OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction

Keshav Kolluru[1*], **Vaibhav Adlakha**[1*], **Samarth Aggarwal**[1], **Mausam**[1], and **Soumen Chakrabarti**[2]
[1] Indian Institute of Technology Delhi
keshav.kolluru@gmail.com, vaibhavadlakha95@gmail.com
samarth.aggarwal.2510@gmail.com, mausam@cse.iitd.ac.in
[2] Indian Institute of Technology Bombay
soumen@cse.iitb.ac.in

Abstract

A recent state-of-the-art neural open information extraction (OpenIE) system generates extractions iteratively, requiring repeated encoding of partial outputs. This comes at a significant computational cost. On the other hand, sequence labeling approaches for OpenIE are much faster, but worse in extraction quality. In this paper, we bridge this trade-off by presenting an iterative labeling-based system that establishes a new state of the art for OpenIE, while extracting 10× faster. This is achieved through a novel Iterative Grid Labeling (IGL) architecture, which treats OpenIE as a 2-D grid labeling task. We improve its performance further by applying coverage (soft) constraints on the grid at training time.

Moreover, on observing that the best OpenIE systems falter at handling coordination structures, our OpenIE system also incorporates a new coordination analyzer built with the same IGL architecture. This IGL based coordination analyzer helps our OpenIE system handle complicated coordination structures, while also establishing a new state of the art on the task of coordination analysis, with a 12.3 pts improvement in F1 over previous analyzers. Our OpenIE system, **OpenIE6**[1], beats the previous systems by as much as 4 pts in F1, while being much faster.

1 Introduction

Open Information Extraction (OpenIE) is an ontology-free information extraction paradigm that generates extractions of the form *(subject; relation; object)*. Built on the principles of domain-independence and scalability (Mausam, 2016), OpenIE systems extract open relations and arguments from the sentence, which allow them to be used for a wide variety of downstream tasks like Question Answering (Yan et al., 2018; Khot et al., 2017), Event Schema Induction (Balasubramanian et al., 2013) and Fact Salience (Ponza et al., 2018).

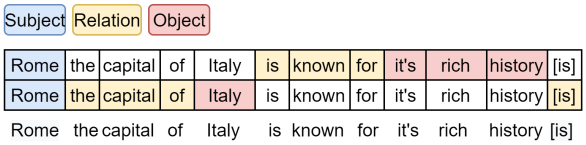

Figure 1: The extractions *(Rome; [is] the capital of; Italy)* and *(Rome; is known for; it's rich history)* can be seen as the output of grid labeling. We additionally introduce a token *[is]* to the input.

End-to-end neural systems for OpenIE have been found to be more accurate compared to their non-neural counterparts, which were built on manually defined rules over linguistic pipelines. The two most popular neural OpenIE paradigms are *generation* (Cui et al., 2018; Kolluru et al., 2020) and *labeling* (Stanovsky et al., 2018; Roy et al., 2019).

Generation systems generate extractions one word at a time. IMoJIE (Kolluru et al., 2020) is a state-of-the-art OpenIE system that re-encodes the partial set of extractions output thus far when generating the next extraction. This captures dependencies among extractions, reducing the overall redundancy of the output set. However, this repeated re-encoding causes a significant reduction in speed, which limits use at Web scale.

On the other hand, *labeling*-based systems like RnnOIE (Stanovsky et al., 2015) are much faster (150 sentences per second, compared to 3 sentences of IMoJIE) but relatively less accurate. They label each word in the sentence as either *S* (Subject), *R* (Relation), *O* (Object) or *N* (None) for each extraction. However, as the extractions are predicted independently, this does not model the inherent dependencies among the extractions.

We bridge this trade-off though our proposed

*Equal Contribution
[1] https://github.com/dair-iitd/openie6

Sentence	Other signs of lens subluxation include mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth .
IGL	(Other signs of lens subluxation; include; mild conjunctival redness, vitreous humour degeneration)
IGL +Constraints	(Other signs of lens subluxation; include; mild conjunctival redness, vitreous humour degeneration, and an increase or decrease of anterior chamber depth)
IGL +Constraints +Coordination Analyzer	(Other signs of lens subluxation; include; mild conjunctival redness) (Other signs of lens subluxation; include; vitreous humour degeneration) (Other signs of lens subluxation; include; an increase of anterior chamber depth) (Other signs of lens subluxation; include; an decrease of anterior chamber depth)

Table 1: For the given sentence, IGL based OpenIE extractor produces an incomplete extraction. Constraints improve the recall by covering the remaining words. Coordination Analyzer handles hierarchical conjunctions.

