

Improving Open Information Extraction via Iterative Rank-Aware Learning

Zhengbao Jiang, Pengcheng Yin, Graham Neubig

Language Technologies Institute

Carnegie Mellon University

{zhengbaj, pcyin, gneubig}@cs.cmu.edu

Abstract

Open information extraction (IE) is the task of extracting open-domain assertions from natural language sentences. A key step in open IE is confidence modeling, ranking the extractions based on their estimated quality to adjust precision and recall of extracted assertions. We found that the extraction likelihood, a confidence measure used by current supervised open IE systems, is not well calibrated when comparing the quality of assertions extracted from *different* sentences. We propose an additional binary classification loss to calibrate the likelihood to make it more globally comparable, and an iterative learning process, where extractions generated by the open IE model are incrementally included as training samples to help the model learn from trial and error. Experiments on OIE2016 demonstrate the effectiveness of our method.¹

1 Introduction

Open information extraction (IE, Sekine (2006); Banko et al. (2007)) aims to extract open-domain assertions represented in the form of n -tuples (e.g., *was born in; Barack Obama; Hawaii*) from natural language sentences (e.g., *Barack Obama was born in Hawaii*). Open IE started from rule-based (Fader et al., 2011) and syntax-driven systems (Mausam et al., 2012; Corro and Gemulla, 2013), and recently has used neural networks for supervised learning (Stanovsky et al., 2018; Cui et al., 2018; Sun et al., 2018; Duh et al., 2017; Jia et al., 2018).

A key step in open IE is confidence modeling, which ranks a list of candidate extractions based on their estimated quality. This is important for downstream tasks, which rely on trade-offs between the precision and recall of extracted

¹Code and data are available at https://github.com/jzbjyb/oie_rank

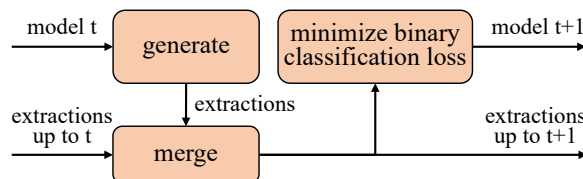


Figure 1: Iterative rank-aware learning.

assertions. For instance, an open IE-powered medical question answering (QA) system may require its assertions in higher precision (and consequently lower recall) than QA systems for other domains. For supervised open IE systems, the confidence score of an assertion is typically computed based on its extraction likelihood given by the model (Stanovsky et al., 2018; Sun et al., 2018). However, we observe that this often yields sub-optimal ranking results, with incorrect extractions of one sentence having higher likelihood than correct extractions of another sentence. We hypothesize this is due to the issue of a disconnect between training and test-time objectives. Specifically, the system is trained solely to raise likelihood of gold-standard extractions, and during training the model is not aware of its test-time behavior of ranking a set of system-generated assertions across sentences that potentially include incorrect extractions.

To calibrate open IE confidences and make them more globally comparable across different sentences, we propose an iterative rank-aware learning approach, as outlined in Fig. 1. Given extractions generated by the model as training samples, we use a binary classification loss to explicitly increase the confidences of correct extractions and decrease those of incorrect ones. Without adding additional model components, this training paradigm naturally leads to a better open IE model, whose extractions can be further included as training samples. We further propose an iter-

ative learning procedure that gradually improves the model by incrementally adding extractions to the training data. Experiments on the OIE2016 dataset (Stanovsky and Dagan, 2016) indicate that our method significantly outperforms both neural and non-neural models.

2 Neural Models for Open IE

We briefly revisit the formulation of open IE and the neural network model used in our paper.

2.1 Problem Formulation

Given sentence $s = (w_1, w_2, \dots, w_n)$, the goal of open IE is to extract assertions in the form of tuples $r = (p, a_1, a_2, \dots, a_m)$, composed of a single predicate and m arguments. Generally, these components in r need not to be contiguous, but to simplify the problem we assume they are contiguous spans of words from s and there is no overlap between them.

Methods to solve this problem have recently been formulated as sequence-to-sequence generation (Cui et al., 2018; Sun et al., 2018; Duh et al., 2017) or sequence labeling (Stanovsky et al., 2018; Jia et al., 2018). We adopt the second formulation because it is simple and can take advantage of the fact that assertions only consist of words from the sentence. Within this framework, an assertion r can be mapped to a unique BIO (Stanovsky et al., 2018) label sequence y by assigning O to the words not contained in r , B_p/I_p to the words in p , and B_{a_i}/I_{a_i} to the words in a_i respectively, depending on whether the word is at the beginning or inside of the span.

