

Syntactic Multi-view Learning for Open Information Extraction

Kuicai Dong^{1,2}, Aixin Sun¹, Jung-Jae Kim², Xiaoli Li^{1,2,3}

¹ School of Computer Science and Engineering, Nanyang Technological University, Singapore

kuicai001@e.ntu.edu.sg, axsun@ntu.edu.sg

² Institute for Infocomm Research, A*STAR, Singapore

³ A*STAR Centre for Frontier AI Research, Singapore

{jjkim, xlli}@i2r.a-star.edu.sg

Abstract

Open Information Extraction (OpenIE) aims to extract relational tuples from open-domain sentences. Traditional rule-based or statistical models have been developed based on syntactic structures of sentences, identified by syntactic parsers. However, previous neural OpenIE models under-explore the useful syntactic information. In this paper, we model both *constituency* and *dependency* trees into word-level graphs, and enable neural OpenIE to learn from the syntactic structures. To better fuse heterogeneous information from both graphs, we adopt multi-view learning to capture multiple relationships from them. Finally, the finetuned constituency and dependency representations are aggregated with sentential semantic representations for tuple generation. Experiments show that both constituency and dependency information, and the multi-view learning are effective. Our model is publicly available.¹

1 Introduction

Open Information Extraction (OpenIE) aims to generate structured tuples from unstructured open-domain text (Yates et al., 2007). The extracted tuples are in the form of $\langle Subject, Relation, Object \rangle$ for binary relation, and $\langle ARG_0, Relation, ARG_1, \dots, ARG_n \rangle$ for n -ary relation. It has been a critical NLP task as it is domain-independent and does not rely on pre-defined ontology schema. The structured relational tuples are beneficial to many downstream tasks such as question answering (Khot et al., 2017), knowledge base population (Martínez-Rodríguez et al., 2018; Gashteovski et al., 2020) and word embedding generation (Stanovsky et al., 2015).

In general, traditional OpenIE systems are either statistical or rule-based. They extract relational tuples mainly based on certain sentence patterns heuristically defined on syntactic structures. The

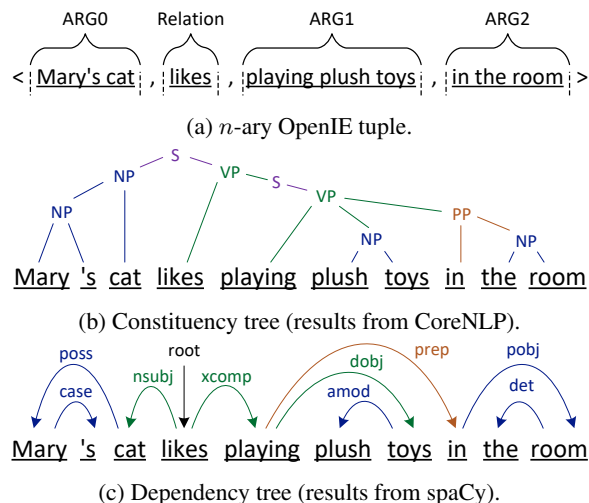


Figure 1: An example of constituency tree, dependency tree, and n -ary OpenIE tuple to be extracted for sentence “Mary’s cat likes playing plush toys in the room.”

limitation of patterns causes traditional OpenIE systems to be ineffective in handling complex sentences. Recently, neural OpenIE systems have been developed and showed promising results. Neural OpenIE systems no longer depend on pre-defined patterns. Instead, they learn to extract relational tuples directly from unstructured text in an end-to-end manner. However, utilizing syntactic information is under-explored among the neural OpenIE systems, although syntactic information is widely explored in other Information Extraction tasks such as Semantic Role Labeling (SRL) (Fei et al., 2021) and Relation Extraction (RE) (Zhang et al., 2018).

As syntactic information was proved to be useful for the traditional OpenIE systems, we argue that it is important for neural OpenIE systems as well. Figure 1a shows an n -ary tuple expected from an example sentence. Figure 1b displays the sentence’s constituency tree, where the phrases are labeled with constituency tags. The dependency tree (shown in Figure 1c) represents syntax through directed and typed edges between words, instead

¹https://github.com/daviddongkc/smile_oie

of between phrases. We observe that the boundary of the OpenIE tuple (shown in Figure 1a) highly coincides with edges from constituency and dependency trees. As such, we believe both constituency and dependency trees provide useful and complementary syntactic information for OpenIE systems. The next question is: how to effectively incorporate syntactic information to neural OpenIE models?

To fully explore the syntactic information from both parse trees, we convert them into ‘node-sharing’ graphs, *i.e.*, *constituency graph* (denoted as const-graph) and *dependency graph* (denoted as dep-graph), respectively. A dependency relation specifies a relationship between two words, and we can represent words as nodes and the relationship as corresponding edge. The key challenge is to map constituency tree to a graph in which all of its nodes are words, not phrases. Meanwhile, the graph needs to largely capture the constituency syntactic information. In this work, we present a novel method for the conversion of constituency trees. With the converted const-graph modelled at the word level, we can now easily integrate it with dep-graph. Moreover, both graphs can be directly integrated with Pre-trained Language Model (PLM) which provides word-level representation.

In order to leverage heterogeneous syntactic information from both const-graph and dep-graph, we propose a novel neural OpenIE model: SMiLe-OIE (Syntactic Multi-view Learning for Open Information Extraction). It first encodes a sentence using BERT (Devlin et al., 2019), and subsequently uses two syntactic encoders, namely Const-encoder and Dep-encoder. The model represents constituency and dependency relations of the sentence with the corresponding BERT representations and applies two Graph Convolutional Networks (GCN) (denoted as Const-GCN and Dep-GCN) to learn graph representations for const-graph and dep-graph separately. The representations from BERT, Const-GCN, and Dep-GCN are aggregated and finally used for tuple generation. To better fuse the heterogeneous syntactic graph representations, SMiLe-OIE introduces a subtask, multi-view learning, to learn multiple types of relationships among const-graph and dep-graph. The multi-view learning loss is used to finetune the graph representations along with OpenIE loss. In summary, our contributions are threefold:

- We propose a novel strategy to map phrase-level relations in constituency tree into word-

level relations, and to enhance each word’s representation with constituency path information.