OpenIE system that is both fast and accurate. It consists of an OpenIE extractor based on a novel iterative labeling-based architecture — **I**terative **G**rid **L**abeling (**IGL**). Using this architecture, OpenIE is modeled as a 2-D grid labeling problem of size (M, N) where M is a pre-defined maximum number of extractions and N is the sentence length, as shown in Figure 1. Each extraction corresponds to one row in the grid. Iterative assignment of labels in the grid helps IGL capture dependencies among extractions without the need for re-encoding, thus making it much faster than generation-based approaches.

While IGL gives high precision, we can further improve recall by incorporating (soft) global coverage constraints on this 2-D grid. We use constrained training (Mehta et al., 2018) by adding a penalty term for all constraint violations. This encourages the model to satisfy these constraints during inference as well, leading to improved extraction quality, without affecting running time.

Furthermore, we observe that existing neural OpenIE models struggle in handling coordination structures, and do not split conjunctive extractions properly. In response, we first design a new coordination analyzer (Ficler and Goldberg, 2016b). It is built with the same IGL architecture, by interpreting each row in the 2-D grid as a coordination structure. This leads to a new state of the art on this task, with a 12.3 pts improvement in F1 over previous best reported result (Teranishi et al., 2019), and a 1.8 pts gain in F1 over a strong BERT baseline.

We then combine the output of our coordination analyzer with our OpenIE extractor, resulting in a further increase in performance (Table 1). Our final OpenIE system — OpenIE6 — consists of IGL-based OpenIE extractor (trained with constraints) and IGL-based coordination analyzer. We evaluate OpenIE6 on four metrics from the literature and find that it exceeds in three of them by at least 4.0 pts in F1. We undertake manual evaluation to reaffirm the gains. In summary, this paper describes OpenIE6, which

- is based on our novel IGL architecture,
- is trained with constraints to improve recall,
- handles conjunctive sentences with our new state-of-art coordination analyzer, which is 12.3 pts better in F1, and
- is 10× faster compared to current state of the art and improves F1 score by as much as 4.0 pts.

2 Related Work

Banko et al. (2007) introduced the Open Information Extraction paradigm (OpenIE) and proposed TextRunner, the first model for the task. Following this, many statistical and rule-based systems have been developed (Fader et al., 2011; Etzioni et al., 2011; Christensen et al., 2011; Mausam et al., 2012; Del Corro and Gemulla, 2013; Angeli et al., 2015; Pal and Mausam, 2016; Stanovsky et al., 2016; Saha et al., 2017; Gashteovski et al., 2017; Saha and Mausam, 2018; Niklaus et al., 2018).

Recently, supervised neural models have been proposed, which are either trained on extractions bootstrapped from earlier non-neural systems (Cui et al., 2018), or on SRL annotations adapted for OpenIE (Stanovsky and Dagan, 2016). These systems are primarily of three types, as follows.

Labeling-based systems like RnnOIE (Stanovsky et al., 2018), and SenseOIE (Roy et al., 2019) identify words that can be syntactic heads of relations, and, for each head word, perform a single labeling to get the extractions. Jiang et al. (2020) extend these to better calibrate confidences across sentences. *Generation-based* systems (Cui et al., 2018; Sun et al., 2018) generate extractions sequentially using seq2seq models. IMoJIE (Kolluru et al., 2020), the current state of art in OpenIE, uses a BERT-based encoder and an iterative decoder that re-encodes the extractions generated so far. This re-encoding captures

dependencies between extractions, increasing overall performance, but also makes it 50x slower than RnnOIE. Recently, *span-based* models (Jiang et al., 2020) have been proposed, e.g., SpanOIE (Zhan and Zhao, 2020), which uses a predicate module to first choose potential candidate relation spans, and for each relation span, classifies all possible spans of the sentence as subject or object.

Concurrent to our work (Ro et al., 2020) proposed Multi^{2}OIE, a sequence-labeling model for OpenIE, which first predicts all the relation arguments using BERT, and then predicts subject and object arguments associated with each relation using multi-head attention blocks. Their model cannot handle nominal relations and conjunctions in arguments, which can be extracted in our iterative labeling scheme.

OpenIE Evaluation: Several datasets have been proposed to automatically evaluate OpenIE systems. OIE2016 (Stanovsky and Dagan, 2016) introduced an automatically generated reference set of extractions, but it was found to be too noisy with significant missing extractions. Re-OIE2016 (Zhan and Zhao, 2020) manually re-annotated the corpus, but did not handle conjunctive sentences adequately. Wire57 (Léchelle et al., 2018) contributed high-quality expert annotations, but for a small corpus of 57 sentences. We use the CaRB dataset (Bhardwaj et al., 2019), which re-annotated OIE2016 corpus via crowd-sourcing.

