Syntactic and Semantic-driven Learning for Open Information Extraction

Jialong Tang[1,3], **Yaojie Lu**[1,3], **Hongyu Lin**[1], **Xianpei Han**[1,2,*], **Le Sun**[1,2,*],
Xinyan Xiao[4], **Hua Wu**[4]

[1]Chinese Information Processing Laboratory   [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Baidu Inc., Beijing, China
{jialong2019,yaojie2017,hongyu,xianpei,sunle}@iscas.ac.cn
{xiaoxinyan,wu_hua}@baidu.com

Abstract

One of the biggest bottlenecks in building accurate, high coverage neural open IE systems is the need for large labelled corpora. The diversity of open domain corpora and the variety of natural language expressions further exacerbate this problem. In this paper, we propose a syntactic and semantic-driven learning approach, which can learn neural open IE models without any human-labelled data by leveraging syntactic and semantic knowledge as noisier, higher-level supervisions. Specifically, we first employ syntactic patterns as data labelling functions and pretrain a base model using the generated labels. Then we propose a syntactic and semantic-driven reinforcement learning algorithm, which can effectively generalize the base model to open situations with high accuracy. Experimental results show that our approach significantly outperforms the supervised counterparts, and can even achieve competitive performance to supervised state-of-the-art (SoA) model.

1 Introduction

Open information extraction (Open IE) aims to extract open-domain textual tuples consisting of a predicate and a set of arguments from massive and heterogeneous corpora (Sekine, 2006; Banko et al., 2007). For example, a system will extract a tuple (*Parragon; operates; more than 35 markets*) from the sentence "*Parragon operates more than 35 markets and has 10 offices.*". In contrary to the traditional IE, open IE is completely domain-independent and does not require the predetermined relations.

Recently, open IE has gained much attention (Fader et al., 2011; Akbik and Löser, 2012; Mausam et al., 2012; Corro and Gemulla, 2013; Moro and Navigli, 2013; Narasimhan et al., 2016;
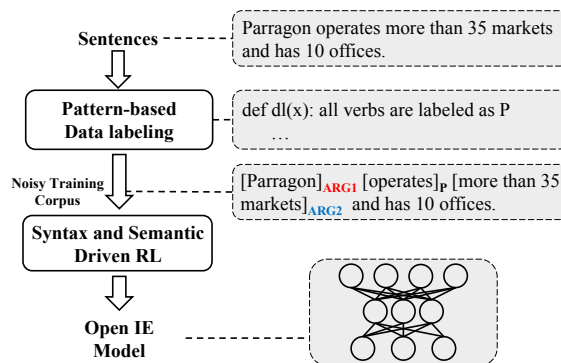

Figure 1: The proposed open IE framework, which consists of two learning strategies: 1) syntactic patterns are used as data labelling functions and a base model is pretrained using the generated labels; 2) a syntactic and semantic-driven RL algorithm is used to generalize the base model to open situations.

Pal and Mausam, 2016; Kadry and Dietz, 2017; Yu et al., 2017; Roth et al., 2018) and most of current open IE systems employ end-to-end neural networks, which first encode a sentence using Bi-LSTMs, then extract tuples by sequentially labelling all tokens in the sentence (Stanovsky et al., 2018; Jiang et al., 2019; Roy et al., 2019) or generating the target tuples token-by-token (Zhang et al., 2017; Cui et al., 2018; Sun et al., 2018). For example, to extract (*Parragon; operates; more than 35 markets*), neural open IE systems will label the sentence as [B-ARG_1, B-P, B-ARG_2, I-ARG_2, I-ARG_2, I-ARG_2, O, O, O, O, O] or generate a token sequence [$<ARG_1>$, *Parragon*, $<P>$, *operates*, $<ARG_2>$, *more, than, 35, markets*].

The neural open IE systems, unfortunately, rely on the large labelled corpus to achieve good performance, which is often expensive and labour-intensive to obtain. Furthermore, open IE needs to extract relations of unlimited types from open domain corpus, which further exacerbates the need for large labelled corpus. Therefore, the labelled
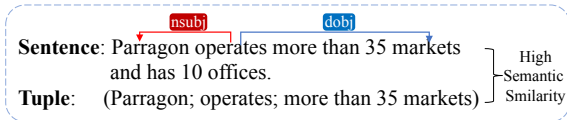
___
[*]Corresponding author

Figure 2: An example of open IE extractions, where the extracted tuple follows the *nsubj* and *dobj* dependency structure and is highly semantic similar to the original sentence.

corpus is one of the biggest bottlenecks for neural open IE systems.