- We propose **SMiLe-OIE**, the first neural OpenIE system that incorporates heterogeneous syntactic information through GCN encoders and multi-view learning.
- Our experimental results show that the proposed neural OpenIE model achieves better performance than state-of-the-art methods.

2 Related Work

Syntax Usage in OpenIE. Open Information Extraction (OpenIE) was first proposed by Yates et al. (2007), and TextRunner is the first OpenIE system that generates relational tuples in open domain. Before deep learning era, many statistical and rule-based systems have been proposed, including Reverb (Fader et al., 2011), R2A2 (Fader et al., 2011), OLLIE (Mausam et al., 2012), Clausie (Corro and Gemulla, 2013), Stanford OpenIE (Angeli et al., 2015), OpenIE4 (Mausam, 2016), NESTIE (Bhutani et al., 2016), and MINIE (Gashteovski et al., 2017). Most of these models extract relational tuples based on syntactic structures such as part-of-speech (POS) tags and dependency trees. In this sense, syntactic information has been essential to OpenIE.

Recently, neural OpenIE systems (Cui et al., 2018; Stanovsky et al., 2018; Roy et al., 2019; Kolluru et al., 2020a; Dong et al., 2021; Vasilkovsky et al., 2022; Kotnis et al., 2022) have been developed and showed promising results. Neural OpenIE systems are able to extract relational tuples end-to-end based on the semantic encoding of input sentence. The analysis of syntactic structure of sentence, which was required by traditional models, seems no longer necessary. As a result, the usage of syntactic information is under-explored in neural OpenIE models. Nevertheless, there exist some neural OpenIE systems that utilize some forms of syntactic information. For example, RnOIE (Stanovsky et al., 2018) projects POS tag of each word into POS embedding and concatenates it with word embedding as input to sentence encoder. SenseOIE (Roy et al., 2019) further concatenates word embedding with dependency embedding. CIGL-OIE (Kolluru et al., 2020b) finds all head verbs in the sentence and pre-defines a few POS patterns to explicitly constrain the model training. MGD-GNN (Lyu et al., 2021) connects words, if they are in dependency relations, in an

undirected graph and applies graph attention network (GAT) to the graph (Veličković et al., 2018) as its graph encoder. Although MGD-GNN uses some graphic information of dependency, it loses other information like the directness and types of dependency relations.

In short, we observe that existing neural OpenIE systems fail to explore some syntactic features and that the integration of syntax is in a shallow manner. Compared to the them, our SMiLe-OIE is able to leverage full features of heterogeneous syntactic information from both constituency and dependency trees.

Integration of Constituency and Dependency Syntax. Although constituency and dependency trees possess common sentential syntactic information, they capture syntactic information from different perspectives. Recent NLP tasks have benefited from integrating these two syntactic representations. Zhou and Zhao (2019) and Strzyz et al. (2019) integrate dependency and constituency syntactic information as a representation of parse tree or sequence, but not of a graph. To the best of our knowledge, HeSyFu (Fei et al., 2021) is the only work that converts dependency and constituency trees into graphs and performs graph learning strategy on both. In this sense, although HeSyFu is designed for SRL task, it is the most relevant model to ours.

SMiLe-OIE differs from HeSyFu mainly in three perspectives: (1) HeSyFu models constituency tree at phrase level, which is inconsistent with word-level representations from BERT. Meanwhile, HeSyFu models dependency tree at word level, so the constituency representations cannot be directly fused with the dependency representations. (2) As a result, HeSyFu integrates the BERT representations and the two parse trees’ representations with complicated bridging processes, which may hinder synergistic integration of the heterogeneous representations. (3) To better fuse heterogeneous syntactic information, SMiLe-OIE adopts multi-view learning to capture multiple relationships between const-graph and dep-graph representations.

Multi-view Learning Multi-view learning aims to learn representations or features from the multi-view data. Generally, data from different views usually contain complementary information. Therefore, multi-view learning is able to exploit such complementary information to learn more com-

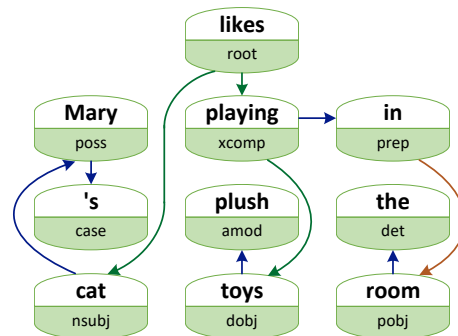


Figure 2: Dependency graph constructed from word-level dependency relations in Figure 1.

prehensive representations than those of single-view learning methods (Li et al., 2019). In the modern era, multi-view data have increased voluminously, leading to more attention of multi-view learning mechanism. The studies of multi-view learning (Yan et al., 2021) mainly fall into: multi-view fusion (Zhao et al., 2017; Sun, 2013), multi-modal learning (Ramachandram and Taylor, 2017; Baltrušaitis et al., 2019), multi-view clustering (Chao et al., 2021), and multi-view representation learning (Li et al., 2019; Guo et al., 2019; Ata et al., 2021). In our work, we perform multi-view learning and fusion on two views of syntactic graphs (i.e., const-graph and dep-graph). To the best of our knowledge, we are the first to use multi-view mechanism to exploit complementary syntactic information in NLP applications.

3 Graph Modelling

In this section, we elaborate on our graph modelling strategy to convert constituency and dependency trees into graphs $G = (U, E)$, where U indicates the set of nodes and E the set of edges. They are $G^{con} = (U^{con}, E^{con})$ for const-graph, and $G^{dep} = (U^{dep}, E^{dep})$ for dep-graph. The two graphs’ nodes correspond to the same set of input sentence’s words. However, the labels of the nodes in the two graphs are different (constituency path for const-graph, and dependency relation type for dep-graph), preserving the syntactic information of constituency and dependency trees. Also, their edge sets are different, where the edges represent word-to-word syntactic relations.