The benchmarks also differ in their scoring functions along two dimensions: (1) computing similarity for each (gold, system) extraction pair, (2) defining a mapping between system and gold extractions using this similarity. OIE16 computes similarity by serializing the arguments into a sentence and finding the number of matching words. It maps each system extraction to one gold (one-to-one mapping) to compute both precision and recall. Wire57 uses the same one-to-one mapping but computes similarity at an argument level. CaRB uses one-to-one mapping for precision but maps multiple gold to the same system extraction (many-to-one mapping) for recall. Like Wire57, CaRB computes similarity at an argument level.

OpenIE for Conjunctive Sentences: Performance of OpenIE systems can be further improved by identifying coordinating structures governed by conjunctions (e.g., 'and'), and splitting conjunctive extractions (see Table 1). We follow CalmIE (Saha and Mausam, 2018), which is part of OpenIE5 system – it splits a conjunctive sentence into smaller sentences based on detected coordination boundaries, and runs OpenIE on these split sentences to increase overall recall.

For detecting coordination boundaries, Ficler and Goldberg (2016a) re-annotate the Penn Tree Bank corpus with coordination-specific tags. Neural parsers trained on this data use similarity and replacability of conjuncts as features (Ficler and Goldberg, 2016b; Teranishi et al., 2017). The current state-of-the-art system (Teranishi et al., 2019) independently detects coordinator, begin, and end of conjuncts, and does joint inference using Cocke–Younger–Kasami (CYK) parsing over context-free grammar (CFG) rules. Our end-to-end model obtains better accuracy than this approach.

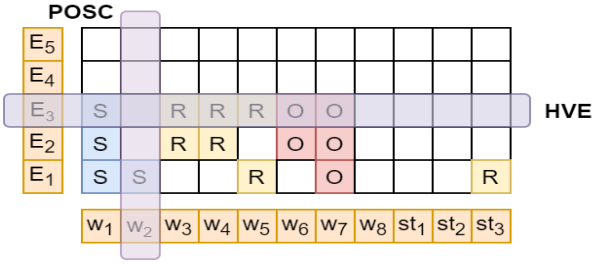

Figure 2: 2-D grid for OpenIE with extraction as rows and words as columns. The values represent the labels (S)ubject, (R)elation, (O)bject. The empty cells represent (N)one. Constraints are applied across rows (**HVE**) and columns (**POSC**).

Constrained Training: Constraining outputs of the model is a way to inject prior knowledge into deep neural networks (Hu et al., 2016; Xu et al., 2018; Nandwani et al., 2019). These constraints can be applied either during training or inference or both. We follow Mehta et al. (2018), which models an output constraint as a differentiable penalty term defined over output probabilities given by the network. This penalty is combined with the original loss function for better training.

Bhutani et al. (2019) propose an OpenIE system to get extractions from question-answer pairs. Their decoder enforces vocabulary and structural constraints on the output both during training and inference. In contrast, our system uses constraints only during training.

3 Iterative Grid Labeling for OpenIE

Given a sentence with word tokens $\{w_1, w_2, \ldots, w_N\}$ the task of OpenIE is to output a set of extractions, say $\{E_1, E_2, \ldots, E_M\}$,

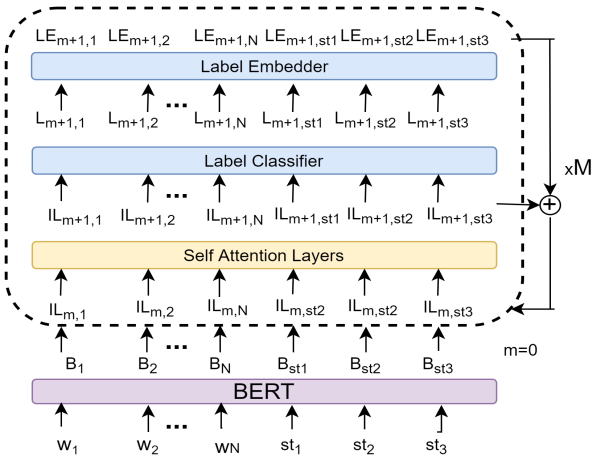

Figure 3: Model architecture for IGL. BERT-embeddings of the words are iteratively passed through self-attention layers. st_1, st_2, st_3 refer to the appended tokens *[is]*, *[of]*, *[from]*, respectively. At every iteration, we get an extraction by labeling the words using a fully-connected layer. Embeddings of the generated labels are added to the iterative layer embeddings before passing them to the next iteration.

where each extraction is of the form *(subject; relation; object)*. For a labeling-based system, each word is labeled as *S* (Subject), *R* (Relation), *O* (Object), or *N* (None) for every extraction. We model this as a 2-D grid labeling problem of size (M, N), where the words represent the columns and the extractions represent the rows (Figure 2). The output at position (m, n) in the grid $(L_{m,n})$ represents the label assigned to the n^{th} word in the m^{th} extraction.

