# Coarse-to-Fine Open Information Extraction via Relation Oriented Reading Comprehension

Tingxin Li* Rui Meng† Feng Chen‡ Jianming Wu§

**Abstract**

Open information extraction (Open IE), aiming at distilling structured, machine-readable triples from natural language text, plays an important role in various applications, including natural language understanding, knowledge graph construction, etc. Previous supervised Open IE approaches are mostly tailored to extract predicate-argument triples, in which the predicate is usually limited to verb phrases, whereas, the semantic relations expressed within noun phrases are being neglected. However, identifying semantic relation between entities is no trivial task due to the implicit and complex relation expressions. To address the above issue, we present ReadOIE, a framework for coarse-to-fine Open IE via relation oriented reading comprehension, to extract relation-entity triples. In our framework, all entity pairs are extracted to generate structured questions and the input sentence is regarded as the context passage. Semantic relations that best answer the questions are then extracted by comprehending the given context. Moreover, in order to identify the non-existence relations between entities, we design a coarse-to-fine relation extraction approach consisted of an extensive detection module and an intensive extraction module. The extensive detection performs relation existence judgement on a coarse level and intensive extraction identifies the relation on a fine-grained level. Extensive experiments on benchmark datasets demonstrate that ReadOIE outperforms the state-of-the-art baselines.

**Keywords**

Open Information Extraction, Machine Reading Comprehension, Relation Extraction

## 1 Introduction

Open information extraction (Open IE) aims to represent unstructured natural language text into a structured triple format. For example, given the sentence "Katy, graduated from Blue University, works for Green Office now.", Open IE systems would extract the following two triples: *(Katy, graduated from , Blue University)* and *(Katy, works for, Green Office)*. Different from traditional information extraction, which adopts supervised methods for schema-guided entity and relation extraction, Open IE does not rely on any handcrafted ontology input and can be applied to different corpus without domain-specific fine tuning. Open IE is an important and fundamental task, which can generate useful intermediate representations for downstream natural language processing (NLP) applications, including question answering [15], knowledge graph construction [27], and event schema induction [16].

| Input Sentence | Barack Obama was the President of the United States. |
|---|---|
| Part-of-Speech | Barack/NNP Obama/NNP was/VBD the/DT President/NNP of/IN the/DT United/NNP States/NNPS. |
| Predicate-Argument Triple | Barack Obama (ARG1) \| was (Predicate) \| the President of the United States. (ARG2) |
| Relation-Entity Triple | Barack Obama (Entity (PER)) \| was the President of (Relation) \| the United States. (Entity (LOC)) |

Figure 1: Comparison between predicate-argument and relation-entity triple structures.

Recent studies have demonstrated the superiority of supervised Open IE models using deep neural networks, e.g., RnnOIE [21], SpanOIE [29], Multi2OIE [19], over traditional statistical or rule-based approaches [7, 3, 25]. The supervised neural network based solutions usually transform the Open IE task as a sequence labeling or a span detection problem and train models on large-scale gold benchmark corpus for triple extraction.

Despite remarkable progress in supervised Open IE, most existing systems are intrinsically designed for extracting predicate-argument triples in Semantic Role Labeling (SRL) structure, in which arguments are noun chunks and predicates are verb chunks indicating the syntactic relation between arguments. On the one hand, the benchmark datasets, e.g., OIE2016[1], OntoNotes5.0[2] and OpenIE4[3], adopted in existing systems follow the SRL scheme, i.e., who did what to

*Cornell University, tl687@cornell.edu.
†Corresponding author, BNU-HKBU United International College, ruimeng@uic.edu.cn.
‡The University of Adelaide, chenfeng1271@gmail.com.
§Guangzhou University, jianmingwu@e.gzhu.edu.cn.

[1]https://github.com/gabrielStanovsky/oie-benchmark
[2]https://catalog.ldc.upenn.edu/LDC2013T19
[3]https://github.com/allenai/openie-standalone

whom at when/where, for triple annotation. On the other hand, existing systems adopt two-stage pipeline framework for triple extraction, in which the predicate is firstly extracted and arguments are subsequently detected based on the given text as well as extracted predicates. In such framework, open relation/predicate extraction, which is the most challenging task in Open IE [23], is performed by either simply identifying all verbs as predicate candidates or conducting word-level classification based on token syntactic and contextual features. Consequently, the extracted predicates are usually limited to verb phrases, whereas, the highly implicit relational expressions within noun phrases are being neglected, hereby degrading information extraction performance. In contrast, another triple structure, i.e., relation-entity triple, which represents semantic relation that holds between two entities mentioned in the text, is preferred and more widely adopted in many downstream NLP and knowledge graph applications [13, 30]. Figure 1 presents the difference between two triple structures. Given the sentence "Barack Obama was the president of the United States", the extracted predicate-argument triple is (Barack Obama, was, the president of the United States). However, the verb "was" is not expressive enough to indicate the relation information conveyed in the sentence. Instead, the relation-entity triple, (Barack Obama, was the president of, the United States), is more powerful for encoding the semantic information in a structured manner.