To resolve the labelled data bottleneck, this paper proposes a syntactic and semantic-driven learning approach, which can learn neural open IE models without any human-labelled data by leveraging syntactic and semantic knowledge as noisier, higher-level supervisions. The motivation of our method is that, although tuple extraction is a hard task, its inverse problem – tuple assessment is easier to resolve by exploiting the syntactic regularities of relation expressions and the semantic consistency between a tuple and its original sentence. For example, Figure 2 shows the ARG1 "*Parragon*" and the ARG2 "*more than 35 markets*" follow the *nsubj* and *dobj* dependency structure, respectively. Meanwhile, the extracted tuple (*Parragon; operates; more than 35 markets*) has a high semantic similarity with its original sentence "*Parragon operates more than 35 markets and has 10 offices.*". And we found that the syntactic regularities can be effectively captured using syntactic rules, and the semantic consistency can be effectively modelled using the recent powerful pre-trained models such as BERT (Devlin et al., 2019).

Based on the above observations, we propose two learning strategies to exploit syntactic and semantic knowledge for model learning. Figure 1 illustrates the framework of our method. Firstly, syntactic open IE patterns are used as data labelling functions, and a base model is pretrained using the noisy training corpus generated by these labelling functions. Secondly, because the pattern-based labels are often noisy and with limited coverage, we further propose a reinforcement learning algorithm which uses syntactic and semantic-driven reward functions, which can effectively generalize the base model to open situations with high accuracy. These two strategies together will ensure the effective learning of open IE models: the data labelling function can pretrain a reasonable initial model so that the RL algorithm can optimize model more effec-

tively; although the pattern-based labels are often noisy and with low coverage, the RL algorithm can generalize the model to open situations with high accuracy.

We conducted experiments on three open IE benchmarks: OIE2016 (Stanovsky and Dagan, 2016), WEB and NYT (Mesquita et al., 2013). Experimental results show that the proposed framework significantly outperforms the supervised counterparts, and can even achieve competitive performance with the supervised SoA approach. [1]

The main contributions of this paper are:

- We propose a syntactic and semantic-driven learning algorithm which can leverage syntactic and semantic knowledge as noisier, higher-level supervisions and learn neural open IE models without any human-labelled data.

- We design two effective learning strategies for exploiting syntactic and semantic knowledge as supervisions: one is to use as data labelling functions and the other is to use as reward functions in RL. Experiments show that the two strategies are effective and can complement each other.

- Because labelled data bottleneck is common in NLP tasks, we believe our syntactic and semantic-driven learning algorithm can motivate the learning of other NLP models, such as event extraction, etc.

2 Syntactic and Semantic-driven Learning for Open IE

In this section, we describe how to learn neural open IE models without any human-labelled data. Two strategies are proposed to exploit syntactic and semantic knowledge as noisier, higher-level supervisions. Firstly, the syntactic patterns are used as data labelling functions for heuristically labelling a training corpus. Secondly, the syntactic and semantic coherence scores between the extracted tuples and their original sentences are used as reward functions for reinforcement learning. These two strategies together will ensure the effective learning of open IE systems: 1) although the labels generated by syntactic patterns are noisy and with limited coverage, they can pretrain a reasonable

---

783

initial model; 2) starting from the pretrained model, the syntactic and semantic-based reward functions provide an effective way to generalize our model to open situations.

In the following, we first introduce the neural networks used for open IE. Then we describe how to pretrain a base open IE model using syntactic patterns as data labelling functions. Finally, we generalize the base model using reinforcement learning with syntactic and semantic-driven rewards.

2.1 Neural Open IE Model

This paper uses RnnOIE neural networks, which have shown its simplicity and effectiveness for open IE (Stanovsky et al., 2018). But it should be noticed that our framework is not specialized to RnnOIE and can be used to train any neural open IE models.

RnnOIE formulates open IE as a sequence labelling task. Given a sentence $S = (w_1, w_2, ..., w_m)$, RnnOIE will first identify all verbs in S as predicates, such as "*operates*" and "*has*" for "*Parragon operates more than 35 markets and has 10 offices.*". For each predicate p, RnnOIE will: 1) first embed each word w_i as $x_i = [e_i; \boldsymbol{I}(w_i = p)]$, where e_i is w_i's word embedding obtained by SoA pre-trained model BERT (Devlin et al., 2019), and $\boldsymbol{I}(w_i = p)$ is an indicator vector which indicates whether w_i is p; 2) then obtain contextual word representations using a stacked BiLSTM with highway connections (Srivastava et al., 2015; Zhang et al., 2016): $\boldsymbol{H} = (h_1, h_2, ..., h_m) = BiLSTM(x_1, x_2, ..., x_m)$; 3) predict the probability of assigning label y_i to a word w_i using a fully connected feedforward classifier: $P(\hat{y}_i|\boldsymbol{S}, p, w_i) = softmax(\boldsymbol{W}h_i + \boldsymbol{b})$; 4) finally decode the full label sequence $\hat{Y}$ using a beamsearch algorithm, *e.g.*, RnnOIE will decode the label sequence [$B\text{-}ARG_1$, $B\text{-}P$, $B\text{-}ARG_2$, $I\text{-}ARG_2$, $I\text{-}ARG_2$, $I\text{-}ARG_2$, O, O, O, O, O] to extract (*Parragon; operates; more than 35 markets*).