We propose a novel **I**terative **G**rid **L**abeling (IGL) approach to label this grid, filling up one row after another iteratively. We refer to the OpenIE extractor trained using this approach as IGL-OIE.

IGL-OIE is based on a BERT encoder, which computes contextualized embeddings for each word. The input to the BERT encoder is $\{w_1, w_2, \ldots, w_N, [is], [of], [from]\}$. The last three tokens (referred as st_i in Figure 3) are appended because, sometimes, OpenIE is required to predict tokens that are not present in the input sentence.[2] E.g., "*US president Donald Trump gave a speech on Wednesday.*" will have one of the extractions as *(Donald Trump; [is] president [of]; US)*. The appended tokens make such extractions possible in a labeling framework.

The contextualized embeddings for each word or appended token are iteratively passed through a 2-layer transformer to get their *IL embeddings* at different levels, until a maximum level M, i.e. a word w_n has a different contextual embedding $IL_{m,n}$ for every row (level) m. At every level m, each $IL_{m,n}$ is passed though a fully-connected labeling layer to get the labels for words at that level (Figure 3). Embeddings of the predicted labels are added to the *IL embeddings* before passing them to the next iteration. This, in principle, maintains the information of the extractions output so far, and hence can capture dependencies among labels of different extractions. For words that were broken into word-pieces by BERT, only the embedding of the first word-piece is retained for label prediction. We sum the cross-entropy loss between the predicted labels and the gold labels at every level to get the final loss, denoted by J_{CE}.

OpenIE systems typically assign a confidence value to an extraction. In IGL, at every level, the respective extraction is assigned a confidence value by adding the log probabilities of the predicted labels (*S*, *R*, and *O*), and normalizing this by the extraction length.

We believe that IGL architecture has value beyond OpenIE, and can be helpful in tasks where a set of labelings for a sentence is desired, especially when labelings have dependencies amongst them.[3] We showcase another application of IGL for the task of coordination analysis in Section 5.

4 Grid Constraints

Our preliminary experiments revealed that IGL-OIE has good precision, but misses out important extractions. In particular, we observed that the set of output extractions did not capture all the information from the sentence (Table 1). We formulate constraints over the 2-D grid of extractions (as shown in Figure 2) which act as an additional form of supervision to improve the coverage. We implement these as soft constraints, by imposing additional violation penalties in the loss function. This biases the model to learn to satisfy the constraints, without explicitly enforcing them at inference time.

To describe the constraints, we first define the notion of a *head verb* as all verbs except light verbs (do, be, is, has, etc.). We run a POS tagger on the input sentence, and find all head verbs in the sentence by removing all light verbs.[4] For example,

[2] 'is', 'of' and 'from' are the most frequent such tokens.

[3] IGL is a generalization of Ju et al. (2018). Their model can only label spans which are subsets of one another.

[4] We used the light verbs listed by Jain and Mausam (2016).

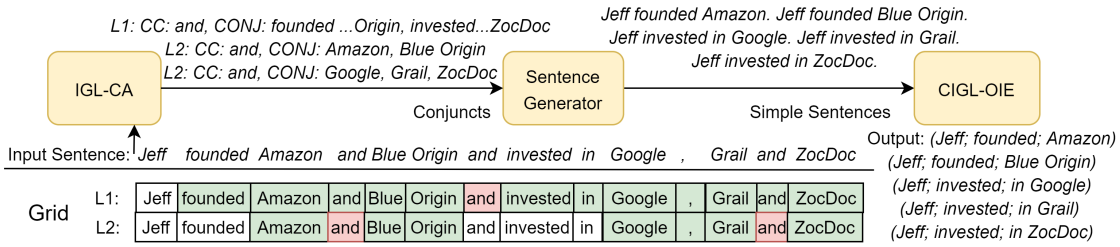

Figure 4: The final OpenIE system. IGL-CA identifies conjunct boundaries by labeling a 2-D grid. This generates simple sentences and CIGL-OIE emits the final extractions.

