Maximal Clique Based Non-Autoregressive Open Information Extraction

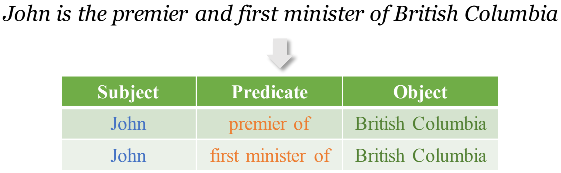
Bowen Yu[1,2*], **Yucheng Wang**[1,2*], **Tingwen Liu**[1,2†],
Hongsong Zhu[1,2], **Limin Sun**[1,2], **Bin Wang**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Xiaomi AI Lab, Xiaomi Inc., Beijing, China
{yubowen,wangyucheng,liutingwen}@iie.ac.cn
{zhuhongsong,sunlimin}@iie.ac.cn wangbin11@xiaomi.com

Abstract

Open Information Extraction (OpenIE) aims to discover textual facts from a given sentence. In essence, the facts contained in plain text are unordered. However, the popular OpenIE systems usually output facts sequentially in the way of predicting the next fact conditioned on the previous decoded ones, which enforce an unnecessary order on the facts and involve the error accumulation between autoregressive steps. To break this bottleneck, we propose MacroIE, a novel non-autoregressive framework for OpenIE. MacroIE firstly constructs a fact graph based on the table filling scheme, in which each node denotes a fact element, and an edge links two nodes that belong to the same fact. Then OpenIE can be reformulated as a non-parametric process of finding maximal cliques from the graph. It directly outputs the final set of facts in one go, thus getting rid of the burden of predicting fact order, as well as the error propagation between facts. Experiments conducted on two benchmark datasets show that our proposed model significantly outperforms current state-of-the-art methods, beats the previous systems by as much as 5.7 absolute gain in F1 score.

1 Introduction

Open Information Extraction (OpenIE) aims to convert natural text to semi-structured knowledge, by mining facts in the form of n-ary tuples $r =$ (subject, predicate, object$_1$, $\cdots$, object$_m$), composed of a single subject and predicate as well as m objects. While traditional IE systems require people to pre-specify the set of interested relations and provide per-relation training data, OpenIE is built on the principles of ontology-free (Tang et al., 2020), making it possible to adapt to various domains and applications, such as extending knowledge bases (Dong et al., 2014), and facilitating question answering systems (Fader et al., 2014).



John is the premier and first minister of British Columbia

Subject	Predicate	Object
John	premier of	British Columbia
John	first minister of	British Columbia

Figure 1: An example of OpenIE in a sentence.

Starting from rule-based systems to neural networks, OpenIE has attracted increasing attention in recent years but remains challenging (Niklaus et al., 2018), due to the intrinsic difficulty in identifying complicated facts, including: (1) Overlapping, one fact element (either subject, predicate, or object) may belong to multiple facts in a sentence. For example, in Figure 1, the entity pairs of two facts are identical but the predicates are different; (2) Discontinuous, one fact element can consist of spans that are separated by intervals, as the predicate of the first fact in Figure 1 comprising two spans *premier* and *of*; (3) Nested, one fact element could contain other elements or share words with other elements. We can see that the two predicates in Figure 1 containing the same word *of*.

Recent studies on systematically handling these challenges have two major research lines: Tagging, and Generation. Tagging-based system (Kolluru et al., 2020a) annotates M different tag sequences corresponding to M facts in the input sentence. To avoid redundant extraction, the labels of one tag sequence are passed to the next iteration to decode another sequence. Generation-based system (Kolluru et al., 2020b) directly decodes the facts as a sequential output by generating one word at a time through the Seq2Seq architecture. Either tagging or generation paradigm predicts facts auto-regressively, which means the current fact prediction relies on the previous output. Despite their success, all of these methods are still limited by the autoregressive prediction process. The reasons are as follows: they enforce an unnecessary order on the facts dur-

[*] The two authors contribute equally.
[†] Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9696–9706
November 7–11, 2021.

ing the training phase, while other fact orders are also correct. In essence, the facts contained in a sentence have no intrinsic order (Sui et al., 2020). Moreover, the models predict conditioned on the previously generated ones, thus a skewed prediction will be inherited and magnified in the later steps (Zhang et al., 2020). As the number of steps grows, i.e., multi-fact extraction, the errors accumulate and may decrease the performance.

In this paper, we break the autoregressive factorization by presenting a novel view of OpenIE as a maximal clique discovery task. In graph theory, cliques refer to subgraphs in a graph such that nodes in each subgraph are pairwise adjacent. Moreover, a maximal clique is a clique that cannot be extended by including one more adjacent node. That means the nodes in a maximal clique have close connections with each other, which is similar to the relationship between elements in a fact. Armed with this insight, we reach an intuition that OpenIE can be cast as finding the maximal cliques from a fact graph (see also Figure 2), in which each node denotes a span (a continuous fact element on its own, or a part of discontinuous elements), and an edge links two nodes belonging to the same fact.

We implement the above idea in **Ma**ximal **c**lique discove**r**y based **o**pen **I**nformation **E**xtractor (MacroIE), a non-autoregressive end-to-end OpenIE framework. It constructs the fact graph with two independent steps: node extraction and edge prediction, and tackles them together with a unified table filling scheme to accurately recognize the discontinuous or nested fact elements. Then an accompanied decoding algorithm is developed to recover desired facts from the fact graph, which has the elegance to extract overlapping facts by design. Owing to the novel task formulation, MacroIE can predict all facts at once, without having to cope with the fact order and the error propagation between facts, thus overcoming the aforementioned limitations of autoregressive methods.