In this paper, we present an Open IE system for extracting relation-entity triples from natural language text. In contrast to previous approaches, which extract all predicates first and then identify arguments for each predicate, we propose to perform relation extraction after entity identification, due to the implicit and complex expression of open relations. Specifically, triples are extracted using a pipeline of named entity recognition and relation extraction. Name entity recognition (NER) has been extensively investigated and a considerable number of deep learning based NER models [18, 22, 4] have gained great success in advancing the state-of-the-art performance. Whereas, relation identification and extraction in Open IE remains an unsolved thorny challenge. Firstly, semantic relation expressions are complex, diverse and implicit. Therefore, extracting such relations requires a deeper comprehension of both the input text and related entities. Moreover, there exists many entities in a sentence and not every entity pair necessarily entail semantic relationship. Therefore, given a pair of entities, the relation extraction task need to not only extract the relation, if exists, but also be able to detect the non-existence of relation to avoid unnecessary error-prone extractions.

To address the aforementioned challenges, we propose ReadOIE, a coarse-to-fine Open IE framework for relation-entity triple extraction. In our framework, entities are firstly detected using existing NER toolkit and relation extraction is then performed for each entity pair in a coarse-to-fine style. Specifically, we formulate triple extraction as an relation oriented reading comprehension task, in which the question is constructed using recognized entity pairs and the corresponding answer is extracted as our target relation. In order to identify the non-existence relation as well, our relation extraction component adopts a novel two-stage strategy: extensive relation detection, which performs relation existence judgement on a coarse level, and intensive relation extraction to perform relation identification in the fine-grained stage. We train our model on OPIEC [11], an Open IE corpus with relation-entity triple annotations. At inference stage, triples are generated and ranked by combing the outputs of the two modules. Our contributions can be summarized as follows:

- We propose ReadOIE, a coarse-to-fine open information extraction framework for extracting relation-entity triples from natural language text. The triple extraction task is formulated as a relation oriented reading comprehension problem, which is capable of capturing the implicit and complex semantic relation expressions in the given text.
- We present a novel two-stage strategy for relation extraction given entity pairs, i.e., coarse relation existence judgement and specific relation extraction, to avoid extracting error-prone relations for entities without semantic relationships.
- We conduct extensive experiments on several Open IE benchmark datasets. Experimental results verify that our proposed ReadOIE outperforms the state-of-the-art supervised Open IE systems. We also open-sourced the code[4] of our system for reproducing experiments and future research.

## 2 Related Work

In recent years, many Open IE systems have been proposed. The methods can roughly be classified into two categories.

Rule-based approaches for OpenIE make use of information extraction rules, which are either handcrafted or learned in an automatic or semi-automatic manner. ReVerb [7] introduces two simple but surprisingly powerful syntactic and lexical constraints on how binary relationships are expressed via verbs. It first

[4] https://github.com/tingxinli1/ReadOIE

identifies relation phrases that satisfy constraints, and then finds a pair of NP arguments for each identified relation phrase. EXEMPLAR [5] adopts hand-crafted patterns based on dependency parse trees to extract triples in Semantic Role Labeling structure, in which a relation trigger is first detected and the arguments connected to it are then identified. OLLIE [17] designs a system for learning a set of extraction pattern templates based on dependency parse paths for triple extraction. A set of high precision seed tuples were adopted at the beginning and open pattern templates over the training set are learned through bootstrap learning. ReNoun [26] is an Open IE system proposed to extract noun-mediated relations. Dependency parse based patterns are learned using distant supervision for noun-based relations. PropS [8] introduces a strategy of transforming a dependency parse tree into a directed graph, a structure designed to represent the proposition structure of a sentence, by a rule-based converter. Predicate-argument triples are then extracted directly from the graph structure. To improve the Open IE performance on complex sentences, ClauseIE [3] and Stanford Open IE [1] are proposed to firstly detect clauses in a sentence and triples are extracted for each clause based on dependency parsing. Although rule-based approaches are transparent and expressive, it is time-consuming to create high-quality rules manually and the rules inherently cannot generalize well to new corpus.

Motivated by the availability of large-scale Open IE benchmark datasets and the great success of neural network based models on various NLP tasks, another category approaches, supervised Open IE systems using deep neural networks, have become popular. RNNOIE [21] formulates the Open IE task as a sequence tagging problem. Verbs and nominal predicates in the sentence are first identified. BiLSTM is then adopted to capture sentence and predicate head context and triples are extracted using predicted BIO tagging labels. SenseOIE [20] proposes to employ an ensemble of multiple unsupervised Open IE extraction results and other syntactic and lexical features for triple extraction using sequence tagging method. In SpanOIE [29], BiLSTM is deployed to obtain the span context representation based on various features, including POS tag, word embedding and dependency relationship. Predicate and argument are extracted based on the span selection model. Multi$^2$OIE [19] presents a system for performing Open IE by using Pre-trained Language Model (PLM) as sentence encoder. It first finds all predicates in the sentence and then extracts the arguments associated with each identified predicate. DeepStruct [24] proposes a method for improving structural understanding abilities of language models by conducting structure pre-training. The