The label prediction \hat{y} is made by the model given a sentence associated with a predicate of interest (s, v) . At test time, we first identify verbs in the sentence as candidate predicates. Each sentence/predicate pair is fed to the model and extractions are generated from the label sequence.

2.2 Model Architecture and Decoding

Our training method in § 3 could potentially be used with any probabilistic open IE model, since we make no assumptions about the model and only the likelihood of the extraction is required for iterative rank-aware learning. As a concrete instantiation in our experiments, we use RnnOIE (Stanovsky et al., 2018; He et al., 2017), a stacked BiLSTM with highway connections (Zhang et al., 2016; Srivastava et al.,

2015) and recurrent dropout (Gal and Ghahramani, 2016). Input of the model is the concatenation of word embedding and another embedding indicating whether this word is predicate:

$$\mathbf{x}_t = [\mathbf{W}_{\text{emb}}(w_t), \mathbf{W}_{\text{mask}}(w_t = v)].$$

The probability of the label at each position is calculated independently using a softmax function:

$$P(y_t | \mathbf{s}, v) \propto \exp(\mathbf{W}_{\text{label}} \mathbf{h}_t + \mathbf{b}_{\text{label}}),$$

where \mathbf{h}_t is the hidden state of the last layer. At decoding time, we use the Viterbi algorithm to reject invalid label transitions (He et al., 2017), such as B_{a_2} followed by I_{a_1} .²

We use average log probability of the label sequence (Sun et al., 2018) as its confidence:³

$$c(\mathbf{s}, v, \hat{y}) = \frac{\sum_{t=1}^{|\mathbf{s}|} \log P(\hat{y}_t | \mathbf{s}, v)}{|\mathbf{s}|}. \quad (1)$$

The probability is trained with maximum likelihood estimation (MLE) of the gold extractions. This formulation lacks an explicit concept of cross-sentence comparison, and thus incorrect extractions of one sentence could have higher confidence than correct extractions of another sentence.

3 Iterative Rank-Aware Learning

In this section, we describe our proposed binary classification loss and iterative learning procedure.

3.1 Binary Classification Loss

To alleviate the problem of incomparable confidences across sentences, we propose a simple binary classification loss to calibrate confidences to be globally comparable. Given a model θ' trained with MLE, beam search is performed to generate assertions with the highest probabilities for each predicate. Assertions are annotated as either positive or negative with respect to the gold standard, and are used as training samples to minimize the hinge loss:

$$\hat{\theta} = \arg \min_{\theta} \mathbb{E}_{\substack{\mathbf{s} \in \mathcal{D} \\ v, \hat{y} \in g_{\theta'}(\mathbf{s})}} \max(0, 1 - t \cdot c_{\theta}(\mathbf{s}, v, \hat{y})), \quad (2)$$

²This formulation cannot easily handle coordination, where multiple instances of an argument are extracted for a single predicate, so we use a heuristic of keeping only the first instance of an argument.

³The log probability is normalized by the length of the sentence to avoid bias towards short sentences. The original confidence score in RnnOIE is slightly different from ours. Empirically, we found them to perform similarly.

Input: training data \mathcal{D} , initial model $\theta^{(0)}$
Output: model after convergence θ
 $t \leftarrow 0$ # iteration
 $\mathcal{E} \leftarrow \emptyset$ # generated extractions
while not converge do
 $\mathcal{E} \leftarrow \mathcal{E} \cup \{(\mathbf{s}, v, \hat{\mathbf{y}}) | v, \hat{\mathbf{y}} \in g_{\theta^{(t)}}(\mathbf{s}), \forall \mathbf{s} \in \mathcal{D}\}$
 $\theta^{(t+1)} \leftarrow \arg \min_{\theta} \mathbb{E}_{(\mathbf{s}, v, \hat{\mathbf{y}}) \in \mathcal{E}} \max(0, 1 - t \cdot c_{\theta}(\mathbf{s}, v, \hat{\mathbf{y}}))$
 $t \leftarrow t + 1$;
end

Algorithm 1: Iterative learning.

	Train	Dev.	Test
# sentence	1 688	560	641
# extraction	3 040	971	1 729

Table 1: Dataset statistics.

where \mathcal{D} is the training sentence collection, g_{θ} represents the candidate generation process, and $t \in \{1, -1\}$ is the binary annotation. $c_{\theta}(\mathbf{s}, v, \hat{\mathbf{y}})$ is the confidence score calculated by average log probability of the label sequence.