3.1 Dependency Graph Modelling

Dependency tree provides syntactic dependency at word level. Thus, the dep-graph of a sentence is identical to the dependency tree of the sentence, except node labels. For each word, as shown in Figure 1c, there is an inbound relation from its

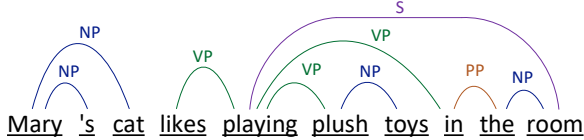


Figure 3: Word-level constituency relations converted from phrase-level relations of constituency tree.

Word	Const. Path	Word	Const. Path
Mary	S-NP-NP	plush	S-VP-S-VP-NP
's	S-NP-NP	toys	S-VP-S-VP-NP
cat	S-NP	in	S-VP-S-VP-PP
likes	S-VP	the	S-VP-S-VP-PP-NP
playing	S-VP-S-VP	room	S-VP-S-VP-PP-NP

Table 1: Example constituency paths.

modifying head word. We follow Fei et al. (2021) to label each word node with its inbound dependency relation type, as exemplified in Figure 2.

3.2 Constituency Graph Modelling

We flatten the phrase-level relations of a constituent structure into a const-graph of word-level relations, which can be directly integrated with word-level granularity from Pre-trained Language Model (PLM) such as BERT. The flattening process is designed to preserve both the phrasal boundary information and the constituency relations which are required for OpenIE task. But note that this flattening process can be used for other related modelling tasks (e.g., SRL, NER, RE).

Word Node Labelling with Constituency Path.

In const-graph, each word is a node, and we label each word node with the path from the root to the word in the constituency structure of the input sentence. Table 1 lists the constituency paths of words in the example sentence in Figure 1. This labelling of words with constituency paths preserves the rich phrasal information of constituency tree.

Word-level Constituency Relations. In const-graph, edges are constituency relations that connect word nodes. We perform relation flattening of the constituency tree in Figure 1b in the following steps: (1) We add an edge between the first and last word in each noun phrase (NP) (e.g., ‘Mary’-‘s’, ‘Mary’-‘cat’, ‘plush’-‘toys’, ‘the’-‘room’). The edge is labelled as ‘NP’. This edge identifies the boundary of NP. (2) If a word and a phrase are siblings (belonging to the same parent node in constituency tree), we connect the word (e.g., Verb in VP, Preposition in PP) to the first word of its

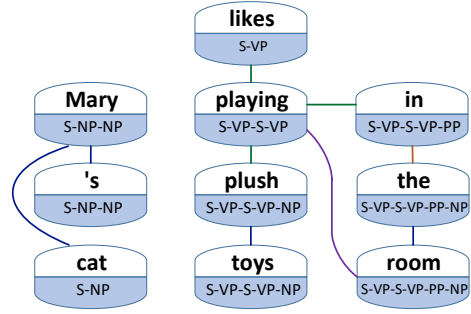


Figure 4: Constituency graph constructed from word-level constituency relations in Figure 3 and hierarchical constituency paths in Table 1.

sibling phrase (e.g., ‘likes’-‘playing’, ‘playing’-‘plush’, ‘playing’-‘in’, ‘in’-‘the’). This edge’s type is marked according to the constituency type of the parent node such as VP and PP. (3) To mark the boundary of intra-sentential clause, we connect the first and last word of each clause (S) and label the edge as ‘S’ (e.g., ‘playing’-‘room’). (4) We remove an edge when the distance between two words is longer than 8 in the input sentence, since the elements in OpenIE tuple are usually short spans.

Figure 3 depicts the word-level relations flattened from the constituency tree in Figure 1b, and Figure 4 depicts the final const-graph.

4 SMiLe-OIE Model

The overall architecture of SMiLe-OIE is illustrated in Figure 5. SMiLe-OIE is based on BERT encoder to get contextualized representations of an input sentence. The BERT representations are then integrated with constituency and dependency information by Const-GCN and Dep-GCN, respectively. Finally, SMiLe-OIE aggregates the BERT, const-graph, and dep-graph representations in order to predict output tuples. Beyond the OpenIE tagging loss, SMiLe-OIE further performs multi-view learning on the const-graph and dep-graph representations, and generates additional multi-view losses to enhance OpenIE tagging accuracy.

4.1 Task Formulation

We formulate OpenIE as a sequence tagging task, using BIO (Beginning, Inside, Outside) tagging scheme like recent neural OpenIE models (Stanovsky et al., 2018). Given an input sentence $s = [t_0, \dots, t_n]$, a variable number of tuples will be extracted. Each tuple can be represented as $[x_0, \dots, x_m]$, where x_j is a contiguous subspan of

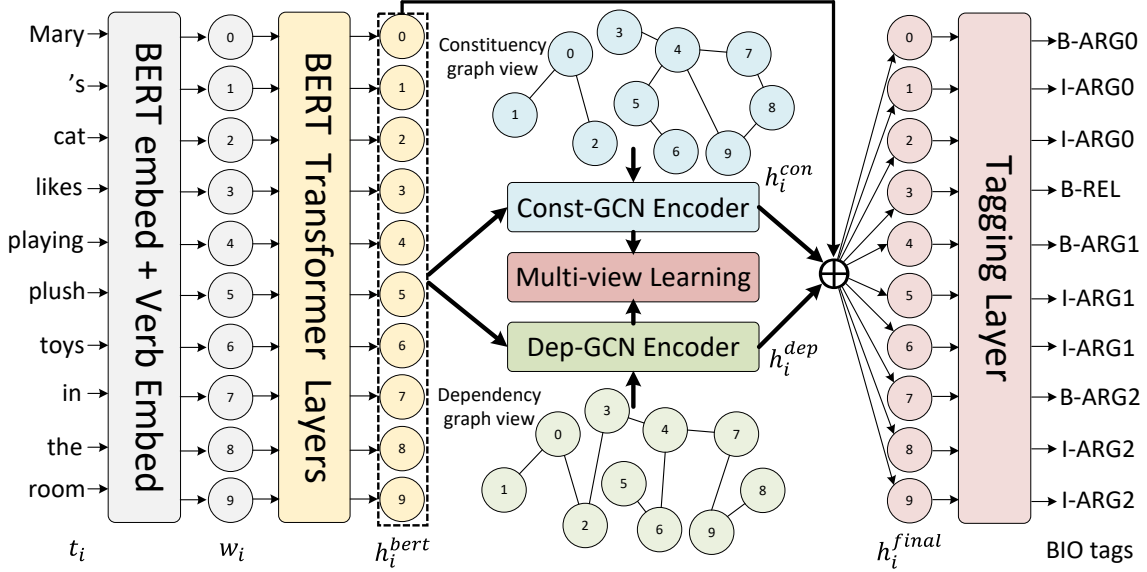


Figure 5: Overview of SMiLe-OIE model architecture.

s . One of x_j is tuple relation (B/I-REL) and the others are tuple arguments (B/I-ARG $_l$).