We conduct experiments on two realistic OpenIE benchmarks in English and Chinese, respectively. Experimental results show that the proposed MacroIE model significantly outperforms existing best-performing methods, with substantial gains of up to 5.7% absolute percentage points in F1 score, establishing a new state of the art for this task. Furthermore, detailed analysis shows that MacroIE gains consistent improvement in complicated fact extraction and multiple fact extraction.

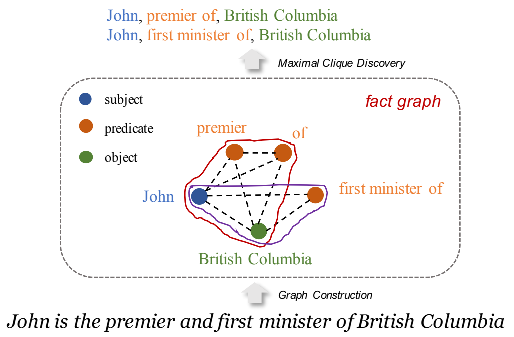

John, premier of, British Columbia
John, first minister of, British Columbia
Maximal Clique Discovery
subject
predicate
object
fact graph
premier
of
John
first minister of
British Columbia
Graph Construction
John is the premier and first minister of British Columbia

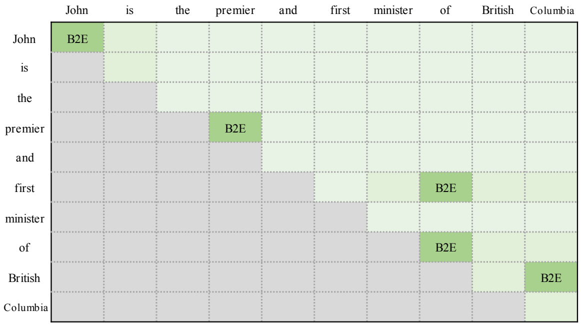
Figure 2: An example of extracting facts as maximal clique discovery. The solid lines of different colors circle different maximal cliques in the fact graph.

2 Methodology

2.1 Task Formulation

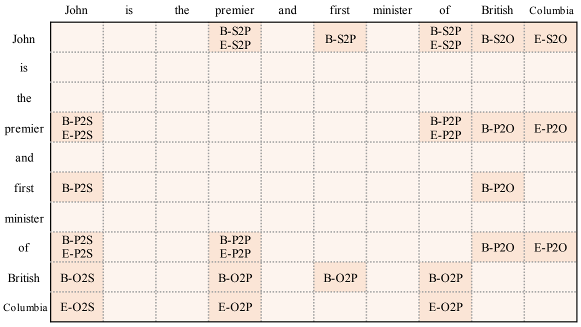
Given a sentence $\mathrm{S} = \{w_1, w_2, \cdots, w_n\}$ where w_i denotes the i-th token, the task of OpenIE is to output a set of facts, say $\{r_1, r_2, \cdots, r_M\}$. For illustration purposes, we assume the facts are binary, e.g., $r =$ (subject, predicate, object) in this section, and we will also demonstrate how to extend our method to n-ary fact extraction in Section 3.2.

The heart of our proposed MacroIE model is reformulating OpenIE as a maximal clique discovery task. A clique, C, in an undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ is a subset of the nodes, such that every two distinct nodes are adjacent. A maximum clique of G is a clique that does not exist exclusively within the node set of a larger clique. That is, all the nodes in one maximum clique are pairwise connected, and adding any other node of G to this clique will break the current balance. This property is similar to the relationship between elements in one fact. As shown in Figure 2, if we regard each continuous span of each fact element as one node in the fact graph, and connect all spans belonging to the same fact with specific roles, then OpenIE is essentially equivalent to finding the maximal clique on the constructed graph, and each maximal clique corresponds to a fact. Moreover, maximal clique discovery has been extensively studied in graph theory, and several classic algorithms such as Bron–Kerbosch (Bron and Kerbosch, 1973) can efficiently list all maximal cliques in polynomial time per clique, so the remaining question to our task formulation is, how to construct a fact graph. MacroIE decomposes it as two uncoupled tasks: span extraction and edge prediction. As the name suggests, the former focuses on extracting fact spans as nodes V, while the latter is responsible for creating edges E.



	John	is	the	premier	and	first	minister	of	British	Columbia
John	B2E									
is										
the										
premier				B2E						
and										
first								B2E		
minister										
of								B2E		
British										B2E
Columbia										

Figure 3: A tagging example for span extraction.



	John	is	the	premier	and	first	minister	of	British	Columbia
John				B-S2P E-S2P		B-S2P		B-S2P E-S2P	B-S2O	E-S2O
is										
the										
premier	B-P2S E-P2S							B-P2P E-P2P	B-P2O	E-P2O
and										
first	B-P2S								B-P2O	
minister										
of	B-P2S E-P2S			B-P2P E-P2P					B-P2O	E-P2O
British	B-O2S			B-O2P		B-O2P		B-O2P		
Columbia	E-O2S			E-O2P				E-O2P		

Figure 4: A tagging example for edge prediction.

2.2 Table Filling Scheme

In MacroIE, both span extraction and edge prediction can be tackled from a table filling perspective. Formally, given sentence S, the table filling scheme maintains an $n \times n$ tag table to represent a set of semantic relations such that the (i, j)-th cell denotes the relationship (or non-relation) between tokens w_i and w_j. Next, we elaborate on how this structure allows an elegant formalization of span extraction (SE) and edge prediction (EP).