In open IE, all extracted tuples are ranked according to their confidence scores, which is important for downstream tasks, such as QA (Fader et al., 2011) and KBP (Angeli et al., 2015). RnnOIE uses average log probabilities as the confidence of an extracted tuple:

$$c(\boldsymbol{S}, p, \hat{\boldsymbol{Y}}) = \frac{\sum_{i=1}^{m} logP(\hat{y}_i|\boldsymbol{S}, p, w_i)}{m} \quad (1)$$

Given a training corpus, RnnOIE can be supervisedly learned by maximum log-likelihood esti-
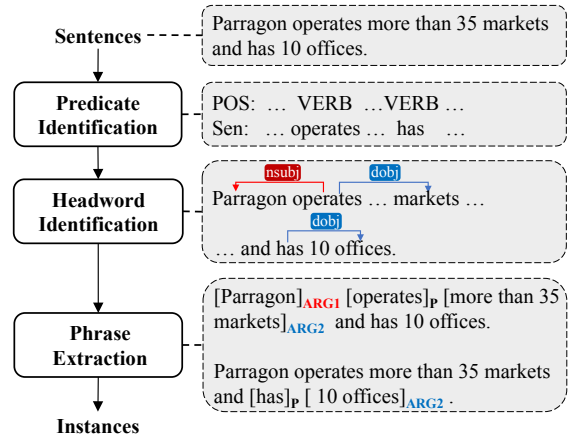

Figure 3: An overview of syntactic patterns as data labelling functions. Two training instances are automatically generated using dependency pattern for predicates "*operates*" and "*has*".

mation (MLE):

$$\log P(\mathbf{Y}|\mathbf{S}, \mathbf{p}) = \sum_{i=1}^{m} \log P(y_i|\boldsymbol{S}, p, w_i) \quad (2)$$

where $\boldsymbol{Y} = (y_1, y_2, ..., y_m)$ are the gold labels. As discussed above, $\boldsymbol{Y}$ are expensive and labour-intensive to obtain and have become the biggest bottlenecks for neural open IE systems. Therefore, it is critical to design a learning approach to get rid of this constraint.

2.2 Model Pretraining using Syntactic Pattern-based Data Labelling Functions

The first strategy is to use syntactic extraction patterns as data labelling functions, and then the heuristically labelled training corpus will be used to pretrain a neural open IE model.

It has long been observed that most relation tuples follow syntactic regularity, and many syntactic patterns have been designed for extracting tuples, such as TEXTRUNNER (Banko et al., 2007) and ReVerb (Fader et al., 2011). However, it is difficult to design high coverage syntactic patterns, although many extensions have been proposed, such as WOE (Wu and Weld, 2010), OLLIE (Mausam et al., 2012), ClausIE (Corro and Gemulla, 2013), Standford Open IE (Angeli et al., 2015), PropS (Stanovsky et al., 2016) and OpenIE4 (Mausam, 2016).

This paper leverages the power of patterns differently. Inspired by the ideas of data programming (Ratner et al., 2016) and distant supervision (Mintz et al., 2009), we use syntactic patterns

784

as data labelling functions, rather than to directly extracting tuples.

Concretely, this paper uses dependency patterns from Standford Open IE (Angeli et al., 2015) to design hand-crafted patterns as data labelling functions. As shown in Figure 3, given a sentence and its dependency parse, two training instances are generated: 1) We first identify all its predicates using part of speech (POS) tags. For example, "*operates*" and "*has*" are identified. 2) For each predicate, we identify its arguments' headwords using predefined dependency patterns [2]. For example, "*Parragon*" and "*markets*" are extracted as the headwords. 3) For each headword, we extract the whole phrase headed to it as subject/object. For example, the phrase "*more than 35 markets*" headed to "*markets*" will be extracted as the object of "*operates*".

Finally, the generated labels are used to pretrain an open IE model by optimizing the objective function (2), which can provide a reasonable initialization for starting our RL algorithm in the next section.

2.3 Model Generalization via Syntactic and Semantic-driven Reinforcement Learning

One main drawback of the automatically generated labels is that they are often noisy and with limited coverage, *i.e.*, many open relation tuples are not covered by the predefined patterns, and the dependency parse may contain errors which in turn will lead to noisy training instances. For example, in Figure 3 the training instance of the predicate "*has*" misses its subject "*Parragon*". Therefore, it is critical to generalize and refine the base model to open situations for good performance.

To this end, this section proposes the second learning strategy: syntactic and semantic-driven reinforcement learning. Specifically, we first measure the goodness of extracted tuples based on syntactic constraints using syntactic rules and semantic consistencies using pre-trained models such as BERT (Devlin et al., 2019). And then we generalize our model using the goodness of extractions as rewards in RL.

By modelling the extraction task as a Markov Decision Process (MDP), we have the following definitions: $< \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R} >$:

- $\boldsymbol{S} = \{\boldsymbol{s}\}$ are states used to capture the information from the current sentence. Specifically, $\boldsymbol{S}$ are hidden states $\boldsymbol{H}$ obtained by stacked BiLSTM.