We assume that each tuple has a verb as relation, since OpenIE relation is typically associated with a verb². Meanwhile, since some verbs in a sentence do not lead to any relational tuple, we assume one verb can be a relation for at most one tuple.

4.2 BERT Encoder with Relation Indicator

We employ BERT (Devlin et al., 2019) as our encoder to analyze semantic interactions among words. We first project all words $[t_0, \dots, t_n]$ into embedding space by summing their word embedding³ and verb embedding, i.e., $w_i = W_{word}(t_i) + W_{verb}(t_i)$. Here, W_{word} is trainable and initialized by BERT word embedding.

W_{verb} is a trainable verb embedding matrix. Verb embedding is to distinguish whether an input word is a relation indicator or not. Given an input sentence, we extract all verbs from the sentence using an off-the-shelf POS tagger. We consider each verb in a sentence to be a potential relation indicator and use the verb embedding to highlight this relation indicator. Specifically, W_{verb} initializes each verb to 1 at a time, and all the other words in the sentence to 0. If a verb in the sentence does not lead to a tuple, we set all of the sequence output tags to be ‘‘O’’. Consequently, the model is able to

²Relation can be referred as predicate, and verb can be referred as predicate head word in other OpenIE works.

³If the word contains multiple sub-words after BERT tokenization, we use the representation of its first sub-word.

learn which verbs lead to relation.⁴

Then, we use $w_s = [w_0, \dots, w_n]$ as the input to the BERT encoder and utilize BERT’s last hidden states as contextualized representations:

$$h_i^{bert} = \text{BERT}(w_i) \in \mathbb{R}^{d_h} \quad (1)$$

4.3 Syntactic GCN Encoders

In this section, we present *syntactic encoders*, which represent the elements of the two graphs with the BERT representations and encode them using GCNs. Recall that the dep-graph and const-graph are represented as $G^z = (U^z, E^z)$, where $z \in \{dep, con\}$. e_{ij}^z in E^z equals to 1 if there is an edge between node n_i^z and node n_j^z ; Otherwise, 0. Each node $n_i^z \in U^z$ has a label (or type), designated as $type\langle n_i^z \rangle$.

The node types of U^{dep} are dependency relations. The syntactic encoder of G^{dep} (called Dep-GCN) takes node embedding as follows:

$$l_i^{dep} = W_{dep}^1(type\langle n_i^{dep} \rangle) \quad (2)$$

where $W_{dep}^1 \in \mathbb{R}^{d_l \times N_{dep}}$ is a trainable matrix, d_l is the size of node embeddings, and N_{dep} is the total number of unique dependency relations.

A node $n_i^{con} \in U^{con}$ has a label of constituency path $type\langle n_i^{con} \rangle = [n_i^{con_0}, \dots, n_i^{con_m}]$, which contains a list of constituent tags. Given a constituency path, we first project all its constituent tags to respective constituent tag vectors, and then average all the constituent tag vectors in this constituency

⁴Implementation details of verb-tuple alignment are described in Appendix A.1.

path as inputs to the syntactic encoder of G^{con} (called Const-GCN) as follows:

$$l_i^{con} = \text{avg}(W_{con}^1(\text{type}(n_i^{con}))) \quad (3)$$

$$= \text{avg}(W_{con}^1(n_i^{con_0}), \dots, W_{con}^1(n_i^{con_m}))$$

where $W_{con}^1 \in \mathbb{R}^{d_i \times N_{con}}$ is a trainable matrix, and N_{con} is the total number of unique constituency tags. $\text{avg}()$ indicates the averaging operation on a sequence of vectors.

Each syntactic encoder (Dep-GCN, Const-GCN) employs a separate GCN to encode the corresponding graph (G^{dep} , G^{con}). The computation of the GCN representation is formulated as:

$$h_i^z = \text{ReLU}\left(\sum_{j=1}^n \alpha_{ij}^z (h_j^{bert} + W_z^2 \cdot l_j^z + b_z)\right) \quad (4)$$

where n refers to the total number of word nodes in the graph, $W_z^2 \in \mathbb{R}^{d_h \times d_i}$ is a trainable weight matrix for syntactic type embeddings, and $b_z \in \mathbb{R}^{d_h}$ is the bias vector. The neighbour connecting strength distribution α_{ij}^z is calculated as below:

$$\alpha_{ij}^z = \frac{e_{ij}^z \cdot \exp((m_i^z)^T \cdot m_j^z)}{\sum_{k=1}^n e_{ik}^z \cdot \exp((m_i^z)^T \cdot m_k^z)} \quad (5)$$

where $m_i^z = h_i^{bert} \oplus l_i^z$, and \oplus is concatenation operator. In this way, node type and edge information are modelled in a unified way.

Finally, we aggregate the sequence representations from BERT encoder in Eq.(1) and the graph representations from Const-Encoder and Dep-Encoder in Eq.(4) as follows:

$$h_i^{final} = h_i^{bert} \oplus h_i^{con} \oplus h_i^{dep} \quad (6)$$

where h_i^{final} is used by the tagging layer for tuple prediction.

4.4 Multi-view Learning

Recall that the const-graph and the dep-graph share the same set of nodes U and have two different sets of node representations h^{con} and h^{dep} , and two different edge sets E^{con} and E^{dep} , respectively. We treat const-graph and dep-graph as two syntactic views $z \in \{dep, con\}$ of the input sentence. We adopt multi-view learning (Ata et al., 2021) in order to explore three types of relationships among the representations of const-graph and dep-graph views. The multi-view learning loss is used to fine-tune these representations, which can provide rich syntactic information for tuple generation.