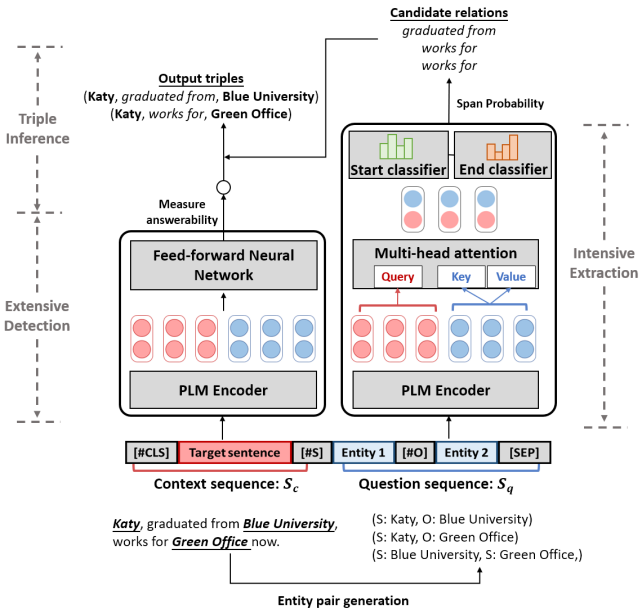


Figure 2: Architecture of ReadOIE.

language models can learn to correspond to structures in various tasks, including Open IE. Our work belongs to the second category. In contrast to existing works, we aim to design an end-to-end system to perform relation-entity triple extraction by firstly identify entities in a sentence and then extract semantic relations between candidate entity pairs.

## 3 Methodology

### 3.1 Taks formulation

In this work, we present an Open IE framework for extracting relation-entity triples. The Open IE problem could be formally defined as follows: given a textual corpus, the target is to extract a set of *(subject, relation, object)* triples, in which the *subject* and *object* are entities corresponding to real-world objects and *relation* denotes the semantic relation between the two entities. Specifically, we formulate the triple extraction task as a relation-oriented machine reading comprehension (MRC) problem. We first extract all entities in each sentence to generate candidate entity pairs and detect the semantic relation between each entity pair subsequently by reading and comprehending the given sentence.

The architecture of ReadOIE is illustrated in Figure 2. The entities in each sentence can be firstly detected using NER techniques [18, 22, 4]. In our work, we adopt Spacy[5] for entity extraction. For each pair of entities, our target is to detect whether there ex-

[5]https://spacy.io/

ists some semantic relation and extract the relation, if necessary. To assist relation extraction, we construct a structured question for each entity pair and the input sentence is regarded as the context passage. Then, the context passage and structured question is concatenated as our input for the relation-oriented MRC task, denoted as $\langle S_c, S_q(e_s, e_o)\rangle$, where $S_c$ and $S_q$ are context passage and structured question sequence, respectively. Note that the relative order of entities in original sentence is reserved in the structured question.

Our framework consists of two parallel modules, i.e., an extensive relation detection module and an intensive relation extraction module. The extensive detection module detects whether a pair of entities are connected by some semantic relationship. The relation detection is performed by predicting the answerability score $\hat{y}^c$ of the structured question. Then the intensive extraction module locates the relation connecting the entity pair mentioned in the query using span selection, i.e., $[\hat{pos}^s, \hat{pos}^e]$. The span model predicts the probabilities of token at position $p$ being relation span start and end, i.e., $\hat{p}^s$ and $\hat{p}^e$. Candidate triples are generated by combining the outputs of two modules. These candidates will be ranked according to relation existence probability, which is calculated by extensive detection module and top $k$ triples with highest confidence score are returned as the result.

### 3.2 Structured Question Generator

We aim to generate appropriate question sequence based on extracted entities to assist relation extraction in the relation-oriented reading comprehension problem. For each sentence input, we enumerate all entity pairs to construct structured questions by filling entities into slots led by special tokens `[#S]` and `[#O]` (abbreviation of subject and object). Each structured question, $S_q$, is denoted as "`[#S]` $ent_1$ `[#O]` $ent_2$", in which $ent_1$ and $ent_2$ represent candidate subject and object, respectively. In contrast to adopting the default separation token `[SEP]` to mark the end of a sequence, we indicate the role of two entities, i.e., subject or object, in each triple explicitly by using special tokens `[#S]` and `[#O]`, to help the model capture entity pair-specific semantic relation. For example, given the sentence "Katy, graduated from Blue University, works for Green Office now.", we can extract three entities: *Katy*, *Blue University* and *Green Office*. For the entity pair *Katy* and *Blue University*, a structured question "`[#S]` *Katy* `[#O]` *Blue University*" will be generated.

Then, we concatenate the context sequence $S_c$ with the structured question sequence $S_q$ using the separator token `[#SEP]`. Besides, the classifier token `[#CLS]` is added at the beginning of the concatenated sequence. The context-question pair sequence is denoted as "`[#CLS]` *target sentence* `[#S]` $ent_1$ `[#O]` $ent_2$ `[SEP]`". Following pretained language model (PLM) based MRC systems [28, 10], the context and question are encoded with different segmentation embeddings. By adopting the the structured question, our model can comprehend the target sentence and learn entity-aware contextual embedding for subsequent relation identification and extraction.