- $\boldsymbol{A} = \{\boldsymbol{a}\}$ are actions used to indicate the target labels which are decided based on the current states $\boldsymbol{S}$ and the beam search strategy.

- $\boldsymbol{T}$ is the state transition function, which is related to the state update.

- $\boldsymbol{R}(\hat{\boldsymbol{Y}}, \boldsymbol{S})$ is the reward function, which models the goodness of the extracted tuples. We will detailly describe our syntactic and semantic-driven reward function in the next paragraph.

Formally, the open IE model is trained to maximize the expected reward of the generated label sequence $\hat{\boldsymbol{Y}}$ using the REINFORCE algorithm with likelihood ratio trick (Glynn, 1990; Williams, 1992):

$$\nabla J(\theta) = E_{\hat{\boldsymbol{Y}} \sim P(\hat{\boldsymbol{Y}}|\boldsymbol{S}, \boldsymbol{p})}[\boldsymbol{R}(\hat{\boldsymbol{Y}}, \boldsymbol{S})]$$
$$\approx \boldsymbol{R}(\hat{\boldsymbol{Y}}, \boldsymbol{S}) \nabla \log P(\hat{\boldsymbol{Y}}|\mathbf{S}, \mathbf{p}) \quad (3)$$

where $\log P(\hat{\boldsymbol{Y}}|\mathbf{S}, \mathbf{p})$ denotes the probability of the generated label sequence.

Reward Function. The reward function, i.e., the goodness of extracted tuples, is critical in our RL algorithm. This paper estimates the reward $\boldsymbol{R}(\hat{\boldsymbol{Y}}, \boldsymbol{S})$ by considering both syntactic constraint and semantic consistency:

$$\boldsymbol{R}(\hat{\boldsymbol{Y}}, \boldsymbol{S}) = Syn(\hat{\boldsymbol{Y}}) * Sem(\hat{\boldsymbol{Y}}, \boldsymbol{S}) \quad (4)$$

where $Syn(\hat{\boldsymbol{Y}})$ is the syntactic constraint score and $Sem(\hat{\boldsymbol{Y}}, \boldsymbol{S})$ is the semantic consistency score.

Following He et al. (2015); Stanovsky et al. (2018); Jiang et al. (2019), we judge an extracted tuple as correct if and only if it's predicate and arguments include their corresponding syntactic headwords (Headwords Match). Otherwise, the extracted tuples are judged as incorrect. That is:

$$Syn(\hat{\boldsymbol{Y}}) = \begin{cases} 1, & \text{Headwords Match} \\ -1, & \text{Else} \end{cases} \quad (5)$$

where 1 means the predicted label sequence $\hat{\boldsymbol{Y}}$ is correct and -1 for incorrect.

For semantic consistency, given an extracted relation and its original sentence, $Sem(\hat{\boldsymbol{Y}}, \boldsymbol{S})$ is computed as:

$$Sem(\hat{\boldsymbol{Y}}, \boldsymbol{S}) = P(positive|\hat{\boldsymbol{Y}}, \boldsymbol{S}) \quad (6)$$

---

[2]The dependency relations are defined in https://nlp.stanford.edu/software/ dependencies_manual.pdf

where $P(positive|\hat{Y}, \boldsymbol{S})$ is the semantic similarity between the predicted label sequence $\hat{Y}$ and its original sentence $\boldsymbol{S}$. This paper estimates this semantic similarity using a BERT-based classifier, which assigns a similarity score to each sentence-tuple pair. Because multiple tuples can be extracted from a single sentence (see Figure 3 for example), we train the classifier using the Stanford Natural Language Inference (SNLI) Corpus (Bowman et al., 2015), so that a high similarity score will be assigned if the original sentence entails the extracted tuple. This semantic consistency can provide useful supervision signals for open IE models. For example, because (*Parragon; has; 10 offices*) has higher semantic similarity than (*has; 10 offices*) to sentence "*Parragon operates more than 35 markets and has 10 offices.*", the model will be guided to more complete extractions.

Semantic-Based Confidence Estimation. In RnnOIE, the confidence score $c(\boldsymbol{S}, p, \hat{Y})$ is estimated only using extraction probabilities. This paper further considers the semantic consistency score for better confidence estimation:

$$c'(\boldsymbol{S}, p, \hat{Y}) = c(\boldsymbol{S}, p, \hat{Y}) + log(Sem(\hat{Y}, \boldsymbol{S}))$$
(7)

where the log is used for semantic consistency because $c(\boldsymbol{S}, p, \hat{Y})$ also uses log probabilities.

3 Experiments

3.1 Experimental Settings

Datasets. We conduct experiments on three open IE benchmarks: OIE2016 (Stanovsky and Dagan, 2016), WEB and NYT (Mesquita et al., 2013). Table 1 shows their statistics. Because only OIE2016 provides training instances and it is the largest dataset, we use OIE2016 as the primary dataset. The WEB and NYT datasets are small and without training instances, therefore we use them for out-of-domain evaluation. For OIE2016, we follow the settings in Jiang et al. (2019). For WEB and NYT, we follow the settings in Stanovsky et al. (2018).

