselines

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. We tested two types of scoring functions, Model F and Model E, as in (Toutanova et al., 2015).  

#### 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) and 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. We tested two types of scoring functions, Model F and Model E, as in (Toutanova et al., 2015).

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

4.4 Results

Table 1 illustrates the performance of each method. 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 $L_1$, $L_2$, and $\alpha L_n$ 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. Here, $\alpha = \infty$ stands for optimizing $L_n$ instead of $L_1 + L_2 + \alpha L_n$. As shown in Figure 3, $\alpha = 1$ achieved higher performance than $\alpha = \infty$, showing that introducing lower-arity semi-supervised loss ($L_1 + L_2$) 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 $L_n$ 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-relationships 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.

Acknowledgements

We thank all the EMNLP reviewers and Daniel Andrade for their valuable comments and suggestions to improve the paper.

References


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


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