### 3.3 Extensive Detection Module

Given the context and question pair $\langle S_c, S_q(e_s, e_o)\rangle$, extensive detection module reads the target sentence in a sketchy way and gives a coarse judgement of the relation existence between the two entities in the structured question. The relation existence detection problem is formulated as a classification task, in which the result is positive if the two entities in structured question are connected by some semantic relation.

For each sentence in the training corpus, we enumerate all the candidate entity pairs by combination and match them with annotated triples. The matched ones will be labeled as positive samples while the unmatched ones are negative. Due to the large amount of negative pairs, we adopt downsampling strategy to balance the samples for better convergence. The output score of the classifier will be normalized for producing the classifier probability of relation existence. To be specific, the input is a concatenated token sequence of target sentence $S_c = \{s_1, s_2, ..., s_m\}$ and structured question $S_q = \{s_{m+1}, s_{m+2}, ..., s_n\}$, containing a candidate subject-object pair. The output is a probability $\hat{y}^c$, indicating the confidence that there exists some semantic relation between the two entities. To represent the input information, we adopt a pretrained language model (PLM) based on multi-layer Transformer [6] as the encoder to generate the entity-aware contextual embedding:

$$\mathbf{H} = \text{PLM}(S_{context}, S_{question}) \tag{3.1}$$

where $\mathbf{H} = \{h_1, h_2, ..., h_n\}$ is the last hidden state output by the encoder. Each vector $h_i$ denotes the representation of corresponding token $s_i$. Then, the embedding matrix is passed to a fully connected layer to obtain the relation existence probability $\hat{y}^c$:

$$\hat{y}^c = \text{Sigmoid}(\text{FFN}(\mathbf{H})) \tag{3.2}$$

where $\text{FFN}(\cdot)$ denotes feed-forward network. The predicted relation existence probability will be adopted as the triple confidence score for triple selection.

### 3.4 Intensive Extraction Module

The intensive extraction module will learn to extract a

phrase in the target sentence, which indicates the relation between given subject and object entity pairs, to answer the structured question. Similar to the extensive detection module, we use a pretrained language model (PLM) based on multi-layer Transformer [6] as the encoder to obtain contextual representation of target sentence tokens. For each context and question pair $\langle S_c, S_q(e_s, e_o)\rangle$, we use $\mathbf{H} = \{h_1, h_2, ..., h_n\}$ to denote the last hidden state output by the encoder.

**Entity-aware Contextual Learning.** Unlike extensive detection module, which only makes a coarse judgement of the relation existence, intensive extraction module needs to be more concentrated in relation span identification. Although the PLM encoder can learn entity-aware contextual embeddings by encoding context and question together, we propose to perform multi-head cross attention between context passage and question on top of the PLM encoder to further strengthen the attention on entity pair while encoding the context passage. Such cross attention can improve the target sentence comprehension concerning the structured question. Specifically, we split the target context-structured question contextual embedding matrix $\mathbf{H}$ (3.1) into two parts: embedding of structured question $\mathbf{H}_{question}$ and embedding of target sentence $\mathbf{H}_{context}$. In the cross attention component, we regard the target sentence as our query and the structured question as key and value, i.e., $\mathbf{Q} = \mathbf{H}_{context}$, $\mathbf{K} = \mathbf{H}_{question}$, and $\mathbf{V} = \mathbf{H}_{question}$. Then, the multi-head cross attention will be performed to obtain entity-aware contextual representation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{3.3}$$

$$\mathbf{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{3.4}$$

$$\mathbf{H}' = \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{3.5}$$

$$= \text{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O \tag{3.6}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$, and $\mathbf{W}^O$ are trainable parameter matrices; $\mathbf{H}'$ is the output entity-aware contextual representation of target sentence.

**Relation Span Selection.** Recent works has shown the superiority of span selection over sequential tagging on phrase extraction tasks [9]. Therefore, we adopt a span selection model for relation extraction. Each span, denoted as $s = (i, j)$, is a subsequence of the sentence, characterized by span head and tail word indices $i$ and $j$. In our model, we adopt two classifiers on top of the multi-head cross attention component to predict the probability that each token being the start or end of the relation span, respectively:

$$\hat{\mathbf{y}}^s = \text{SoftMax}(\text{FFN}_\text{s}(\mathbf{H}')) \tag{3.7}$$

$$\hat{\mathbf{y}}^e = \text{SoftMax}(\text{FFN}_\text{e}(\text{Concat}(\hat{\mathbf{y}}^s, \mathbf{H}'))) \tag{3.8}$$

where $\hat{\mathbf{y}}^s, \hat{\mathbf{y}}^e \in \mathbb{R}^L$, denote the probability distribution of each position being the start/end of relation span, and $L$ is the length of the target sentence. Note that the input for end classifier is a concatenation of $\mathbf{y}^s$ and $\mathbf{H}'$, so that the end position prediction is conditioned on the start position prediction result. The end classifier is trained with gold relation span starts, while the start classifier result will be adopted during the inference stage. For span selection, we take the positions with the largest start probability and end probability as the span start and span end, respectively:

$$\hat{pos}^b = \text{argmax}(\hat{\mathbf{y}}^b) \tag{3.9}$$

where $\hat{pos}^b$, $b \in \{s, e\}$ stands for the start (or end) position of the predicted relation span.