2.2.1 Span Extraction

Each node in our fact graph represents a continuous span involved in one fact asserted by the input sentence. As demonstrated in Figure 2, different spans may be nested, e.g., *first minister of* and *of* both serve as nodes. To make our extractor capable of handling this situation, we construct a two-dimensional tag table, which determines whether each pair of tokens in the sentence is the boundary of a fact span with a B2E (beginning-end) tag. Since different spans do not share the same boundary pair, our tagging scheme can naturally solve the difficulty of expressing nested spans. Figure 3 illustrates an example, in which token pair (*first*, *of*) and (*of*, *of*) are both assigned with B2E, thus *first minister of* and *of* can be simultaneously identified. Note that only the upper triangular table is necessary for indicating the boundary relations, so the number of cells to be labeled is $\frac{n(n+1)}{2}$.

2.2.2 Edge Prediction

The goal of edge prediction is to connect the nodes in the fact graph and signify their roles in respective facts with edge types. One intuitive solution is: firstly enumerating all possible span pairs extracted from the SE step and then classifying the relations between each of these pairs. While being easy to implement, this process is vulnerable to errors cascading down the pipeline. To decouple the dependency between SE and EP, we propose to distinguish and align the boundary tokens of span pairs from scratch with a two-dimensional tag table. The tag set for EP is defined as: {B-S2S, B-S2P, B-S2O, B-P2S, B-P2P, B-P2O, B-O2S, B-O2P, B-O2O, E-S2S, E-S2P, E-S2O, E-P2S, E-P2P, E-P2O, E-O2S, E-O2P, E-O2O}. Each tag in which is constituted of two parts: position and role. The former indicates whether the two corresponding positions in the table are the beginning (B) of two spans that belong to the same fact, or the end (E). And the latter encodes the role relationship of two span in their involved facts, including subject (S), predicate (P), and object (O). For example, *John* and *British Columbia* play the role of subject and object respectively in the facts expressed in Figure 4, thus the tag of (*John*, *British*) is B-S2O, while E-O2S is labeled at the place of (*Columbia*, *John*) in the table. Notice that the same pair of spans can have different role relationships, so each cell in the edge prediction table may be assigned with multiple tags.

One may wonder why we choose to annotate the role of span in EP instead of SE. One span could play different roles in different facts of the same sentence. Let us consider the sentence *Jone visited Beijing, this is the capital of China*, it contains two facts: (*Jone*, *visited*, *Beijing*) and (*Beijing*, *capital of*, *China*), where *Beijing* serves as object and subject in them respectively. Our scheme can ideally deal with this overlapping situation by determining the role of one span according to its outgoing edge types in the fact clique (lines 12-19 in Algorithm 1).

2.3 Model Architecture

With our table filling scheme, we build an end-to-end neural architecture MacroIE (Figure 5) to jointly extract nodes and predict edges. Our architecture first encodes the n-token sentence to

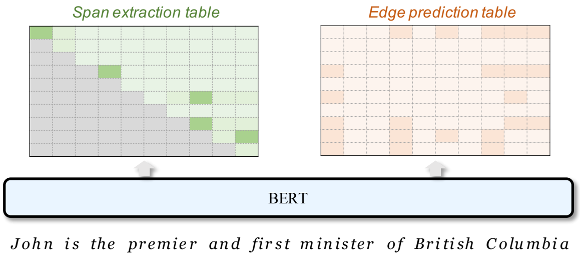


Figure 5: Model Architecture.

produce contextualized token embedding sequence $[\mathbf{h}_1, \cdots, \mathbf{h}_n]$ with pre-trained language models such as BERT (Devlin et al., 2019). Then we can generate a representation $\mathbf{h}_{i,j}$ for the token pair (w_i, w_j) as follows:

$$\mathbf{h}^{\mathcal{I}}_{i,j} = \mathbf{W}^{\mathcal{I}}_a[\mathbf{h}_i; \mathbf{h}_j; \mathbf{h}_i \odot \mathbf{h}_j; |\mathbf{h}_i - \mathbf{h}_j|] + \mathbf{b}^{\mathcal{I}}_a, \quad (1)$$

where $[;]$ is the vector concatenation, $\odot$ is the element-wise multiplication. $\mathbf{W}^{\mathcal{I}}_a$ is a weight matrix and $\mathbf{b}^{\mathcal{I}}_a$ is a bias vector to be learned during training, $\mathcal{I} \in$ {SE, EP}is the subtask indicator. Then, we feed $\mathbf{h}^{\mathcal{I}}_{i,j}$ into a fully-connected layer, which is followed by a Sigmoid function to compute label probability:

$$P(y^{\mathcal{I}}_{i,j}) = \text{Sigmoid}(\mathbf{W}^{\mathcal{I}}_b \mathbf{h}^{\mathcal{I}}_{i,j} + \mathbf{b}^{\mathcal{I}}_b). \quad (2)$$

By learning different table filling parameters for SE and EP, we can generate different $P(y^{\mathcal{I}}_{i,j}) \in \mathbb{R}^{N^{\mathcal{I}}}$, where $N^{\mathcal{I}}$ is the number of possible tags in $\mathcal{I}$. Each dimension of $P(y^{\mathcal{I}}_{i,j})$ denotes the probability of a tag between w_i and w_j. Then the tag set of (x_i, x_j) can be predicted as:

$$\text{tag}^{\mathcal{I}}_{i,j} = \{k | P(y^{\mathcal{I}}_{i,j} = k) > \eta\}, \quad (3)$$

where $P(y^{\mathcal{I}}_{i,j} = k)$ represents the probability of assigning the tag k to the place of (i, j) in the table of subtask $\mathcal{I}$. η is the threshold that converts $P(y^{\mathcal{I}}_{i,j})$ to tags. We enumerate several values in $(0, 1)$ and pick the one that maximizes the evaluation metrics on the development set as the threshold. During training, we minimize the negative log-likelihood of $P(y^{\mathcal{I}}_{i,j})$ over the correct tags with the binary cross-entropy loss. The losses from the tag table of SE and EP are aggregated as the training objective.