3.2 Baselines

We compare our method with the following baselines:

- **Pattern-based open IE systems** which utilize syntactic patterns to extract relations, including *ClausIE* (Corro and Gemulla, 2013), *StandfordOpenIE* (Angeli et al., 2015),

Dataset	Type	Train	Dev	Test
OIE2016	sentence	1,688	560	641
	relation	3,040	971	1,729
WEB	sentence	–	–	500
	relation	–	–	461
NYT	sentence	–	–	222
	relation	–	–	222

Table 1: Statistics of OIE2016, WEB and NYT.

PropS (Stanovsky et al., 2016) and *OpenIE4* (Mausam, 2016).

- **Supervised neural open IE systems**, including *RnnOIE-Supervised* (Stanovsky et al., 2018) and *RankAware* (Jiang et al., 2019). *RnnOIE* is described in Section 2.1. *RankAware* is the state-of-the-art model in OIE2016 dataset, which uses iterative rank-aware learning for better confidence estimation.

3.3 Overall Results

Table 2 and Figure 4 shows the overall results. For our method, we use three settings: the first is the full model using the proposed syntactic and semantic-driven learning – *RnnOIE-Full*; the second is the base model which is not generalized using our reinforcement learning strategy – *RnnOIE-Base*; the third is our method with the base model trained using a gold-labelled corpus – *RnnOIE-SupervisedRL*. From Table 2 and Figure 4, we can see that: [3]

1) The syntactic and semantic-driven learning approach can effectively resolve the training data bottleneck of neural open IE systems. In all three datasets, *RnnOIE-Full* significantly outperforms its supervised counterpart – *RnnOIE-Supervised (BERT)*. On OIE2016, *RnnOIE-Full* can even achieve competitive performance with the supervised SoA model – *RankAware*. We believe this verifies the motivation of our method: the quality of extractions can be accurately evaluated using syntactic and semantic knowledge, and this knowledge can be effectively leveraged for the learning of open IE systems.

---

[3]The performance of *RnnOIE-Supervised* reported in our paper is lower than the original paper (Stanovsky et al., 2018) because the authors use a more lenient lexical overlap metric in their released code: https://github.com/gabrielStanovsky/oie-benchmark. Following Jiang et al. (2019), we judge an extraction as correct if the predicate and arguments include the syntactic head of the gold standard counterparts.

Model	OIE2016		WEB		NYT	
	AUC	F1	AUC	F1	AUC	F1
Pattern-based						
ClausIE (Corro and Gemulla, 2013)	2.6	14.4	11.3	24.3	1.5	6.1
StandfordOpenIE (Angeli et al., 2015)	1.7	3.2	—	—	—	—
PropS (Stanovsky et al., 2016)	0.6	6.5	0.3	4.8	0.2	3.3
OpenIE4 (Mausam, 2016)	3.4	16.4	7.1	27.7	2.3	**15.8**
Supervised Learning						
RnnOIE-Supervised (original) (Stanovsky et al., 2018)	5.0	20.4	—	—	—	—
RnnOIE-Supervised (BERT)	7.2	22.9	3.3	16.0	0.9	8.4
RankAware (Jiang et al., 2019)	12.5	31.5	—	—	—	—
Syntactic and Semantic-driven Learning						
RnnOIE-Full (Pretrained base model with RL)	13.8	**32.5**	**15.8**	**37.9**	**2.6**	14.3
RnnOIE-Base (Pretrained base model w/o RL)	5.9	24.2	10.5	31.7	1.8	10.3
RnnOIE-SupervisedRL (Supervised base model with RL)	**15.9**	32.2	11.2	29.4	**2.6**	11.4

Table 2: The overall results on OIE2016, WEB and NYT. For fair comparison, all results of baselines are adapted from their original papers except the BERT version *RnnOIE – RnnOIE-BERT*.

2) Syntactic pattern-based data labelling is an effective learning strategy. By generating training corpus, *RnnOIE-Base* achieves competitive performance on OIE2016 compared with its supervised counterpart – *RnnOIE-Supervised (BERT)* This verifies that the heuristically labelled dataset, although may noisy, can also provide a good start for building open IE systems. On the other side, we found noisy training corpus itself is not enough for high-performance open IE systems: in OIE2016 there is a 134% AUC gap (5.9 to 13.8) from *RnnOIE-Base* to *RnnOIE-Full*. This also verifies the need for further generalization techniques.

3) Syntactic and Semantic-driven RL is effective for generalize and refine open IE models. Compared with *RnnOIE-Base*, *RnnOIE-Full* can get a 134% AUC improvement, from 5.9 to 13.8. By further generalizing the supervised *RnnOIE-Supervised (BERT)* baseline using RL, *RnnOIE-SupervisedRL* can further obtain a 121% AUC improvement, from 7.2 to 15.9. The above results verify the effectiveness of our RL algorithm, and this may be because a) the RL is based on the explore-and-exploit strategy, and the explore stage can consider many unseen cases; b) the syntactic and semantic knowledge is good supervision signals for open IE systems, and the syntactic and semantic-aware rewards can effectively exploit these signals.