The binary classification loss distinguishes positive extractions from negative ones generated across different sentences, potentially leading to a more reliable confidence measure and better ranking performance.

3.2 Iterative Learning

Compared to using external models for confidence modeling, an advantage of the proposed method is that the base model does not change: the binary classification loss just provides additional supervision. Ideally, the resulting model after one-round of training becomes better not only at confidence modeling, but also at assertion generation, suggesting that extractions of higher quality can be added as training samples to continue this training process iteratively. The resulting iterative learning procedure (Alg. 1) incrementally includes extractions generated by the current model as training samples to optimize the binary classification loss to obtain a better model, and this procedure is continued until convergence.

4 Experiments

4.1 Experimental Settings

Dataset We use the OIE2016 dataset (Stanovsky and Dagan, 2016) to evaluate our method, which only contains verbal predicates. OIE2016 is automatically generated from the QA-SRL dataset (He et al., 2015), and to remove noise, we remove

extractions without predicates, with less than two arguments, and with multiple instances of an argument. The statistics of the resulting dataset are summarized in Tab. 1.

Evaluation Metrics We follow the evaluation metrics described by Stanovsky and Dagan (2016): area under the precision-recall curve (AUC) and F1 score. An extraction is judged as correct if the predicate and arguments include the syntactic head of the gold standard counterparts.⁴

Baselines We compare our method with both competitive neural and non-neural models, including RnnOIE (Stanovsky et al., 2018), OpenIE4,⁵ ClausIE (Corro and Gemulla, 2013), and PropS (Stanovsky et al., 2016).

Implementation Details Our implementation is based on AllenNLP (Gardner et al., 2018) by adding binary classification loss function on the implementation of RnnOIE.⁶ The network consists of 4 BiLSTM layers (2 forward and 2 backward) with 64-dimensional hidden units. ELMo (Peters et al., 2018) is used to map words into contextualized embeddings, which are concatenated with a 100-dimensional predicate indicator embedding. The recurrent dropout probability is set to 0.1. Adadelta (Zeiler, 2012) with $\epsilon = 10^{-6}$ and $\rho = 0.95$ and mini-batches of size 80 are used to optimize the parameters. Beam search size is 5.

4.2 Evaluation Results

Tab. 4 lists the evaluation results. Our base model (RnnOIE, § 2) performs better than non-neural systems, confirming the advantage of supervised training under the sequence labeling setting. To test if the binary classification loss (E.q. 2, § 3) could yield better-calibrated confidence, we perform one round of fine-tuning of the base model with the hinge loss (+**Binary loss** in Tab. 4). We show both the results of using the confidence (E.q. 1) of the fine-tuned model to rerank the extractions of the base model (**Rerank Only**), and the end-to-end performance of the fine-tuned model in assertion generation (**Generate**). We

⁴The absolute performance reported in our paper is much lower than the original paper because the authors use a more lenient lexical overlap metric in their released code: <https://github.com/gabrielStanovsky/oie-benchmark>.

⁵<https://github.com/dair-iitd/OpenIE-standalone>

⁶<https://allennlp.org/models#open-information-extraction>

sentence	old rank	new rank	label
A CEN forms an important but small part of a Local Strategic Partnership.	3	1	✓
An animal that cares for its young but shows no other sociality traits is said to be “subsocial”.	2	2	✗
A casting director at the time told Scott that he had wished that he’d met him a week before; he was casting for the “G.I. Joe” cartoon.	1	3	✗

Table 2: Case study of reranking effectiveness. Red for predicate and blue for arguments.

sentence	label
A Democrat, he became the youngest mayor in Pittsburgh’s history in September 2006 at the age of 26.	✓
A motorcycle speedway long-track meeting, one of the few held in the UK, was staged at Ammanford.	✗

Table 3: Case study of generation effectiveness. Red for predicate and blue for arguments.

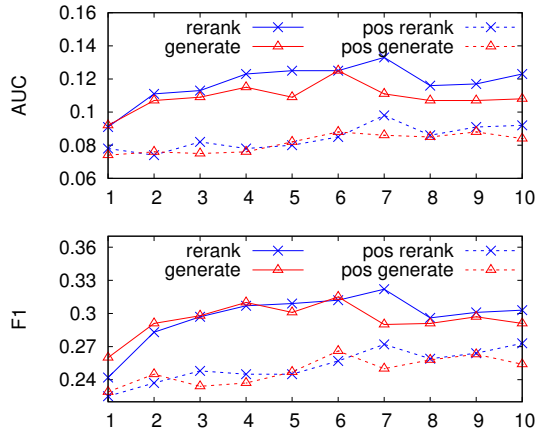


Figure 2: AUC and F1 at different iterations.

found both settings lead to improved performance compared to the base model, which demonstrates that calibrating confidence using binary classification loss can improve the performance of both reranking and assertion generation. Finally, our proposed iterative learning approach (Alg. 1, § 3) significantly outperforms non-iterative settings.