OpenIE models typically assign a confidence value to an predicted fact (Kolluru et al., 2020a). In MacroIE, each fact is assigned a confidence value by summing the log probabilities of the nodes and edges in the respective clique and normalizing this by the number of edges and nodes.

Algorithm 1 Overall workflow

```
Input: Sentence S = $\{w_1, w_2, \cdots, w_n\}$
Output: The fact set expressed in S, denoted as F.
 1: Fill the SE table $\mathrm{T}_s$ and EP table $\mathrm{T}_e$ with Equation 3
 2: Decode $\mathrm{T}_s$ to obtain the span set P
 3: Initialize the fact graph G with P
 4: for span $v \in$ P do
 5:     for another span $g \in$ P do
 6:         if B-* $\in$ $\mathrm{T}_e$($v$.begin, $g$.begin) & E-* $\in$ $\mathrm{T}_e$($v$.end, $v$.end) then
 7:             Connect $v$ and $g$ in G
 8:         end if
 9:     end for
10: end for
11: Find the maximal cliques C in G with Algorithm ??
12: for clique $c \in$ C do
13:     for span $v \in c$ do
14:         Initialize the role list of $v$ with $\varnothing$, denoted as $\mathrm{R}_v$
15:         for another span $g \in c$ do
16:             Add the outgoing role part of each tag in $\mathrm{T}_e$($v$.begin, $g$.begin) and $\mathrm{T}_e$($v$.end, $g$.end) to $\mathrm{R}_v$
17:         end for
18:         Select the most frequent role type in $\mathrm{R}_v$ as the role of $v$ in the clique $c$
19:     end for
20:     Merge the spans of the same role type with their order in S as the fact element.
21:     Assemble elements to constitute a fact and add it to F
22: end for
23: return F
```

2.4 Workflow

In this subsection, we introduce the overall procedure of our framework. Algorithm 1 gives the details. The workflow starts by constructing the tag tables for SE and EP respectively (Section 2.2) Then we extract spans whose boundary token pair is labeled with the tag B2E in the SE table as the nodes of our fact graph G. For each span pair, we think they are adjacent in G on the condition that their boundary tokens are strictly aligned in the EP table, as shown in lines 4-10. Based on G, we can leverage the classic graph algorithm such as Bron–Kerbosch (Bron and Kerbosch, 1973) to find all the maximal cliques, where each clique represents one fact expressed in S. Now, the only piece left to is determining the role (subject, object, or predicate) of each node v in the corresponding fact clique c (lines 12-19). Specifically, we enumerate all token pairs (w_i, w_j) in the EP table when w_i is the boundary token of v and w_j is the boundary word of another node in c, and count the outcoming role part of the predicted tags. For example, if the role tag of (w_i, w_j) $\in$ {S2S, S2P, S2O}, then we will increase the counter of subject. The most predicted type $\in$ {subject, predicate, object} is regarded as the role of v in c. Finally, all nodes c are assembled according to their roles to output the de-

	OpenIE4		SAOKE	
	# of sentences	# of facts	# of sentences	# of facts
Train	91,277	180,517	37,544	133,400
Dev	638	2,548	4,693	16,563
Test	634	2,715	4,693	16,407

Table 1: The statistics of OpenIE4 and SAOKE.

	Overlapping	Discontinuous	Nested	Complicated
OpenIE4	18,862	7,428	37,724	44,373
SAOKE	26,878	22,480	12,572	32,096

Table 2: The statistics about complicated facts.

sired fact. If there are multiple nodes with the same role in c, we think the fact contains discontinuous elements and the spans embodied by these nodes will be merged following their original order in S.

3 Experiments

3.1 Datasets

Our experiments are conducted on two benchmarks. (1) OpenIE4 is published in IMoJIE (Kolluru et al., 2020b) and pre-processed by Kolluru et al. (2020a). The training data is automatically labeled by running OpenIE-4, ClausIE, and RnnOIE on the sentences sampled from Wikipedia. While the dev and test sets (CaRB (Bhardwaj et al., 2019)) are manually labeled to ensure its quality. (2) SAOKE is a human-annotated Chinese OpenIE dataset collected from Baidu Baike and released by Sun et al. (2018). This is the largest publicly available human-annotated data set for OpenIE. Compared with OpenIE4, SAOKE avoids the problem of data noise caused by model-derived automatic annotation, so it can evaluate the model performance more accurately. However, because the authors did not give the details of the training/dev/test set partition, this dataset has not been well used. In this work, we re-split SAOKE and reproduce the recent state-of-the-art OpenIE methods to comprehensively evaluate our proposed model. The descriptive statistics of the datasets are reported in Table 1.

Besides, we also count the number of sentences in the data set that contains at least one complicated fact, as shown in Table 2. It can be seen that identifying the complicated facts is very important for OpenIE, because the sentences containing complicated facts account for 48% and 68% in OpenIE4 and SAOKE, respectively.