### 3.5 Parallel Training

In the training stage, two modules are trained in a parallel style. The reason is that semantically related entity pairs are very few compared with unrelated entity pairs, especially for long sentences. Therefore, it would be challenging for the intensive extraction module to learn appropriately if there are too many training samples with no annotated relation span. Through parallel training, unrelated entity pairs are only included in the training samples of extensive detection module, so that two modules can achieve better convergence by focusing on different training samples from different perspectives.

The extensive detection module is trained as a binary classifier. The loss is defined as follows:

$$\mathcal{L}_0 = -\frac{1}{N}\sum_{i=1}^{N} \text{BCE}(y_i^c, \hat{y}_i^c) \tag{3.10}$$

where BCE denotes binary cross entropy; $N$ is the number of training examples; $\hat{y}_i^c$ and $y_i^c$ are the predicted score and ground truth, respectively.

For intensive extraction module, the start and end classifier loss functions are:

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N} \text{BCE}(\hat{\mathbf{y}}_{i,j}^s, \mathbf{y}_{i,j}^s) \tag{3.11}$$

where $\hat{\mathbf{y}}_{i,j}^b$ and $\mathbf{y}_{i,j}^b$, $b \in \{s, e\}$, denote the predicted start (or end) probability of position $j$ in $i^{th}$ sample and the ground truth.

We jointly optimize the start and end classifiers by using the sum of start and end classifiers' loss as intensive extraction module optimization objective:

$$\mathcal{L}_1 = \mathcal{L}_s + \mathcal{L}_e \tag{3.12}$$

### 3.6 Triple Inference

In the inference stage, two modules work independently for predicting relation existence and extracting candidate relation span, for each entity pair extracted from a sentence. Then, candidate *(subject, relation, object)* triples are generated by combining the entity pairs with corresponding intensive relation extraction results. The relation existence probability calculated by extensive detection module is adopted as the triple confidence score. Finally, candidate triples are ranked by the confidence score and top k triples are returned as the result.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** Since our work aims at extracting relation-entity triples, we adopt benchmark datasets annotated with relation-entity triple structures in our experiments. We adopt OPIEC-clean [11] as our training and development set. OPIEC is an Open IE corpus constructed from English Wikipedia. The clean version of the dataset contains only sentences whose arguments are clean noun phrases. As the full dataset is too large (292.4 GB), we use the released official samples in our experiments. Besides, we filtered out incomplete triples, i.e., triples lack of relation, subject or object, to refine the dataset. For fair comparisons with other Open IE systems, following [21, 24], we adopt the benchmark datasets, i.e., WEB, NYT, PENN, as our testing sets. Considering the fact that testing sets and training/development set are from different domains, the performance on testing sets can demonstrate the model generalization capability. The descriptive statistics of our experimental datasets are summarized in Table 1.

Table 1: Dataset Statistics

| Dataset | Usage | # Sentences |
|---|---|---|
| OPIEC | Train/Dev | 11835 |
| NYT | Test | 150 |
| WEB | Test | 52 |
| PENN | Test | 461 |

**Baselines.** To show the superiority of ReadOIE, we compare our proposed model with top-performing Open IE systems, which cover both rule based models and deep learning based end-to-end models. Rule based models include ClausIE [3], Open IE4 [2], and PropS [8]. Deep learning models include RnnOIE [21], SpanOIE [29], Multi$^2$OIE [19], and DeepStruct [24]. Note that for DeepStruct, we include both multi-task pretrained version and fine-tuned version since they perform differently on various datasets.

**Evaluation.** We follow the automatic evaluation API using lexical match between predicted extractions and ground truth extractions[21]. Such method measures the extraction quality using soft matching based on words instead of hard matching based on phrases, which gives more fair results as exact correct extractions are hard to achieve in Open IE task. We adopt the area under the curve (AUC) and F1-score (F1) to evaluate the overall performance.

**Setting.** For training extensive detection module, due to the large amount of negative pairs, we downsample the negative pairs to improve the sample balance to 1 positive to 10 negatives for better convergence. We adopt BERT-base-uncased[6] as the PLM encoder for both modules. For hyperparameters, we set the output embedding dimension of cross attention to 768 and set the number of attention heads as 12. When constructing inputs, we pad the noun phrases in the structured questions to max length 5 and pad the target sentence to max length 150. In the training stage, we use AdamW [14] as the optimizer. We set the learning rate to $1 \times 10^{-5}$ and set a linear learning rate scheduler with a warm up stage during half of the first epoch. We train the model for maximum 10 epochs and set the early stop patience as 2 to avoid overfitting. We set $K = 1$ when selecting top candidates during the ranking stage.