We also investigate the performance of our iterative learning algorithm with respect to the number of iterations in Fig. 2. The model obtained at each iteration is used to both rerank the extractions generated by the previous model and generate new extractions. We also report results of using only positive samples for optimization. We observe the AUC and F1 of both reranking and generation increases simultaneously for the first 6 iterations and converges after that, which demonstrates the effec-

System	AUC	F1
<i>Non-neural Systems</i>		
PropS	.006	.065
ClausIE	.026	.144
OpenIE4	.034	.164
<i>Neural Systems</i>		
Base Model (RnnOIE)	.050	.204
+Binary loss (§ 3.1), Rerank Only	.091	.225
+Binary loss (§ 3.1), Generate	.092	.260
+Iterative Learning (§ 3.2)	.125	.315

Table 4: AUC and F1 on OIE2016.

tiveness of iterative training. The best performing iteration achieves AUC of 0.125 and F1 of 0.315, outperforming all the baselines by a large margin. Meanwhile, using both positive and negative samples consistently outperforms only using positive samples, which indicates the necessity of exposure to the errors made by the system.

Case Study Tab. 2 compares extractions from RnnOIE before and after reranking. We can see the order is consistent with the annotation after reranking, showing the additional loss function’s efficacy in calibrating the confidences; this is particularly common in extractions with long arguments. Tab. 3 shows a positive extraction discovered after iterative training (first example), and a wrong extraction that disappears (second example), which shows that the model also becomes better at assertion generation.

overgenerated predicate	wrong argument	missing argument
41%	38%	21%

Table 5: Proportions of three errors.

Error Analysis Why is the performance still relatively low? We randomly sample 50 extractions generated at the best performing iteration and conduct an error analysis to answer this question. To count as a correct extraction, the number and order of the arguments should be exactly the same as the ground truth and syntactic heads must be included, which is challenging considering that the OIE2016 dataset has complex syntactic structures and multiple arguments per predicate.

We classify the errors into three categories and summarize their proportions in Tab. 5. “Overgenerated predicate” is where predicates not included in ground truth are overgenerated, because all the verbs are used as candidate predicates. An effective mechanism should be designed to reject useless candidates. “Wrong argument” is where extracted arguments do not coincide with ground truth, which is mainly caused by merging multiple arguments in ground truth into one. “Missing argument” is where the model fails to recognize arguments. These two errors usually happen when the structure of the sentence is complicated and coreference is involved. More linguistic information should be introduced to solve these problems.

5 Conclusion

We propose a binary classification loss function to calibrate confidences in open IE. Iteratively optimizing the loss function enables the model to incrementally learn from trial and error, yielding substantial improvement. An error analysis is performed to shed light on possible future directions.

Acknowledgements

This work was supported in part by gifts from Bosch Research, and the Carnegie Bosch Institute.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. [Open information extraction from the web](#). In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676.

Luciano Del Corro and Rainer Gemulla. 2013. [Clausie: clause-based open information extraction](#). In *22nd International World Wide Web Conference*, pages 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*, pages 407–413.

Kevin Duh, Benjamin Van Durme, and Sheng Zhang. 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*, pages 64–70.

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.

Yarin Gal and Zoubin Ghahramani. 2016. [A theoretically grounded application of dropout in recurrent neural networks](#). In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 1019–1027.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. [Allennlp: A deep semantic natural language processing platform](#). *CoRR*, abs/1803.07640.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. [Deep semantic role labeling: What works and what’s next](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 473–483.

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.

Shengbin Jia, Yang Xiang, and Xiaojun Chen. 2018. [Supervised neural models revitalize the open relation extraction](#). *CoRR*, abs/1809.09408.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

- Satoshi Sekine. 2006. [On-demand information extraction](#). In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. [Training very deep networks](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 2377–2385.
- 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.
- Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. 2016. [Getting more out of syntax with props](#). *CoRR*, 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*, pages 885–895.
- Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018. [Logician: A unified end-to-end neural approach for open-domain information extraction](#). In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 556–564.
- Matthew D. Zeiler. 2012. [ADADELTA: an adaptive learning rate method](#). *CoRR*, abs/1212.5701.
- 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 *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5755–5759.