3.2 Extension of MacroIE

In Section 2, for the sake of brevity, we assume that the facts to be extracted are all binary, that is, they contain a subject, a predict, and an object, and they can be extracted directly from the sentence. However, some special extraction requirements are often encountered in real scenarios. We discuss how to simply modify our MacroIE model to adapt to these scenarios in this section.

N-ary Extraction N-ary fact can be formed as (subject, predicate, object$_1$, $\cdots$, object$_m$). In our MacroIE model, the role of fact element is determined by the edge types in the edge prediction table. So we can easily handle $n-$ ary fact extraction by extending the tag set of edge prediction. For example, if $m = 2$, then the tag set of edge prediction is {B-S2S, B-S2P, B-S2O$_1$, B-S2O$_2$, B-P2S, B-P2P, B-P2O$_1$, B-P2O$_2$, B-O$_1$2S, B-O$_1$2P, B-O$_1$2O$_1$, B-O$_1$2O$_2$, B-O$_2$2S, B-O$_2$2P, B-O$_2$2O$_1$, B-O$_2$2O$_2$, E-S2S, E-S2P, E-S2O$_1$, E-S2O$_2$, E-P2S, E-P2P, E-P2O$_1$, E-P2O$_2$, E-O$_1$2S, E-O$_1$2P, E-O$_1$2O$_1$, E-O$_1$2O$_2$, E-O$_2$2S, E-O$_2$2P, E-O$_2$2O$_1$, E-O$_2$2O$_2$}. The span extraction module and the decoding workflow remain the same. m is set as 3 and 4 in OpenIE4 and SAOKE respectively according to the data set statistics.

Absent Word Prediction. OpenIE4 is required to predict tokens that are not present in the sentence. For example, one of the fact required to be extracted from *US president Donald Trump gave a speech on Wednesday.* in OpenIE4 is (*Donald Trump*, [*is*] *president* [*of*], US). To address this problem, following IGL-OIE (Kolluru et al., 2020a), we select the most frequent such tokens including *is*, *of* and *from*, and insert them to the extracted fact properly. In concrete, we observe that all of these tokens can only be inserted to the boundary positions of fact elements, so we add three special tags to the span extraction table, including *is*-B2E, B2E-*of*, B2E-*from*, where *is*-B2E denotes that *is* should be inserted between the span, and B2E-*of* means we need to add *of* to the end position of corresponding span.

Hidden Predicate Extraction. In SAOKE, some predicates may be expressed implicitly in sentences, such as *Description* and *Location*. For example, the expression of *Paris France)* implies the fact (*France*, *Location*, *Paris*). Fortunately, the number of such hidden predicates is limited in SAOKE, so we can integrate it into the tag set of edge prediction. Taking

Location as an example, we designed eight additional tags to represent it: {B-S2O-Loc, B-S2S-Loc,B-O2S-Loc,B-O2O-Loc, E-S2O-Loc, E-S2S-Loc,E-O2S-Loc, E-O2O-Loc}, in which B-S2O-Loc means the relation between subject and object is *Location*. During decoding, If Loc appears in all edges of a clique, and there is no predicate in the clique, then we take *Location* as the predicate of the fact represented by the clique.

3.3 Baselines

We summarize the OpenIE studies and compare our model against several recent neural systems following previous work. They include labeling (RnnOIE (Stanovsky et al., 2018), SenseOIE (Roy et al., 2019) and IGL-OIE (Kolluru et al., 2020a)), generation (NOIE (Cui et al., 2018) and IMoJIE (Kolluru et al., 2020b)) and span-based (SpanOIE (Zhan and Zhao, 2020)) systems. To make comparison on SAOKE, we re-implement the state-of-the-art models IGL-OIE and IMoJIE based on the BERT-base-Chinese encoder using official implementations. Their hyper-parameters have been carefully tuned on the dev set.

Note that we compare against IGL-OIE rather than the final system OpenIE6 in (Kolluru et al., 2020a). OpenIE6 is an OpenIE system based on IGL-OIE with human-designed soft rules (generated by POS tools) and a coordination analyzer (trained with additional data). In our experiments, all the baseline models and the proposed MacroIE model are trained on the benchmark data, without using additional rules or tools. Therefore, we think it may be unfair to compare with OpenIE6. In addition, OpenIE6 cannot be trained and tested on the Chinese data set SAOKE, because the rules and Coordination Analyzer it uses are Specially designed for English, and hard to extend to other languages.

3.4 Evaluation Metrics

(1) CaRB(1-1) (Bhardwaj et al., 2019) considers the number of common tokens in (gold, predicted) pair for each argument of the fact. Then a one-to-one mapping is created by greedily matching gold with one of the predicted facts on the basis of token-level F1 score. (2) CaRB (Kolluru et al., 2020a) is a variant of CaRB(1-1) that retains CaRB(1-1)'s similarity computation, but uses a one-to-one mapping and a multi-to-one mapping for precision and recall, respectively. (3) Gestalt (Sun et al., 2018) replaces the token-level matching strategy of CaRB(1-1) with gestalt pattern matching. Firstly, it formats a fact into a string by filling the predicate and arguments into the placeholders of one fact. Then the gestalt pattern matching function (Black, 2004) measures the similarity of two fact strings. If the similarity is greater than a threshold (e.g., 0.85), then the two facts are judged as telling the same thing. For more details, please refer to the original paper or our evaluation code in supplementary.