We consider three categories of relationships between these two views. In the first category, the multi-view learning captures the inter-node and

intra-view relationship in each view. In the second category, it aligns instances of the same node across various views, *i.e.*, intra-node and inter-view relationship. In the third category, it ensures the nodes that are connected in one view should be similar with each other in another view, *i.e.*, inter-node inter-view relationship.

Inter-node Intra-view Relationship. We design a loss to ensure the representations of connected nodes i and j in the same view z , *i.e.*, h_i^z and h_j^z , to be similar.⁵ This is to ensure coherence within the same view.

$$L_{R_1} = - \sum_{z \in \{dep, con\}} \sum_{i \in U} \sum_{j \in U} e_{ij}^z \cdot \log P(h_j^z, h_i^z) \quad (7)$$

$$P(h_j^z, h_i^z) = \frac{\exp(h_j^z \cdot h_i^z)}{\sum_{k \in U} \exp(h_k^z \cdot h_i^z)} \quad (8)$$

Intra-node Inter-view Relationship. While const-graph and dep-graph exhibit diversity, they ultimately converge on a common set of words. The same word, although bearing different syntactic functions, well connects the two views. Therefore, we design a loss for the intra-node and inter-view relations. Specifically, we make sure a node i 's const-graph representation h_i^{con} to be similar to its dep-graph representation h_i^{dep} by minimizing the following loss:

$$L_{R_2} = - \sum_{z \in \{dep, con\}} \sum_{i \in U} \sum_{z' \neq z} \log P(h_i^{z'}, h_i^z) \quad (9)$$

where z' indicates the other view than z , and $P(h_i^{z'}, h_i^z)$ is computed in a similar way as in Eq. 8.

Inter-node Inter-view Relationship. We observe that const-graph and dep-graph share many common edges. In another word, two nodes linked in const-graph are often linked with each other in dep-graph as well. Consequently, we explore inter-view and inter-node relations in order to leverage the frequent edge sharing between the two graphs. Specifically, if node i and node j are connected in const-graph, we design a loss to move node i 's const-graph representation h_i^{con} towards node j 's dep-graph representation h_j^{dep} , as follows:

$$L_{R_3} = - \sum_{z \in \{dep, con\}} \sum_{i \in U} \sum_{z' \neq z} \sum_{j \in U} e_{ij}^{z'} \cdot \log P(h_i^{z'}, h_j^z) \quad (10)$$

⁵In the original paper (Ata et al., 2021), the corresponding objective of their multi-view learning is to make node representations within each view close to each other. In our problem setting, we redefine this objective on 'connected nodes' only.

Dataset	Source	#Sent	#Tuple
LSOIE-wiki-train	QA-SRL 2.0	19,591	45,890
LSOIE-wiki-test	QA-SRL 2.0	4,660	10,604
LSOIE-sci-train	QA-SRL 2.0	38,826	80,271
LSOIE-sci-test	QA-SRL 2.0	9,093	17,031
CaRB-train	OpenIE 4	92,774	190,661
CaRB-test	Crowdsourcing	1,282	5,263

Table 2: Statistics of OpenIE datasets used in training and evaluating SMiLe-OIE.

Loss Function. We combine the losses of the three categories of relations in Equations (7), (9), and (10) with the OpenIE sequence tagging loss L_{CE} . L_{CE} is the cross-entropy loss between the gold and the predicted word labels in sequence tagging (*i.e.*, the BIO tags shown in Figure 5). The overall loss for our multi-view learning OpenIE is:

$$L = L_{CE} + \alpha \cdot L_{R_1} + \beta \cdot L_{R_2} + \gamma \cdot L_{R_3} \quad (11)$$

where α , β , and γ are hyper-parameters, indicating the importance of each individual loss.

5 Experiments

We mainly conduct our experiments on LSOIE (Solawetz and Larson, 2021), a large-scale OpenIE data converted from QA-SRL 2.0 in two domains, *i.e.*, Wikipedia and Science. It is 20 times larger than the next largest human-annotated OpenIE data, and thus is reliable for fair evaluation.⁶ LSOIE provides n -ary OpenIE annotations and gold tuples are in the $\langle ARG_0, Relation, ARG_1, \dots, ARG_n \rangle$ format. The dataset has two subsets, and we use both, namely LSOIE-wiki and LSOIE-sci, for comprehensive evaluation. LSOIE-wiki has 24,251 sentences and LSOIE-sci has 47,919 sentences.

CaRB (Bhardwaj et al., 2019) dataset is the largest crowdsourced OpenIE dataset.⁷ However, CaRB only provides 1,282 annotated sentences, which are insufficient for training neural OpenIE models. As a result, we use the CaRB dataset purely for testing. We follow Kolluru et al. (2020b) to convert bootstrapped OpenIE4 tuples as labels for distant supervised model training. CaRB provides binary OpenIE annotations and gold tuples are in the form of $\langle Subject, Relation, Object \rangle$. Finally, we summarize the statistics of the training and testing datasets of LSOIE-wiki, LSOIE-sci, and CaRB in Table 2.

⁶<https://github.com/JacobSolawetz/large-scale-oie>

⁷<https://github.com/dair-iitd/CaRB>

5.1 Baselines for Comparison

Baselines without Syntax. RnnOIE (Stanovsky et al., 2018) is the first sequence tagging model based on Bi-LSTM networks. In our work, we implement⁸ its model with GloVe word representation (Pennington et al., 2014), and name it as ‘GloVe+bi-LSTM’. We further add a CRF layer to be another baseline model ‘GloVe+bi-LSTM+CRF’. Meanwhile, we utilize BERT word representation along with its transformer layers, and name this baseline model as ‘BERT’. ‘CopyAttention’ (Cui et al., 2018) is the first neural OpenIE model which casts tuple generation as a sequence generation task. ‘IMOJIE’ (Kolluru et al., 2020a) extends CopyAttention and is able to produce a variable number of extractions per sentence. It iteratively generates the next tuple, conditioned on all previously generated tuples. ‘CIGL-OIE + IGL-CA’ (Kolluru et al., 2020b) models OpenIE as a 2-D grid sequence tagging task and iteratively tags the input sentence until the number of extractions reaches a pre-defined maximum.