### 4.2 Experimental Results

The experimental results of comparative study are shown in Table 2. We report the AUC and F1 of different models on WEB, NYT, and PENN. For systems CluaseIE, OpenIE4, PropS and RnnOIE, we use the results[7] reported by RnnOIE [21] directly. We use the official codes released by authors for SpanOIE[8] and Multi$^2$OIE[9]. For DeepStruct, as the codes have not been release yet and AUC results are not reported in their work, we only show the F1 scores given in [24].

Overall, our model shows significant improvement across all three datasets in terms of F1. Specifically, compared with previous best model, our model increases the F1 by 2.5%, 7.8%, and 12.8%, respectively. For AUC, our system outperforms best existing baseline in NYT and PENN by 3.8% and 23.4%, respectively. On average, our model achieves 10.9% improvement in F1 and 6.3% improvement in AUC. The comparative study shows the superiority of the relation-oriented supervision compared with previous supervised models, which firstly detect predicate and then extract arguments for each predicate, including RnnOIE, SpanOIE, and Multi$^2$OIE. Note that the Deep-

[6] https://github.com/google-research/bert
[7] https://github.com/gabrielStanovsky/supervised-oie
[8] https://github.com/zhanjunlang/Span_OIE
[9] https://github.com/youngbin-ro/Multi2OIE

Table 2: Experimental results of comparative study. The best results among all models are displayed in bold and the second-bests are indicated by underline.

| Dataset | WEB | | NYT | | PENN | | Average | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **F1** | **AUC** | **F1** | **AUC** | **F1** | **AUC** | **F1** | **AUC** |
| ClausIE [3] | 44.9 | 40.0 | 29.6 | 23.0 | 34.6 | 28.4 | 36.4 | 30.5 |
| Open IE4 [2] | 55.7 | 40.5 | 38.3 | 24.0 | 42.6 | 28.1 | 45.5 | 30.9 |
| PropS [8] | 58.3 | 46.8 | 37.2 | 22.1 | 39.1 | 27.7 | 44.9 | 32.2 |
| RnnOIE [21] | 66.8 | 46.9 | 35.3 | <u>25.0</u> | 44.1 | 25.7 | 48.7 | 32.5 |
| SpanOIE [29] | 65.1 | <u>54.1</u> | 36.3 | 17.3 | 40.0 | 20.8 | 47.1 | 30.7 |
| Multi$^2$OIE [19] | <u>67.9</u> | **62.0** | 40.3 | 23.1 | 49.6 | <u>34.0</u> | <u>52.6</u> | <u>39.7</u> |
| DeepStruct [24] (multi-task) | 50.8 | - | 43.6 | - | <u>54.5</u> | - | 49.6 | - |
| DeepStruct [24] (finetune) | 49.1 | - | <u>45.0</u> | - | 45.1 | - | 46.4 | - |
| **ReadOIE** (ours) | **70.4** | 51.9 | **52.8** | **28.8** | **67.3** | **57.4** | **63.5** | **46.0** |

Struct (multi-task) model is pretrained on a mixture dataset, in which datasets from multiple tasks are integrated and DeepStruct (fine-tune) model is fine tuned on OIE2016 dataset, which is a benchmark dataset for Open IE annotated by predicate-argument triple structures. The DeepStruct performance decreased after fine tuning, which indicates that the supervision provided by commonly adopted Open IE datasets, annotated with predicate-argument triple structure, tends to degrade the model capability of extracting relation-entity triples.

In our framework, we propose to adopt structured question constructed by entity pair in reading comprehension procedure for relation existence classification. We concatenate the context sequence with structured question sequence together as the input for PLM encoder in order to learn entity-aware contextual embedding to assist relation existence detection performance. To figure out whether our model can capture entity-aware contextual features, we select a typical sample from our test set and visualize the attention weights of token `[#CLS]` in Figure 3. We can observe that by adopting special tokens `[#S]` and `[#O]` in the structured question, the PLM encoder can better capture the entity pair information while encoding the context passage. Specifically, most of the information flowing to token `[#CLS]` comes from the question sequence, i.e., entities, `[#S]` and `[#O]`, which verifies that our model can capture the entity-aware contextual representations for relation existence detection.

### 4.3 Case Study

Compared with existing Open IE systems, our framework shows superiority in extracting relation-entity triples. Specifically, existing works are mostly trained on datasets annotated with predicate-argument triple structure and follow the pipeline of predicate identification and argument extraction. Such systems are

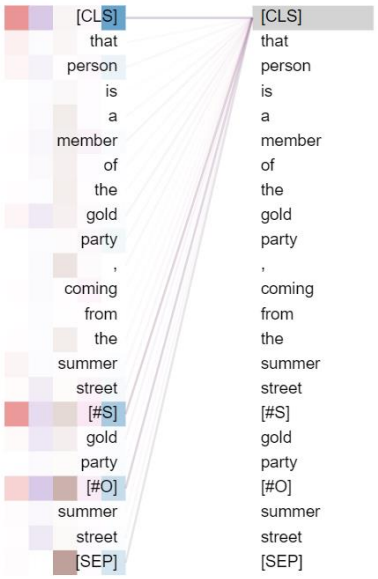


Figure 3: Visualization of attention weights.

prone to extract predicate-argument triples, in which the extracted predicates are usually limited to verb phrases, whereas, the highly implicit relational expressions within noun phrases can not be captured well. We demonstrate the extraction examples by different approaches in Table 3. Two sentences from WEB corpus and triple with highest confidence score of SpanOIE, Multi$^2$OIE and our framework are presented.