For each system, we report the F1 score by applying the above three scorers to the predicted facts. OpenIE systems typically associate a confidence value with each extracted fact, which can be varied to generate a precision-recall (P-R) curve. We also report the area under the P-R curve (AUC) and the point in the P-R curve corresponding to the optimal F1 (Opt. F1) for all scorers.

3.5 Hyper-parameter settings

We build MacroIE upon the pre-trained weights of BERT-base-cased (Devlin et al., 2019) and BERT-base-Chinese for English and Chinese respectively. The network parameters are optimized by Adam (Kingma and Ba, 2014) with a learning rate of 1e-5. The batch size is fixed to 12. The threshold for converting probability to tag is set as 0.3. All the hyper-parameters are tuned on the dev set. We run our experiments on the NVIDIA Tesla V100 GPU server for at most 20 epochs, and choose the model with the best gestalt F1 score on the dev set to output results on the test set. we report the test score of the run with the median dev score among 5 randomly initialized runs.

3.6 Main results

Table 3 reports the performance comparisons across all metrics on OpenIE4 dataset. Overall, our method, MacroIE outperforms others on all metrics (e.g., it obtains a 3.4% improvement on Gestalt Optimal F1 score over the next best method). Even using the relatively simpler neural architecture, our system is still significantly superior to the state-of-the-art approaches (ImoJIE and IGL-OIE), which are based on iterative message-passing networks. Such performance gains mainly come from: (1) These models are required to consider the prediction order of facts, which places an unnecessary burden on the model. While MacroIE formulates OpenIE as a maximal clique discovery problem, thus removing the confusing supervision caused by the fact order issue in the training process. (2) MacroIE predicts all fact at once, which is immune

System	CaRB(1-1)			CaRB			Gestalt		
	F1	AUC	Opt.F1	F1	AUC	Opt.F1	F1	AUC	Opt.F1
SenseOIE (Roy et al., 2019)	23.9	-	23.9	28.2	-	28.2	-	-	-
SpanOIE (Zhan and Zhao, 2020)	37.9	-	37.9	48.5	-	48.5	-	-	-
RnnOIE (Stanovsky et al., 2018)	39.3	18.3	39.5	49.0	26.1	49.1	7.4	4.9	7.7
NOIE (Cui et al., 2018)	38.3	19.8	38.7	51.1	32.8	51.6	9.0	4.3	9.2
IMoJIE (Kolluru et al., 2020b)	41.2	22.2	41.4	53.3	33.3	53.5	9.6	5.6	9.4
IGL-OIE (Kolluru et al., 2020a)	41.0	22.9	41.1	52.2	33.7	52.4	10.1	5.4	9.7
MacroIE (ours)	**43.5**	**25.0**	**43.8**	**54.8**	**36.3**	**55.1**	**12.9**	**6.0**	**13.1**

Table 3: Main results on OpenIE4. The improvement over baselines is significant (p-value < 0.05).

System	CaRB(1-1)			CaRB			Gestalt		
	F1	AUC	Opt.F1	F1	AUC	Opt.F1	F1	AUC	Opt.F1
IMoJIE (Kolluru et al., 2020b)	36.6	22.6	37.0	38.7	25.4	39.5	36.4	22.5	37.3
IGL-OIE (Kolluru et al., 2020a)	37.6	22.8	38.4	39.3	25.5	40.6	37.1	23.6	38.4
MacroIE (ours)	**41.2**	**24.5**	**41.5**	**42.7**	**27.8**	**43.7**	**42.8**	**27.2**	**43.7**

Table 4: Main results on SAOKE. The improvement over baselines is significant (p-value < 0.05).

from cascading error in the autoregressive decoding process of ImoJIE and IGL-OIE.

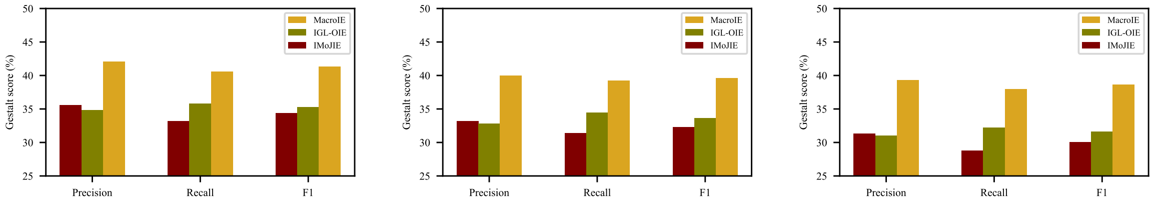
An abnormal phenomenon is that there is a huge difference between the scores in CaRB and Gestalt. CaRB focuses on the word-level coincidence rate between the predicted and the gold facts, while Gestalt is more rigorous, requiring not only the common words but also the correct word order. We find that the automatically-derived training data of OpenIE4 is very noisy, containing a large number of error fact tuples. Moreover, the annotation criteria of the training set and test set are different, which may significantly affect the performance evaluation. We suggest that OpenIE4 is not unbiased enough to be used as a benchmark, so we further conduct experiments on SAOKE. The results are listed in Table 4. On the manually labeled Chinese dataset which eliminates the impact of data bias, the scores of CaRB and Gestalt tend to be consistent. Our neural model again achieves the best performance among all models in terms of all metrics, and the performance gap even widens compared with Table 3. It outperforms IGL-OIE, by 5.7 F1 pts, 3.6 pts of AUC, and 4.7 pts of optimal F1 in Gestalt, demonstrating the applicability of our proposed MacroIE in different languages.

3.7 Analysis

In this subsection, we try to answer some potential questions that others may ask for a deep understanding of our method.