Baselines with Syntax. We build baselines that utilize syntactic information, based on BERT. We first study the performance of using either dependency or constituency tree as additional syntactic feature, *i.e.*, ‘BERT+Dep-GCN’ and ‘BERT+Const-GCN’. Then, we present three models of fusing heterogeneous syntactic information from dependency and constituency trees. ‘Dep-GCN \oplus Const-GCN’ refers to the proposed parallel aggregation of the two graph representations using two syntactic GCNs. For comparison, we build a model of sequential aggregation ‘Dep-GCN \rightarrow Const-GCN’, which passes the dependency graph representation from Dep-GCN as input to Const-GCN, and another model ‘Const-GCN \rightarrow Dep-GCN’, which passes the constituency graph representation from Const-GCN as input to Dep-GCN.⁹

5.2 Evaluation

Evaluation Metric. For LSOIE-wiki and LSOIE-sci dataset, Solawetz and Larson (2021) consider two tuples to match if their relations (or verbs) are identical, regardless of the matching of tuple arguments. We consider this scoring function to be over-lenient. Therefore, we revise their scoring function to consider both relation and arguments

⁸Details of implementation are described in Appendix A.2.

⁹Details of experimental setups and re-implementation details are described in Appendix A.1 and A.2.

Models	LSOIE-wiki		LSOIE-sci		CaRB	
	F1	AUC	F1	AUC	F1	AUC
Without Dependency & Constituency Graph						
GloVe + bi-LSTM (Stanovsky et al., 2018)	43.90	38.04	50.51	45.95	48.52	27.10
GloVe + bi-LSTM + CRF	44.48	38.72	50.85	46.23	48.72	27.51
BERT (Solawetz and Larson, 2021)	47.54	44.71	57.02	53.23	51.45	30.62
CopyAttention (Cui et al., 2018)	39.52	35.99	48.82	46.84	51.6 [†]	32.8 [†]
IMoJIE (Kolluru et al., 2020a)	49.24	47.55	58.75	55.81	53.5 [†]	33.3 [†]
CIGL-OIE + IGL-CA (Kolluru et al., 2020b)	44.75	41.98	56.62	52.37	54.0[†]	35.7[†]
With Dependency Graph						
BERT + Dep-GCN	48.71	47.87	58.14	55.32	52.49	32.85
With Constituency Graph						
BERT + Const-GCN	49.74	48.55	58.67	55.76	52.83	33.10
With Dependency & Constituency Graph						
BERT + Dep-GCN \oplus Const-GCN	49.89	49.15	59.23	<u>56.47</u>	52.70	33.28
BERT + Dep-GCN \rightarrow Const-GCN	<u>50.21</u>	<u>49.28</u>	<u>59.53</u>	55.92	52.85	33.37
BERT + Const-GCN \rightarrow Dep-GCN	49.71	48.80	58.81	56.04	53.28	33.51
SMiLe-OIE	51.73	50.88	60.51	57.22	<u>53.76</u>	<u>34.92</u>

Table 3: Results on OpenIE datasets. Scores with [†] are from Kolluru et al. (2020b). The best scores are in boldface, and the second best scores underlined.

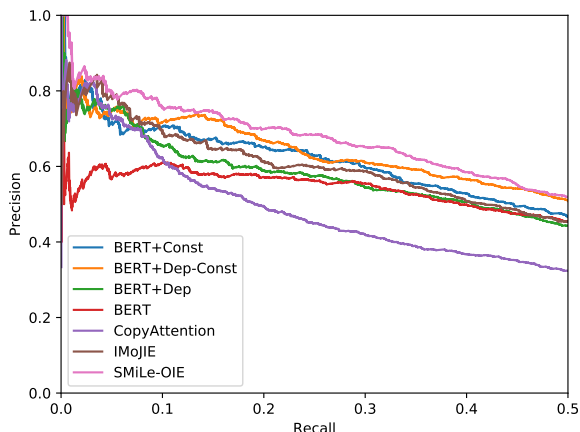


Figure 6: Precision-recall curves of SMiLe-OIE and other baselines on LSOIE-wiki test set.

matching, *i.e.*, exact tuple matching, for accurate and fair comparison. For CaRB dataset, we use the default CaRB scoring function (Bhardwaj et al., 2019) to evaluate binary tuple with lexical level matching, *i.e.*, partial tuple matching. Both scoring functions report F1 score based on precision and recall computed by tuples matching. Each tuple extracted is associated with a confidence value, so we can generate a precision-recall (P-R) curve and report the area under P-R curve (AUC).

5.3 Dataset

Evaluation Results. We compare SMiLe-OIE with other neural OpenIE baseline systems, summarizing their evaluation results in Table 3 and depicting their P-R curves in Figure 6. As shown in Figure 6, SMiLe-OIE achieves better precision at different recalls comparing to other baseline systems. Observe that both “BERT + Dep-GCN” and “BERT + Const-GCN” outperform “BERT”. It shows that leveraging syntactic information, either dependency or constituency tree, benefits OpenIE task significantly.

The comparison results in Table 3 also show that the integration of the *heterogeneous* syntactic information is better than leveraging a *single* syntactic structure. Both parallel and sequential aggregation of the two graph representations achieve better results than “BERT” with either “Dep-GCN” or “Const-GCN”.