According to the extraction results, we can observe that existing systems tend to extract verb or verb phrases as the predicate, e.g., *has been*, *received* and *was*, whereas, our system can capture the relations expressed using noun phrases, e.g., *lifelong resident of* in the first sentence sample. Moreover, existing methods tend to extract arguments consisted of noun chunks connected by prepositions. Some predicate may take compound arguments with long sequences, e.g.,

Table 3: Example of extractions from SpanOIE, Multi$^2$OIE and our framework ReadOIE

| | |
|---|---|
| Sentences | 1. Professional tennis player, **Andre Kirk Agassi**$_{sub}$ has been a lifelong **resident**$_{rel}$ of **Las Vegas**$_{obj}$.<br>2. Although she was instrumental in the discovery of the structure of DNA, **Crick**$_{sub}$ and Watson are credited with the discovery and **received**$_{rel}$ a **Nobel Prize**$_{obj}$ for the achievement. |
| SpanOIE | 1. (*Professional tennis player, Andre Kirk Agassi*; **has been**; *a lifelong resident of Las Vegas*)<br>2. (*the achievement*; **received**; *a Nobel Prize*) |
| Multi$^2$OIE | 1. (*Professional tennis player Kirk Agassi*; **has been**; *a lifelong resident of Las Vegas*)<br>2. (*she*; **was**; *instrumental in the discovery of the structure of DNA*) |
| **ReadOIE** | 1. (*Andre Kirk Agassi*; **has been a lifelong resident of**; *Las Vegas*)<br>2. (*Crick*; **received**; *Nobel Prize*) |

*instrumental in the discovery of the structure of DNA* in Multi$^2$OIE. In contrast, our system extracts entities expressed in a more concise and clean form, such as *Andre Kirk* and *Nobel Prize*.

### 4.4 Ablation Study

We conduct ablation study to further demonstrate the contribution of individual model components. Our main focus is to explore (1) whether adopting structured question in relation-oriented reading comprehension would improve the performance; (2) and whether the extensive detection module is helpful to filter out unrelated entity pairs and refine the extraction results.

Therefore, we first follow a commonly used relation extraction approach [12], which simply append the entity pairs at the end of the target sentences using default separating token "`[#SEP]`" without using different segmentation encoding (denoted as "-Struct Quest"). Through such approach, the model will not take the inputs as a context-question pair sequence. Moreover, we also test our model without using the extensive detection module and directly use the probabilities obtained by the intensive extraction module to select final triples (denoted as "-Extensive"). The experimental results of F1-score are shown in Table 4.

Table 4: Experimental results of ablation study

| **Model** | **WEB** ($\Delta$) | **NYT** ($\Delta$) | **PENN** ($\Delta$) |
|---|---|---|---|
| All | 70.4 | 52.8 | 67.3 |
| -Struct Quest | 66.5 (-3.9) | 45.5 (-7.3) | 53.8 (-13.5) |
| -Extensive | 56.4 (-14.0) | 44.8 (-8.0) | 40.4 (-26.9) |

According to the ablation study, we can observe that without adopting structured question would results in 3.9%, 7.3% and 13.5% drop in F1-score on three datasets. The results demonstrate that the entity-aware contextual features captured by applying structured question is important for triple extraction. Without extensive extraction module, the F1-scores decrease dramatically by 14.0%, 8.0%, and 26.9%, respectively, which indicates that extensive detection is of great importance to improve the extraction performance.

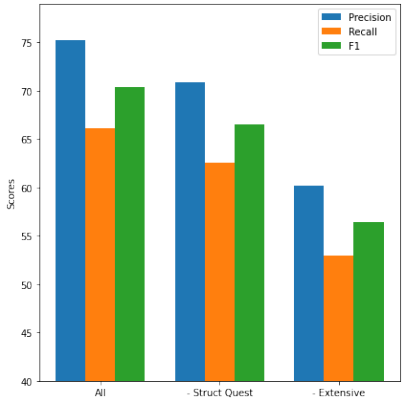


Figure 4: Performance comparison of model variants on WEB dataset.

Moreover, we show the detailed performance comparison results, i.e., precision, recall and F1 score, of model variants on WEB dataset in Figure 4. According to the detailed results, the precision and recall decreased by 5% and 3% if the structured question component is removed; and the absence of extensive detection module would degrade the precision and recall by 15% and 13%, respectively. The results verify that the two components adopted in our framework contribute to the improvement in both precision and recall, especially for the extensive detection module. The module can successfully filter our non-related entity pairs and the participant of coarse judgement of relation existence is crucial to more accurate triple extraction.