Analysis on Extracting Complicated Facts. As claimed in the introduction, an ideal OpenIE system should be capable of identifying facts having overlapping, discontinuous or nested structures. To evaluate this ability, we construct a subset of test data where only sentences with at least one complicated fact are included. The datasets' distribution about different kinds of complicated facts is detailed in Appendix C. From Figure 6, we find that MacroIE achieves excellent performance in all three patterns, indicating that our model is more suitable for complicated scenarios than the baselines. We attribute the performance improvement to two design choices: (1) the table filling scheme is effective in recognizing discontinuous and nested fact elements; (2) the structure of our fact graph and its accompanied decoding algorithm naturally address the challenge of extracting overlapping facts.

Analysis on Extracting Multiple Facts. We compare the ability of models in extracting multiple facts in a sentence. We divide the sentences in test sets into 5 sub-classes by fact count m. Each class contains sentences where $m \leq 3$, $4 \leq m \leq 6$, $7 \leq m \leq 9$, $10 \leq m \leq 12$ or $m > 12$. The results are shown in Figure 7. In general, MacroIE achieves the best results in all sub-classes. All the baselines present an obvious decreasing trend with the increasing of fact numbers in the sentence, while MacroIE shows stable performance. The greatest improvement of the F1 score comes from the most difficult sub-class, e.g., MacroIE outperforms IGL-OIE by 12.1% for more than 12 facts in a sentence. These observations confirm a core flaw



(a) Performance on Overlapping Facts (b) Performance on Discontinuous Facts (c) Performance on Nested Facts

Figure 6: Gestalt scores on (a) extracting overlapping facts, (b) identifying discontinuous facts, and (c) detecting nested facts. All the analyses are conducted on the test set of SAOKE.

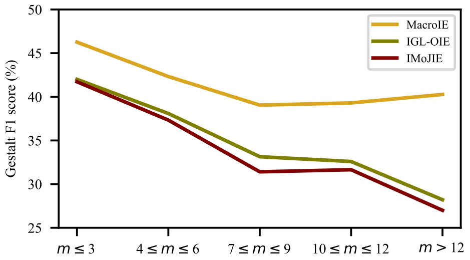


Figure 7: Gestalt F1 score of conducting extraction in sentences that contain different numbers of facts (m).

in the autoregressive decoding process of baseline models: a wrong prediction can mislead all the following prediction steps. Such accumulated error decreases the performance especially in predicting long sequences, i.e., multi-fact prediction. On the contrary, MacroIE is non-autoregressive. Its extraction of different facts is independent of each other, so the error propagation between facts is avoided.

Analysis on Cascading Error. An interesting design of our model is decoupling the dependency between span extraction and edge prediction via the ingenious table filling scheme, thus immune from the cascading error. To explore the effectiveness of this design, we implement a two-stage pipeline version of our model. In which we enumerate all possible span pairs output from the span extraction step and classify the relations between each of these pairs. The experimental results show that this change hurts the results as the Gestalt F1 score on SAOKE drops from 42.8% to 41.3%.

Analysis on Computational Cost. IGL-OIE is the current fastest OpenIE model (Kolluru et al., 2020a). While IGL-OIE runs sequence tagging M times, M is a pre-defined maximum number of facts, our model tags the same n-word sentence for $2n$ times to construct the fact graph. This means our model is more time-consuming ($O(M)$ vs. $O(n)$). Actually, IGL-OIE and MacroIE both use BERT as the basic encoder, which is the most time-consuming part and takes up the most of model parameters, so the time cost of table filling is not significant. For instance, The test-time speed of IGL-OIE is 1.8x faster than MacroIE when running on the SAOKE test set in the same GPU server. Considering the distinct performance advantage of MacroIE presented in previous subsections, we think its computational cost is acceptable.

Discussion. There is a commonly accepted conclusion in previous works (Kolluru et al., 2020a,b): *capturing dependency among facts is crucial for OpenIE models*. It seems that we break this conclusion by presenting a non-autoregressive OpenIE system MacroIE. MacroIE ensures the one-to-one correspondence between the maximal clique and the fact by discovering the maximal clique on the constructed fact graph, so the extraction of different facts is independent of each other. However, in the construction phase of the fact graph, including span extraction and edge prediction, MacroIE may implicitly use the dependency information between facts. Therefore, we want to attribute our performance improvement to MacroIE can capture the dependencies that do exist in a non-autoregressive manner, as opposed to autoregressive manners used in prior works. It will be an interesting research direction to explore whether there is a causal interdependence between facts and how to make more elegant and intuitive use of such correlation.

4 Related Work

Open information extraction has attracted much attention from researchers during the past decade (Niklaus et al., 2018). Banko et al. (2007) are the first to introduce the Open Information Extraction (OpenIE) paradigm, and propose TextRunner, the first highly scalable model for the task. In the following, various OpenIE systems applying costly hand-crafted rules or self-supervised learning paradigm based on linguistic patterns such as part-of-speech tags and syntactic features have been proposed over the years (Wu and Weld, 2010; Fader et al., 2011; Schmitz et al., 2012; Akbik and Löser, 2012; Mesquita et al., 2013; Del Corro and Gemulla, 2013; Yahya et al., 2014; Angeli et al., 2015; Falke et al., 2016; White et al., 2016). They strongly rely on external NLP tools. Thus, their performance depends on the quality of the features obtained from these NLP tools. However, these features are not always accurate for various domains and contexts (Bekoulis et al., 2018).