Lastly, multi-view learning can effectively guide the fusion of the heterogeneous syntactic information, leading to the significant improvement and outperforming all the baseline systems except “CIGL-OIE + IGL-CA” on CaRB dataset. We find that “CIGL-OIE + IGL-CA” uses a complicated method of coordination boundary analysis dedicated for CaRB dataset. However, the coordination

Models	LSOIE-wiki		LSOIE-sci	
	F1	AUC	F1	AUC
SMiLe-OIE	51.73	50.88	60.51	57.22
- w/o L_{R_1}	50.86	49.23	59.71	56.43
- w/o L_{R_2}	50.24	49.35	58.47	55.72
- w/o L_{R_3}	51.05	50.52	59.84	56.53
w/o GCN	49.57	47.84	58.68	55.67
- w/o L_{R_1}	48.15	46.26	57.22	53.92
- w/o L_{R_2}	48.64	45.59	56.54	54.59
- w/o L_{R_3}	49.83	46.62	57.51	55.24

Table 4: Ablation study of SMiLe-OIE. The best scores are in boldface

boundary analysis of “CIGL-OIE + IGL-CA” cannot be generalized to other datasets, e.g., LSOIE-wiki and LSOIE-sci, and it is thus not preferable.

5.4 Ablation Study

We ablate each part of our model and evaluate the ablated models against the LSOIE-wiki and LSOIE-sci datasets, and the results are reported in Table 4. The upper part of the table reports the ablation study results of removing each of the three multi-view learning losses L_{R_1} , L_{R_2} , and L_{R_3} . It shows that L_{R_2} (intra-node inter-view relationship) has slightly more contribution than the other two losses. The lower part reports the results of removing the GCN layers for dependency and constituency graphs. In this setting, we only concatenate the syntactic label representation to each word, without leveraging the syntactic graph structure. We observe that the GCN layers have larger impact on SMiLe-OIE than multi-view loss, although both contribute to the best performance. Meanwhile, we study a few variants of const-graph, and the results are reported in Appendix 5.5.

5.5 Effectiveness of Const-graph

To verify the effectiveness of the proposed method for converting phrase-level relations of constituency tree into word-level relations of the const-graph (see Section 3.2), we build three variants:

- **Variant 1:** we replace the constituency path, as the label of word node, with the last constituency tag in the path;
- **Variant 2:** in step 2 of constituency relations flattening, we connect a word to the last word of its sibling phrase, instead of connecting to the first word;

BERT+ Const-GCN	LSOIE-wiki		LSOIE-sci	
	F1	AUC	F1	AUC
const-graph	49.74	<u>48.55</u>	58.67	55.76
- variant 1	48.91	48.04	57.73	54.89
- variant 2	<u>49.57</u>	48.67	58.17	<u>55.32</u>
- variant 3	49.25	47.78	<u>58.25</u>	54.03

Table 5: Effectiveness study of const-graph: the best scores are in boldface, and the second best underlined.

- **Variant 3:** in step 4 of constituency relations flattening, we keep edges whose distance between two words is longer than 8. We evaluate the proposed conversion method and its three variants based on our baseline model BERT+Const-GCN. Note that neither Dep-Encoder nor multi-view learning is applied in BERT+Const-GCN.

As shown in Table 5, the proposed method outperforms all the three variants. The const-graph with constituency path outperform its variant 1 that uses a single constituency tag. It proves that using constituency path is better than simply using the last constituency tag. Comparing to variant 2, the const-graph which connects the word to first word of its sibling, achieves better scores. Moreover, we find the keeping distant edges in variant 3 can deteriorate the model performance. As such, it is effective to remove distant edges from const-graph.

6 Conclusion

We design a novel strategy to map constituency tree into constituency graph only with word nodes, paving way for integrating constituency syntax with BERT and dependency syntax. With the aid of Const-GCN and Dep-GCN, we propose a new OpenIE system SMiLe-OIE which combines heterogeneous syntactic information through multi-view learning. Experiment results show that leveraging syntactic information can benefit OpenIE task significantly, and multi-view learning can effectively guide the heterogeneous syntactic information fusion. In future work, we will explore other types of structured information to further improve OpenIE.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046 and #A19E2b0098).

Limitations

We analyze the limitation of our SMiLe-OIE from three perspectives: syntactic parse errors, POS tagging errors and multiple extractions issue. (1) As we integrate both constituency and dependency parsing results with OpenIE task, our system will inevitably suffer from the noises introduced by the off-the-shelf tools: spaCy and CoreNLP. (2) Meanwhile, the number of tuple extractions is highly correlated with the number of verbs extracted by the POS tagger. Therefore, the POS tagger's errors may also affect the quality of OpenIE. Based on our statistics of LSOIE-wiki and LSOIE-sci, the POS tagger fails to extract 8% of verbs that are supposed to be relation indicators. (3) Moreover, there are 6% of relation indicators corresponding to multiple tuple extractions, while our system extracts up to one tuple per relation indicator. Our system, suffering from the POS errors and the multiple extractions issue, fails to predict 14% of the gold tuples.

References

- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. [Leveraging linguistic structure for open domain information extraction](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.
- Sezin Kircali Ata, Yuan Fang, Min Wu, Jiaqi Shi, Chee Keong Kwoh, and Xiaoli Li. 2021. [Multi-view collaborative network embedding](#). *ACM Trans. Knowl. Discov. Data*, 15(3).
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. [Multimodal machine learning: A survey and taxonomy](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.
- Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. [CaRB: A crowdsourced benchmark for open IE](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.
- Nikita Bhutani, H. V. Jagadish, and Dragomir Radev. 2016. [Nested propositions in open information extraction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Austin, Texas. Association for Computational Linguistics.
- Guoqing Chao, Shiliang Sun, and Jinbo Bi. 2021. [A survey on multiview clustering](#). *IEEE Transactions on Artificial Intelligence*, 2(2):146–168.
- Luciano Del Corro and Rainer Gemulla. 2013. [Clausie: clause-based open information extraction](#). In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366. International World Wide Web Conferences Steering Committee / ACM.
- Lei Cui, Furu Wei, and Ming Zhou. 2018. [Neural open information extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kuicai Dong, Zhao Yilin, Aixin Sun, Jung-Jae Kim, and Xiaoli Li. 2021. [DocOIE: A document-level context-aware dataset for OpenIE](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2377–2389, Online. Association for Computational Linguistics.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. [Identifying relations for open information extraction](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. 2021. [Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online. Association for Computational Linguistics.
- Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. [MinIE: Minimizing facts in open information extraction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.
- Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. 2020. [On aligning OpenIE extractions with knowledge bases: A case study](#). In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 143–154, Online. Association for Computational Linguistics.

- Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. [Deep multimodal representation learning: A survey](#). *IEEE Access*, 7:63373–63394.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. [Answering complex questions using open information extraction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Vancouver, Canada. Association for Computational Linguistics.
- Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020a. [IMoJIE: Iterative memory-based joint open information extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.
- Keshav Kolluru, Adlakha Vaibhav, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020b. [OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761, Online. Association for Computational Linguistics.
- Bhushan Kotnis, Kiril Gashteovski, Daniel Rubio, Ammar Shaker, Vanesa Rodriguez-Tembras, Makoto Takamoto, Mathias Niepert, and Carolin Lawrence. 2022. [MILIE: Modular & iterative multilingual open information extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6939–6950, Dublin, Ireland. Association for Computational Linguistics.
- Yingming Li, Ming Yang, and Zhongfei Zhang. 2019. [A survey of multi-view representation learning](#). *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883.
- Zhiheng Lyu, Kaijie Shi, Xin Li, Lei Hou, Juanzi Li, and Binheng Song. 2021. [Multi-grained dependency graph neural network for chinese open information extraction](#). In *Advances in Knowledge Discovery and Data Mining*, pages 155–167, Cham. Springer International Publishing.
- José-Lázaro Martínez-Rodríguez, Ivan López-Arévalo, and Ana B. Ríos-Alvarado. 2018. [Openie-based approach for knowledge graph construction from text](#). *Expert Syst. Appl.*, 113:339–355.
- Mausam. 2016. [Open information extraction systems and downstream applications](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077. IJCAI/AAAI Press.
- Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. [Open language learning for information extraction](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vimojie ectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Dhanesh Ramachandram and Graham W. Taylor. 2017. [Deep multimodal learning: A survey on recent advances and trends](#). *IEEE Signal Processing Magazine*, 34(6):96–108.
- Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. [Supervising unsupervised open information extraction models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 728–737, Hong Kong, China. Association for Computational Linguistics.
- Jacob Solawetz and Stefan Larson. 2021. [LSOIE: A large-scale dataset for supervised open information extraction](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2595–2600, Online. Association for Computational Linguistics.
- Gabriel Stanovsky, Ido Dagan, and Mausam. 2015. [Open IE as an intermediate structure for semantic tasks](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 303–308, Beijing, China. Association for Computational Linguistics.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. [Supervised open information extraction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.
- Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. [Sequence labeling parsing by learning across representations](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5350–5357, Florence, Italy. Association for Computational Linguistics.
- Shiliang Sun. 2013. [A survey of multi-view machine learning](#). *Neural computing and applications*, 23(7):2031–2038.
- Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salkhov, Mikhail Stepnov, Andrey Chertok, and Sergey

Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. [Graph attention networks](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xiaoqiang Yan, Shizhe Hu, Yiqiao Mao, Yangdong Ye, and Hui Yu. 2021. [Deep multi-view learning methods: A review](#). *Neurocomputing*, 448:106–129.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. [TextRunner: Open information extraction on the web](#). In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. [Graph convolution over pruned dependency trees improves relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. 2017. [Multi-view learning overview: Recent progress and new challenges](#). *Information Fusion*, 38:43–54.

Junru Zhou and Hai Zhao. 2019. [Head-Driven Phrase Structure Grammar parsing on Penn Treebank](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Experimental Setups

Dependencies. We build and run our system with Pytorch 1.9.0 and AllenNLP 0.9.0 framework. We collect all verbs from the sentences in all datasets through spaCy¹⁰ POS tagger. In addition, we obtain the constituency annotations through Stanford CoreNLP¹¹ and the dependency annotations through spaCy. We have total 27 types of constituency labels and 45 types of dependency labels.

¹⁰<https://spacy.io/>

¹¹<https://stanfordnlp.github.io/CoreNLP/>

Verb-tuple Alignment. We assume that every tuple has a verb in its relation (see Section 4). However, this assumption does not mean that each verb in a sentence can lead to one tuple. If a sentence contains multiple verbs identified by the POS tagger, we create multiple training instances. In each training instance, one verb is considered as the relation indicator, *i.e.*, its W_{verb} initialized to 1, while W_{verb} for all other verbs in the sentence are initialized to 0. The corresponding tuple taking this verb as relation is the gold label for this training instance. If this verb does not lead to a tuple, we set the label for all words in the sentence to be “O”, *i.e.*, no tuple extracted for this verb. As such, the model is able to learn which verb leads to a tuple extraction. During testing, multiple test instances are created if a sentence contains multiple verbs; one verb serves as a relation indicator in each test instance. No tuple is extracted if all predictions of this test instance are “O”.

Parameters. The hidden dimension d_h for BERT representation h_i^{bert} , Dep-GCN graph representation h_i^{dep} , and Const-GCN graph representation h_i^{con} is 768. We use single-layer GCNs for both constituency and dependency graphs. The hidden dimension d_l for Dep-Encoder type embedding l_i^{dep} and Const-Encoder path embedding l_i^{con} is 400. Hyper-parameters α, β, γ are set to 0.024, 0.012, and 0.012, respectively. Hyper-parameters selection is based on grid searching. The experiments are conducted with Tesla V100 32GB GPU and Intel[®] Xeon[®] Gold 6148 2.40 GHz CPU.

A.2 Re-implementation Details

Note that all baselines are implemented to extract n -ary tuples on LSOIE-wiki and LSOIE-sci datasets, and binary tuples on CaRB dataset. ‘CopyAttention’, ‘IMoJIE’, and ‘CIGL-OIE + IGL-CA’ are binary OpenIE systems and cannot be tested naturally on LSOIE-wiki and LSOIE-sci datasets. We re-implement their models to cater n -ary tuple extraction based on the code repositories.¹² In the evaluation, we evaluate ‘CopyAttention’, ‘IMoJIE’, and ‘CIGL-OIE + IGL-CA’ on LSOIE-wiki and LSOIE-sci datasets through our n -ary re-implementations.

¹²The source code of ‘CopyAttention’ and ‘IMoJIE’ can be found in <https://github.com/dair-iitd/imojie>. The source code of ‘CIGL-OIE + IGL-CA’ can be found in <https://github.com/dair-iitd/openie6>