## 5 Conclusion

In this paper, we propose ReadOIE, a coarse-to-fine Open IE framework for extracting relation-entity triples from natural language text via relation-oriented reading comprehension. Entities are first recognized and structured question are constructed from entity pairs. Then, relations will be extracted by reading and com-

prehending the sentence. Our framework contains two modules: an extensive detection module, which aims to detect whether there exists relation between two entities in the structured question, and an intensive extraction module, which extracts the candidate relation span that best answers the question. Then, triples are extracted by combining the outputs of two modules. Extensive experiments on benchmark datasets demonstrate that our proposed framework achieves significant improvement over state-of-the-art baselines.

## References


[1] G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *ACL*, pages 344–354, 2015.

[2] J. Christensen, Mausam, S. Soderland, and O. Etzioni. An analysis of open information extraction based on semantic role labeling. In *K-CAP*, pages 113–120, 2011.

[3] L. D. Corro and R. Gemulla. Clausie: clause-based open information extraction. In *WWW*, pages 355–366, 2013.

[4] S. S. S. Das, A. Katiyar, R. J. Passonneau, and R. Zhang. Container: Few-shot named entity recognition via contrastive learning. In *ACL*, pages 6338–6353, 2022.

[5] F. de Sá Mesquita, J. Schmidek, and D. Barbosa. Effectiveness and efficiency of open relation extraction. In *EMNLP*, pages 447–457, 2013.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[7] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545, 2011.

[8] T. Falke, G. Stanovsky, I. Gurevych, and I. Dagan. Porting an open information extraction system from english to german. In *EMNLP*, pages 892–898, 2016.

[9] J. Fu, X. Huang, and P. Liu. SpanNER: Named entity re-/recognition as span prediction. In *ACL*, pages 7183–7195, 2021.

[10] L. Gao, Y. Wang, T. Liu, J. Wang, L. Zhang, and J. Liao. Question-driven span labeling model for aspect-opinion pair extraction. In *AAAI*, pages 12875–12883, 2021.

[11] K. Gashteovski, S. Wanner, S. Hertling, S. Broscheit, and R. Gemulla. OPIEC: an open information extraction corpus. In *AKBC*, 2019.

[12] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li. A unified MRC framework for named entity recognition. In *ACL*, pages 5849–5859, 2020.

[13] X. Lin, H. Li, H. Xin, Z. Li, and L. Chen. Kbpearl: A knowledge base population system supported by joint entity and relation linking. *Proc. VLDB Endow.*, 13(7):1035–1049, 2020.

[14] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*, 2019.

[15] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, and S. Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*, pages 8449–8456, 2020.

[16] Mausam. Open information extraction systems and downstream applications. In *IJCAI*, pages 4074–4077, 2016.

[17] Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni. Open language learning for information extraction. In *EMNLP-CoNLL*, pages 523–534, 2012.

[18] S. Mayhew, N. Gupta, and D. Roth. Robust named entity recognition with truecasing pretraining. In *AAAI*, pages 8480–8487, 2020.

[19] Y. Ro, Y. Lee, and P. Kang. Multi$^2$oie: Multilingual open information extraction based on multi-head attention with BERT. In *EMNLP*, pages 1107–1117, 2020.

[20] A. Roy, Y. Park, T. Lee, and S. Pan. Supervising unsupervised open information extraction models. In *EMNLP-IJCNLP*, pages 728–737, 2019.

[21] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan. Supervised open information extraction. In *NAACL-HLT*, pages 885–895, 2018.

[22] J. Wan, D. Ru, W. Zhang, and Y. Yu. Nested named entity recognition with span-level graphs. In *ACL*, pages 892–903, 2022.

[23] C. Wang, X. He, and A. Zhou. Open relation extraction for chinese noun phrases. *IEEE Trans. Knowl. Data Eng.*, 33(6):2693–2708, 2021.

[24] C. Wang, X. Liu, Z. Chen, H. Hong, J. Tang, and D. Song. Deepstruct: Pretraining of language models for structure prediction. In *ACL (Findings)*, pages 803–823, 2022.

[25] A. S. White, D. A. Reisinger, K. Sakaguchi, T. Vieira, S. Zhang, R. Rudinger, K. Rawlins, and B. V. Durme. Universal decompositional semantics on universal dependencies. In *EMNLP*, pages 1713–1723, 2016.

[26] M. Yahya, S. Whang, R. Gupta, and A. Y. Halevy. Renoun: Fact extraction for nominal attributes. In *EMNLP*, pages 325–335, 2014.

[27] H. Yu, N. Zhang, S. Deng, H. Ye, W. Zhang, and H. Chen. Bridging text and knowledge with multi-prototype embedding for few-shot relational triple extraction. In *COLING*, pages 6399–6410, 2020.

[28] F. Yuan, L. Shou, X. Bai, M. Gong, Y. Liang, N. Duan, Y. Fu, and D. Jiang. Enhancing answer boundary detection for multilingual machine reading comprehension. In *ACL*, pages 925–934, 2020.

[29] J. Zhan and H. Zhao. Span model for open information extraction on accurate corpus. In *AAAI*, pages 9523–9530, 2020.

[30] S. Zhou, B. Yu, A. Sun, C. Long, J. Li, and J. Sun. A survey on neural open information extraction: Current status and future directions. In *IJCAI*, pages 5694–5701, 2022.