for the sentence, *"Obama gained popularity after Oprah endorsed him for the presidency"*, the head verbs are *gained* and *endorsed*. In order to cover all valid extractions like *(Obama; gained; popularity)* and *(Oprah; endorsed him for; the presidency)*, we design the following coverage constraints:

- *POS Coverage* (**POSC**): All words with POS tags as nouns (N), verbs (V), adjectives (JJ), and adverbs (RB) should be part of at least one extraction. E.g. the words *Obama*, *gained*, *popularity*, *Oprah*, *endorsed*, *presidency* must be covered in the set of extractions.
- *Head Verb Coverage* (**HVC**): Each head verb should be present in the relation span of some (but not too many) extractions. E.g. *(Obama; gained; popularity)*, *(Obama; gained; presidency)* is not a comprehensive set of extractions.
- *Head Verb Exclusivity* (**HVE**): The relation span of one extraction can contain at most one head verb. E.g. *gained popularity after Oprah endorsed* is not a good relation as it contains two head verbs.
- *Extraction Count* (**EC**): The total number of extractions with head verbs in the relation span must be no fewer than the number of head verbs in the sentence. In the example, there must be at least two extractions containing head verbs, as the sentence itself has two head verbs.

Notation: We now describe the penalty terms for these constraints. Let p_n be the POS tag of w_n. We define an indicator $x_n^{imp} = 1$ if $p_n \in \{\text{N, V, JJ, RB}\}$, and 0 otherwise. Similarly, let $x_n^{hv} = 1$ denote that w_n is a head verb. At each extraction level m, the model computes $Y_{mn}(k)$, the probability of assigning the n^{th} word the label $k \in \{\text{S, R, O, N}\}$. We formulate the penalties associated with our constraints as follows:

- **POSC** - To ensure that the n^{th} word is covered, we compute its maximum probability ($posc_n$) of belonging to any extraction. We introduce a penalty if this value is low. This penalty is aggregated over words with important POS tags, $J_{posc} = \sum_{n=1}^{N} x_n^{imp} \cdot posc_n$, where
$$posc_n = 1 - \max_{m \in [1,M]} \left(\max_{k \in \{\text{S,R,O}\}} Y_{mn}(k) \right)$$
- **HVC** - A penalty is imposed for the n^{th} word, if it is not present in relation of any extraction or if it is present in relation of many extractions. This penalty is aggregated over head verbs, $J_{hvc} = \sum_{n=1}^{N} x_n^{hv} \cdot hvc_n$, where $hvc_n = \left| 1 - \sum_{m=1}^{M} Y_{mn}(R) \right|$.
- **HVE** - A penalty is imposed if the relation span of an extraction contains more than one head verb. This penalty is summed over all extractions. I.e., $J_{hve} = \sum_{m=1}^{M} hve_m$, where
$$hve_m = \max \left(0, \left(\sum_{n=1}^{N} x_n^{hv} \cdot Y_{mn}(R) \right) - 1 \right)$$
- **EC** - ec_m denotes the score $\in [0, 1]$ of the m^{th} extraction containing a head verb, i.e. $ec_m = \max_{n \in [1,N]} \left(x_n^{hv} \cdot Y_{mn}(R) \right)$. A penalty is imposed if the sum of these scores is less than the actual number of head verbs in the sentence.
$$J_{ec} = \max \left(0, \sum_{n=1}^{N} x_n^{hv} - \sum_{m=1}^{M} ec_m \right)$$

Ideally, no constraint violations of **HVC** and **HVE** would imply that **EC** would also never gets violated. However, as these are soft constraints, this scenario is never materialized in practice. We find that our model performs better and results in fewer constraint violations when trained with **POSC**, **HVC**, **HVE** and **EC** combined. The full loss function is $J = J_{CE} + \lambda_{posc} J_{posc} + \lambda_{hvc} J_{hvc} + \lambda_{hve} J_{hve} + \lambda_{ec} J_{ec}$, where $\lambda_\star$ are hyperparameters. We refer to the OpenIE extractor trained using this constrained loss as Constrained Iterative Grid Labeling OpenIE Extractor (CIGL-OIE).

The model is initially trained without constraints for a fixed *warmup* number of iterations, followed by constrained training till convergence.

5 Coordination Boundary Detection

Coordinated conjunctions (CC) are conjunctions such as *"and"*, *"or"* that connect, or coordinate words, phrases, or clauses (they are called the conjuncts). The goal of coordination analysis is to detect coordination structures — the coordinating conjunctions along with their constituent conjuncts. In this section we build a novel coordination analyzer and use its output downstream for OpenIE.