4) The RL-based generalization strategy is critical for scaling open IE systems to open situations. In OIE2016, we can see that, although supervised systems can outperform pattern-
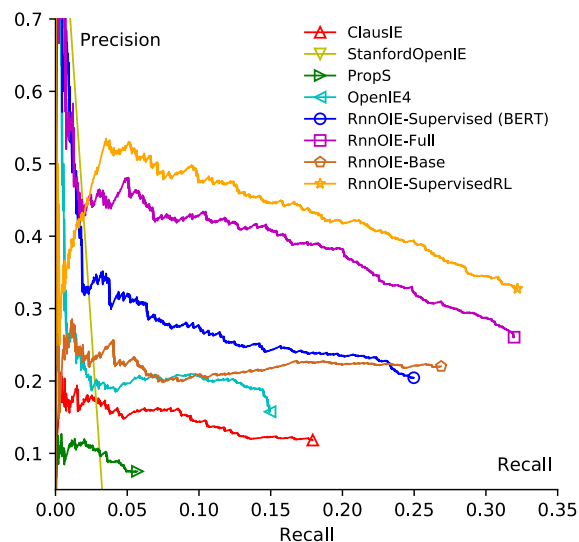

Figure 4: PR curves of different systems on OIE2016.

based systems, their performance decreases significantly in out-of-domain WEB and NYT datasets. *RnnOIE-Supervised (BERT)* even perform worse than *ClausIE* and *OpenIE4* on WEB and NYT. On the contrary, *RnnOIE-Full* can still achieve robust performance. This verifies the effectiveness of the proposed RL-based algorithm for generalizing to open situations. It is worth to notice that *RnnOIE-Full* even outperforms *RnnOIE-SupervisedRL* on out-of-domain datasets. The reason behind it may be: a) The gold-labelled corpus is useful in in-domain situations (OIE2016). However, supervised base model may be overfitting and further affects the generalization process in RL. b) *RnnOIE-Full*

	AUC	Δ_{AUC}	F1	Δ_{F1}
RnnOIE-Full	13.8		32.5	
w/o semantic	12.3	-10.9%	31.7	-2.5%
w/o syntactic	3.0	-78.3%	16.9	-48.0%

Table 3: The performance of *RnnOIE-Full* with different reward settings on OIE2016.

Confidence Estimation Algorithm	AUC	F1
Avg Log	12.0	29.1
Semanctic Consistency	10.8	**32.5**
Avg Log + Semanctic Consistency	**13.8**	**32.5**

Table 4: The performance of *RnnOIE-Full* with different confidence estimation settings on OIE2016.

learns shallow linguistic features which are more general. Therefore it performs better in out-of-domain situations (WEB and NYT).

3.4 Detailed Analysis

To analyze our method in detail, this section further investigates the effects of syntactic and semantic knowledge, semantic-based confidence estimation and RL exploration beam size.

Additionally, we compare *RnnOIE-Full* with two open IE systems to find out how far can data labelling functions get us.

Effects of Syntactic and Semantic Knowledge. Our reward function $R(\hat{Y}, S)$ consists of both syntactic constraint $Syn(\hat{Y})$ and semantic consistency $Sem(\hat{Y}, S)$. To analyze the effects of syntactic and semantic knowledge, we conduct ablation experiments by removing the semantic part (*w/o semantic*) and removing the syntactic part (*w/o syntactic*) in reward function. Table 3 shows their performances on OIE2016.

We can see that: 1) both syntactic and semantic knowledge are useful: removing any of them will result in a performance decrease; 2) syntactic constraint is crucial for our model: removing it will result in a significant AUC decrease (from 13.8 to 3.0). This is because if we drop syntactic constraint $Syn(\hat{Y})$, all explored relations will be treated as true, therefore our RL algorithm cannot rectify the wrong extractions.

Effect of Confidence Estimation. Table 4 shows the performance using different confidence estimation algorithms, including: *Avg Log* (average log probabilities) which is computed as Function 1, *Semantic Consistency* which is computed as Function 6 and *Avg Log + Semantic Consistency*.
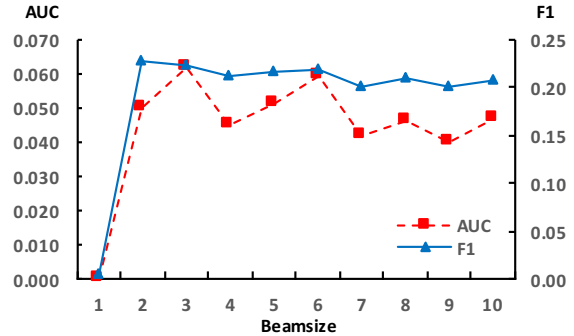

Figure 5: AUC and F1-scores of *RnnOIE-Full* with different beam sizes on the OIE2016 validation set.