Recently, OpenIE has achieved great advances with the help of supervised neural networks to bypass the handcrafted patterns and alleviate error propagation. There are two main paradigms in the relevant research. The first one, called tagging-based model (Stanovsky et al., 2018; Roy et al., 2019; Jiang et al., 2019), labels each word in the sentence as either subject, predicate, object, or None for extraction. To identify complicated facts containing overlapping, discontinuous, or nested elements, the recent tagging-based model (Kolluru et al., 2020a) generates a list of tag sequences for one sentence where each sequence corresponding to one extracted fact. The tag sequences are labeled one by one iteratively, e.g., the predicted labels of one tag sequence are passed to the next iteration to fill up another sequence to avoid redundant extraction. Generating-based methods belong to another major paradigm. These methods cast OpenIE as a sequence-to-sequence generation problem, where the input sequence is the sentence and the output sequence is the desired facts (Cui et al., 2018; Sun et al., 2018; Liu et al., 2020; Kolluru et al., 2020b). In principle, generation is powerful because it is able to assign one word to multiple facts and change word order, thus the complicated fact extraction problem can be solved in nature.

Generally speaking, the best-performing OpenIE models at present, whether based on tagging or generation, all pre-define a sequential order for the target facts and then make prediction according to the order autoregressively, which means the current fact prediction relies on the previous output. As discussed in the introduction, this design inevitably has to sort the target facts in a certain order during the training phase, while the facts contained in a sentence have no intrinsic order in essence. What more serious is that a mispredicted fact will directly affect the extraction of all the following facts, resulting in cascading error. In this paper, for the first time, we break the sequential extraction process and propose a one-stage OpenIE model, which is able to extract all kinds of facts without relying on the dependency among facts, realizing non-autoregressive open information extraction.

Maximal clique discovery is to find the clique with most nodes in a given graph (Lu et al., 2017). This problem has been extensively studied in graph theory and directly applied in various fields, such as community search in social networks (Papadopoulos et al., 2012), team formation in expert networks (Lappas et al., 2009), anomaly detection in complex networks (Leung and Leckie, 2005), and discontinuous named entity recognition (Wang et al., 2021). Motivated by the finding that all the elements in a fact of OpenIE have pairwise strong connections, which is similar to the property of maximal clique in the graph, we extend the concept of maximal clique discovery to OpenIE, and successfully implement task transformation. Our results show that OpenIE can be cleverly cast as a maximal clique discovery problem on a fact graph.

5 Conclusion

In this paper, we present a non-autoregressive OpenIE system MacroIE. It predicts the fact set at once based on a novel view of OpenIE as a maximal clique discovery problem, thus be relieved of predicting the extraction order of multiple facts in previous autoregressive OpenIE models. Experimental results show that our proposed networks outperform state-of-the-art baselines in all of the metrics on two public datasets. Further analysis demonstrates the ability of our model in handling complicated and multiple fact extractions. OpenIE is one of the most complex tasks in information extraction (IE). In the future, we would like to explore similar maximal clique based task formulation in other IE tasks, such as event and aspect extraction.

6 Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This research is supported by the National Key Research and Development Program of China (No.2016YFB0801003), the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02040400), and the National Natural Science Foundation of China (No.61902394).

References

Alan Akbik and Alexander Löser. 2012. Kraken: N-ary facts in open information extraction. In *Proceedings of AKBC-WEKEX*.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of ACL*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. Carb: A crowdsourced benchmark for open ie. In *Proceedings of EMNLP*.

Paul E Black. 2004. Ratcliff/obershelp pattern recognition. *Dictionary of algorithms and data structures*, 17.

Coen Bron and Joep Kerbosch. 1973. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of ACL*.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of WebConf*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

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

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of KDD*.

Tobias Falke, Gabriel Stanovsky, Iryna Gurevych, and Ido Dagan. 2016. Porting an open information extraction system from english to german. In *Proceedings of EMNLP*.

Zhengbao Jiang, Pengcheng Yin, and Graham Neubig. 2019. Improving open information extraction via iterative rank-aware learning. In *Proceedings of ACL*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020a. Openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of EMNLP*.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. 2020b. Imojie: Iterative memory-based joint open information extraction. In *Proceedings of ACL*.

Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of KDD*.

Kingsly Leung and Christopher Leckie. 2005. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of ACCS*.

Guiliang Liu, Xu Li, Jiakang Wang, Mingming Sun, and Ping Li. 2020. Extracting knowledge from web text with monte carlo tree search. In *Proceedings of Webconf*.

Can Lu, Jeffrey Xu Yu, Hao Wei, and Yikai Zhang. 2017. Finding the maximum clique in massive graphs. *Proceedings of VLDB*.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of EMNLP*.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of COLING*.

Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. 2012. Community detection in social media. *Data Mining and Knowledge Discovery*.

Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. Supervising unsupervised open information extraction models. In *Proceedings of EMNLP*.

Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of EMNLP*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of NAACL*.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*.

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 WSDM*.

Jialong Tang, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, Xinyan Xiao, and Hua Wu. 2020. Syntactic and semantic-driven learning for open information extraction. In *Findings of EMNLP*.

Yucheng Wang, Bowen Yu, Hongsong Zhu, Tingwen Liu, Nan Yu, and Limin Sun. 2021. Discontinuous named entity recognition as maximal clique discovery. In *Proceedings of ACL*.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings EMNLP*.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *Proceedings of ACL*.

Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of EMNLP*.

Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *Proceedings of AAAI*.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of seq2seq models in joint entity and relation extraction. In *Findings of EMNLP*.