Sentences can have hierarchical coordinations, i.e., some coordination structures nested within the conjunct span of others (Saha and Mausam, 2018). Therefore, we pose coordination analysis as a hierarchical labeling problem, as illustrated in Figure 4. We formulate a 2-D grid labeling problem, where all coordination structures at the same hierarchical level are predicted in the same row.

Specifically, we define a grid of size (M, N), where M is the maximum depth of hierarchy and N is the number of words in the sentence. The value at $(m, n)^{th}$ position in the grid represents the label assigned to the n^{th} word in the m^{th} hierarchical level, which can be *CC* (coordinated conjunction), *CONJ* (belonging to a conjunct span), or *N* (None). Using IGL architecture for this grid gives an end-to-end Coordination Analyzer that can detect multiple coordination structures, with two or more conjuncts. We refer to this Coordination Analyzer as IGL-CA.

Coordination Analyzer in OpenIE: Conjuncts in a coordinate structure exhibit *replaceability* – a sentence is still coherent and consistent, if we replace a coordination structure with any of its conjuncts (Ficler and Goldberg, 2016b). Following CalmIE's approach, we generate simple (non-conjunctive) sentences using IGL-CA. We then run CIGL-OIE on these simple sentences to generate extractions. These extractions are de-duplicated and merged to yield the final extraction set (Figure 4). This pipelined approach describes our final OpenIE system — **OpenIE6**.

For a conjunctive sentence, CIGL-OIE's confidence values for extractions will be with respect to multiple simple sentences, and may not be calibrated across them. We use a separate confidence estimator, consisting of a BERT encoder and an LSTM decoder trained on (sentence, extraction) pairs. It computes a log-likelihood for every extraction w.r.t. the original sentence — this serves as a better confidence measure for OpenIE6.

6 Experimental Setup

We train OpenIE6 using the OpenIE4 training dataset used to train IMoJIE[5]. It has 190,661 extractions from 92,774 Wikipedia sentences. We convert each extraction to a sequence of labels over the sentence. This is done by looking for an exact string match of the words in the extraction with the sentence. In case there are multiple string matches for one of the arguments of the extraction, we choose the string match closest to the other arguments. This simple heuristic covers almost 95% of the training data. We ignore the remaining extractions that have multiple string matches for more than one argument.

We implement our models using Pytorch Lightning (Falcon, 2019). We use pre-trained weights of "BERT-base-cased"[6] for OpenIE extractor and "BERT-large-cased"[6] for coordination analysis. We do not use BERT-large for OpenIE extractor as we observe almost same performance with a significant increase in computational costs. We set the maximum number of iterations, M=5 for OpenIE and M=3 for Coordination Analysis. We use the SpaCy POS tagger[7] for enforcing constraints. The various hyper-parameters used are mentioned in Appendix B.

Comparison Systems: We compare OpenIE6 against several recent neural and non-neural systems. These include generation (IMoJIE and Cui et al. (2018)[8]), labeling (RnnOIE, SenseOIE) and span-based (SpanOIE) systems. We also compare against non-neural baselines of MinIE (Gashteovski et al., 2017), ClausIE (Del Corro and Gemulla, 2013), OpenIE4 (Christensen et al., 2011)[9] and OpenIE5 (Saha et al., 2017; Saha and Mausam, 2018).[10] We use open-source implementations for all systems except SenseOIE, for which the code is not available and we use the system output provided by the authors.

Evaluation Dataset and Metrics: We evaluate all systems against CaRB's reference extractions, as they have higher coverage and quality compared to other datasets. Apart from CaRB's scoring function, we also use scoring functions of OIE16 and

[5] Available from github:dair-iitd/imojie
[6] github:huggingface/transformers
[7] https://spacy.io
[8] We use the BERT implementation available at github:dair-iitd/imojie
[9] github:allenai/openie-standalone
[10] github:dair-iitd/openie-standalone