Model	AUC	F1
NeuralOpenIE (Cui et al., 2018)	47.3	—
SencseOIE (Roy et al., 2019)	—	70.0
RnnOIE-Full	**56.0**	**76.7**

Table 5: The results evaluated by Lexical Overlap on OIE2016. For fair comparison, all results of baselines are adapted from their original papers.

We can see that: 1) the semantic evidence and the model prediction evidence are complementary to each other: *Avg Log + Semantic Consistency* obtains the best performance and gets 15% and 28% AUC improvements to *Avg Log* and *Semantic Consistency*; 2) *Semantic Consistency* can provide useful information for confidence estimation: *Semantic Consistency* itself can achieve comparable performance with *Avg Log*.

Effect of Beam Size. The beam size is an important hyper-parameter which controls the exploration breadth of our RL algorithm. Figure 5 shows the performance with different beam sizes.

We can see that: 1) an appropriate beam size is needed for generalizing open IE model. If the beam size is too small, *RnnOIE-Full* cannot explore new unseen cases because its explore strategy is too greedy; late2) The proposed RL algorithm is robust and achieves good performance with reasonable beam sizes (≥ 3). Because larger beam size will increase the computational complexity, we set beam size to 3 in all other experiments.

More Complex Data Labelling Functions. In section 2.2, we directly use dependency patterns from Standford Open IE (Angeli et al., 2015) to design hand-crafted patterns as data labelling functions. It raises a question that, if we use more complex patterns as data labelling functions to obtain more diverse and accurate labelled data, is it still

Error	Sentence
Missing Argument	[DePauw University]$_{ARG1}$ [awarded]$_P$ [the degree " Doctor of Divinity]$_{ARG2}$ " [<mark>in 1892</mark>]$_{Missing}$.
Overgenerated Predicate	[A British version of this show was developed , known as " Gladiators : Train 2]$_{ARG1}$ [<mark>Win</mark>]$_{Overgenerated}$ " .
Incorrect Annotation	Coke has tended to increase its control [when results were sluggish]$_{ARG2}$ in a [<mark>given</mark>]$_{Incorrect\ annotated}$ [country]$_{ARG1}$.

Table 6: Bad cases of the proposed model *RnnOIE-Full*.

necessary to move on to RL approach? To answer this question, we compare *RnnOIE-Full* with two open IE systems, NeuralOpenIE (Cui et al., 2018) and SencseOIE (Roy et al., 2019), to find out how far can data labelling functions get us.

From Table 5, we can see that: Cui et al. (2018) formulates open IE as a sentence generation task and uses OpenIE4 (Mausam, 2016) to generate train examples (AUC 47.3); Roy et al. (2019) uses three open IE systems to extract additional features to enrich human labelled train examples (F1 70.0 without other defined embedding features). Different from them, *RnnOIE-Full* does not use any labelled data and includes model generalization via RL (AUC 56.0; F1 76.7). This verifies the effectiveness and the necessity of the proposed RL-based algorithm.

3.5 Error Analysis.

We further conduct error analysis for *RnnOIE-Full*. We found there are mainly three types of error cases: *Missing Argument*, *Overgenerated Predicate* and *Wrong Annotation*. Table 6 shows their examples.

Missing Argument is the case where the extractions miss some arguments, especially for some optional arguments such as *Time* and *Place* in *RnnOIE-Full*. For instance, the first case in Table 6 shows the extraction for predicate "*award*" misses the optional time argument "*in 1892*", although it correctly contains two main arguments "*DePauw University*" and "*the degree "Doctor of Divinity*"". We found this maybe because optional arguments usually play a less important role in semantic consistency, our syntactic and semantic-driven RL algorithm will pay less attention to this generalization.

Overgenerated Predicate is the case where the predicates of extractions are not included in the ground truth. The second case in Table 6 shows a bad case where "*Win*" is wrongly extracted as the predicate. This is a common error in all neural-based approaches because they generally treat all verbs in a sentence as predicates and do not have a mechanism to reject incorrect ones. One strategy to handle this error is to jointly detect predicates and arguments, which we leave as future work.

Incorrect Annotation is the case where the ground truth labels are incorrect. Because expressions in open IE are highly diversified, we found the gold annotations may be incorrect or inconsistent. The third case in Table 6 shows an incorrect ground truth annotation "*given*", which is wrongly labelled as a predicate. This further verifies the bottleneck of high-quality, large scale labelled corpus for open IE.

4 Related Work

Open IE. Open IE approaches can be mainly categorized into two categories: pattern-based and neural-based. Pattern-based open IE approaches extract relational tuples using syntactic patterns (Banko et al., 2007; Fader et al., 2011; Wu and Weld, 2010; Mausam et al., 2012; Mausam, 2016; Corro and Gemulla, 2013; Angeli et al., 2015; Stanovsky et al., 2016); In recent years, neural-based approaches have achieved significant progress, which formulate open IE as either a sequence labelling task Stanovsky et al. (2018); Jiang et al. (2019); Roy et al. (2019) or a sentence generation task via encoder-decoder framework Cui et al. (2018); Zhang et al. (2017); Sun et al. (2018).