System	CaRB		CaRB(1-1)		OIE16-C		Wire57-C	Speed
	F1	AUC	F1	AUC	F1	AUC	F1	Sentences/sec.
MinIE	41.9	-	38.4	-	52.3	-	28.5	8.9
ClausIE	45.0	22.0	40.2	17.7	61.0	38.0	33.2	4.0
OpenIE4	51.6	29.5	40.5	20.1	54.3	37.1	34.4	20.1
OpenIE5	48.0	25.0	42.7	20.6	59.9	39.9	35.4	3.1
SenseOIE	28.2	-	23.9	-	31.1	-	10.7	-
SpanOIE	48.5	-	37.9	-	54.0	-	31.9	19.4
RnnOIE	49.0	26.0	39.5	18.3	56.0	32.0	26.4	**149.2**
(Cui et al., 2018)	51.6	32.8	38.7	19.8	53.5	37.0	33.3	11.5
IMoJIE	53.5	33.3	41.4	22.2	56.8	39.6	36.0	2.6
IGL-OIE	52.4	33.7	41.1	22.9	55.0	36.0	34.9	142.0
CIGL-OIE	**54.0**	**35.7**	42.8	24.6	59.2	40.0	36.8	142.0
CIGL-OIE + IGL-CA (OpenIE6)	52.7	33.7	**46.4**	**26.8**	**65.6**	**48.4**	**40.0**	31.7

Table 2: Evaluation of OpenIE. Using constrained learning, CIGL-OIE gives better scores on all metrics compared to IMoJIE. Adding a coordination analyzer, CIGL-OIE + IGL-CA (OpenIE6) gives the best scores in 3 of the 4 metrics. MinIE, SenseOIE, SpanOIE do not output confidences. Code of SenseOIE is not available to compute speed.

System	Precision	Yield	Total Extrs
CIGL-OIE	77.9	131	174
OpenIE6	**78.8**	**222**	**291**

Table 3: Manual comparison of Precision and Yield on 100 random conjunctive sentences from CaRB Gold.

Wire57 benchmarks on the CaRB reference set, which we refer to as *OIE16-C* and *Wire57-C*. Additionally we use *CaRB(1-1)*, a variant of CaRB that retains CaRB's similarity computation, but uses a one-to-one mapping for both precision and recall (similar to OIE16-C, Wire57-C).

For each system, we report a final F1 score using precision and recall computed by these scoring functions. OpenIE systems typically associate a confidence value with each extraction, which can be varied to generate a precision-recall (P-R) curve. We also report the area under P-R curve (AUC) for all scoring functions except Wire57-C, as its matching algorithm is not naturally compatible with P-R curves. We discuss details of these four metrics in Appendix A.

For determining the speed of a system, we analyze the number of sentences it can process per second. We run all the systems on a common set of 3,200 sentences (Stanovsky et al., 2018), using a V100 GPU and 4 cores of Intel Xeon CPU (the non-neural systems use only the CPU).

7 Experiments and Results

7.1 Speed and Performance

How does OpenIE6 compare in speed and performance?

Table 2 reports the speed and performance comparisons across all metrics for OpenIE. We find that the base OpenIE extractor — IGL-OIE — achieves a 60× speed-up compared to IMoJIE, while being lower in performance by 1.1 F1, and better in AUC by 0.4 pts, when using CaRB scoring function.

We find that training IGL-OIE along with constraints (CIGL-OIE), helps to improve the performance without affecting inference time. This system is better than all previous systems over all the considered metrics. It beats IMoJIE by (0.5, 2.4) in CaRB (F1, AUC) and 0.8 F1 in Wire57-C.

Further, adding the coordination analyzer module (IGL-CA) gives us OpenIE6, which is 10× faster than IMoJIE (32 sentences/sec) and achieves significant improvements in performance in 3 of the 4 metrics considered. It improves upon IMoJIE in F1 by 5.0, 8.8, 4.0 pts in CaRB(1-1), OIE16-C and Wire57-C, respectively. However, in the CaRB metric, adding this module leads to a decrease of (1.5, 0.9) pts in (F1, AUC).

On closer analysis, we notice that the current scoring functions for OpenIE evaluation do not handle conjunctions properly. CaRB over-penalizes OpenIE systems for incorrect coordination splits whereas other scoring functions under-penalize them. This is also evidenced in the lower CaRB scores of for both OpenIE-5[11] (vs. OpenIE4) and

[11] OpenIE5 uses CalmIE for conjunctive sentences.

System	Wire57-C	CaRB		Constraint Violations					Num. of Extrs
	F1	F1	AUC	POSC	HVC	HVE	EC	HVC+HVE+EC	
IMoJIE	36.0	53.5	33.3	687	521	105	330	957	1354
IGL-OIE	34.9	52.4	33.7	1494	375	**128**	284	787	1401
IGL-OIE (POSC)	36.7	49.6	33.4	**396**	303	200	**243**	746	**1577**
IGL-OIE (HVC,HVE,EC)	35.8	53.2	32.7	1170	295	144	246	**655**	1509
CIGL-OIE	**36.8**	**54.0**	**35.7**	766	**274**	157	237	668	1531
Gold	100	100	100	371	324	272	224	820	2714

Table 4: Performance and number of constraint violations for training with different sets of constraints. CIGL-OIE represents training IGL architecture based OpenIE extractor with all the constraints - POSC, HVC, HVE and EC

OpenIE6 (vs. CIGL-OIE) — the two systems that focus on conjunctive sentences. We trace this issue to the difference in mapping used for recall computation (one-to-one vs many-to-one). We refer the reader to Appendix A.3 for a detailed analysis of this issue.