Syntactic and semantic knowledge has also been leveraged to enhance open IE systems. Moro and Navigli (2013) design additional syntactic and semantic features to enhance their kernel-based open IE system. Roy et al. (2019) incorporate the outputs of multiple pattern-based Open IE systems as additional features to supervised neural open IE systems to overcome the problem of insufficient. Compared with these studies which exploit syntactic and semantic knowledge as additional features of a supervised system, this paper exploits syntactic and semantic knowledge as supervision signals, so that neural open IE models can be effectively learned without any labelled data.

Data Augmentation for NLP. The labelled data bottleneck is a common problem in NLP, therefore many data augmentation techniques have been pro-

posed, such as data programming (Ratner et al., 2016), distant supervision (Mintz et al., 2009). Data programming paradigm (Ratner et al., 2016) creates training datasets by explicitly representing users' expressions or domain heuristics as a generative model. Distant supervision paradigm (Mintz et al., 2009) heuristically generates labelled dataset by aligning facts in KB with sentences in the corpus. The proposed data labelling functions are also motivated by the ideas of data programming and distant supervision.

Reinforcement Learning for IE. Reinforcement learning (RL) (Sutton and Barto, 1998) follows the explore and exploit paradigm and is apt for optimizing non-derivative learning objectives in NLP (Wu et al., 2018). Recently, RL has gained much attention in information extraction (Qin et al., 2018b,a; Takanobu et al., 2019). In open IE, Narasimhan et al. (2016) firstly using traditional Q-learning method to extract textual tuples. However, their reward function is chosen to maximize the final extraction accuracy which still relies on human-labelled datasets and can not capture the syntactic and semantic supervisions explicitly.

5 Conclusions

This paper proposes an open IE learning approach, which can learn neural models without any human-labelled data by leveraging syntactic and semantic knowledge as noisier, higher-level supervisions. Specifically, two effective learning strategies are proposed, including the pattern-based data labelling functions and the syntactic and semantic-driven RL algorithm. Experimental results show that our method significantly outperforms supervised counterparts, and can even achieve competitive performance to supervised SoA model. Furthermore, because labelled data is a common bottleneck in NLP, we believe our syntactic and semantic-driven learning approach can also be used for other NLP tasks, such as event extraction, etc.

Acknowledgments

References

Alan Akbik and Alexander Löser. 2012. KrakeN: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montréal, Canada. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Michele Banko, Michael Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *22nd International World Wide Web Conference*, page 355–366.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Peter W Glynn. 1990. Likelihood ratio gradient estimation for stochastic systems. In *Communications of the ACM*, page 75–84.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Zhengbao Jiang, Pengcheng Yin, and Graham Neubig. 2019. Improving open information extraction via iterative rank-aware learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5295–5300, Florence, Italy. Association for Computational Linguistics.

Amina Kadry and Laura Dietz. 2017. Open relation extraction for support passage retrieval: Merit and open issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1149–1152, Shinjuku, Tokyo, Japan.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 4074–4077, New York City, New York, USA. Morgan Kaufmann Publishers Inc.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Seattle, Washington, USA. 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 AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2013. Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2148–2154. Morgan Kaufmann Publishers Inc.

Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2355–2365, Austin, Texas. Association for Computational Linguistics.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, San Diego, CA. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. DSGAN: Generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Melbourne, Australia. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147, Melbourne, Australia. Association for Computational Linguistics.

Alexander Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Re. 2016. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*.

Benjamin Roth, Costanza Conforti, Porner Nina, Karn Sanjeev, and Schutze Hinrich. 2018. Neural architectures for open-type relation argument extraction. In *CoRR*, page abs/1803.01707.

Arpita Roy, Youngja Park, Lee Taesung, and Pan Shimei. 2019. Supervising unsupervised open information extraction models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pages 728–737, Hong Kong, China. Association for Computational Linguistics.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 731–738, Sydney, Australia. Association for Computational Linguistics.

Rupesh Kumar Srivastava, Klaus Greff, and Ju rgen Schmidhuber. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, page 2377–2385. Curran Associates, Inc.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages

2300–2305, Austin, Texas. Association for Computational Linguistics.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. In *CoRR*, page abs/1603.01648.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Mingming Sun, Xu Li, and Ping Li. 2018. Logician and orator: Learning from the duality between language and knowledge in open domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2119–2130, Brussels, Belgium. Association for Computational Linguistics.

R. S. Sutton and A. G Barto. 1998. *Reinforcement learning: An introduction.*

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of Association for the Advancement of Artificial Intelligence (www.aaai.org).*

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine learning*, page 229–256.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 854–864, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Sheng Zhang, Kevin Duh, and Benjamin Van Durme. 2017. MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 64–70, Valencia, Spain. Association for Computational Linguistics.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yao, Sanjeev Khudanpur, and James R. Glass. 2016. Highway long short-term memory rnns for distant speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, page 5755–5759.

792