To resolve this variation in different scoring functions, we undertake a manual evaluation. Two annotators (authors of the paper), blind to the underlying systems (CIGL-OIE and OpenIE6), independently label each extraction as correct or incorrect for a subset of 100 conjunctive sentences. Their inter-annotator agreement is 93.46% (See Appendix C for details of manual annotation setup). After resolving the extractions where they differ, we report the precision and yield in Table 3. Here, yield is the number of correct extractions generated by a system. It is a surrogate for recall, since its denominator, number of all correct extractions, is hard to annotate for OpenIE.

We find that OpenIE6 significantly increases the yield (1.7×) compared to CIGL-OIE along with a marginal increase in precision. This result underscores the importance of splitting coordination structures for OpenIE.

7.2 Constraints Ablation

How are constraint violations related to model performance?

We divide the constraints into two groups: one which is dependent on head verb(s): {HVC, HVE and EC}, and the other which is not – POSC. We separately train IGL architecture based OpenIE extractor with these two groups of constraints, and compare them with no constraints (IGL-OIE), all constraints (CIGL-OIE) and IMoJIE. In Table 4, we report the performance on Wire57-C and CaRB, and also report the number of constraint violations in each scenario.

Training IGL architecture based OpenIE extractor with POSC constraint (IGL-OIE (POSC)), leads to a reduction in POSC violations. However, the number of violations of (HVC+HVE+EC) remains high. On the other hand, training only with head verb constraints (HVC,HVE,EC) reduces their violations but the POSC violations remains high. Hence, we find that training with all the constraints achieves the best performance. Compared to IGL-OIE, it reduces the POSC violation from 1494 to 766 and (HVC+HVE+EC) violations from 787 to 668. The higher violations of Gold may be attributed to an overall larger number of extractions in the reference set.

7.3 Coordination Analysis

How does our coordination analyzer compare against other analyzers? How much does the coordination analyzer benefit OpenIE systems?

Following previous works (Teranishi et al., 2017, 2019), we evaluate two variants of our IGL architecture based coordination analyzer (IGL-CA) – using BERT-Base and BERT-Large, on coordination-annotated Penn Tree Bank (Ficler and Goldberg, 2016a). We compute the Precision, Recall and F1 of the predicted conjunct spans. In Table 5, we find that both BERT-Base and BERT-Large variants outperform the previous state-of-art (Teranishi et al., 2019) by 9.4 and 12.3 F1 points respectively. For fair comparison, we train a stronger variant of Teranishi et al. (2019), replacing the LSTM encoder with BERT-Base and BERT-Large. Even in these settings, IGL-CA performs better by 1.8 and 1.3 F1 points respectively, highlighting the significance of our IGL architecture. Overall, IGL-CA establishes a new state of the art for this task.

To affirm that the gains of better coordination analysis help the downstream OpenIE task, we experiment with using different coordination analyzers with CIGL-OIE and IMoJIE. From Table 6, we see a considerable improvement in the downstream

System	Precision	Recall	F1
(Teranishi et al., 2017)	71.5	70.7	71.0
(Teranishi et al., 2019)	75.3	75.6	75.5
BERT-Base:			
(Teranishi et al., 2019)	83.1	83.2	83.1
IGL-CA	86.3	83.6	84.9
BERT-Large:			
(Teranishi et al., 2019)	86.4	86.6	86.5
IGL-CA	**88.1**	**87.4**	**87.8**

Table 5: P, R, F1 of the system evaluated on Penn Tree Bank for different systems. We use both BERT-Base and BERT-Large as the encoder

Coordination Analyzer	IMoJIE	CIGL-OIE
None	36.0	36.8
CalmIE	37.7	38.0
(Teranishi et al., 2019)	36.1	36.5
IGL-CA	**39.5**	**40.0**

Table 6: Wire57 F1 scores of IMoJIE and CIGL-OIE with addition of different coordination analyzers. IGL-CA improves both of the OpenIE extractors.

OpenIE task using IGL-CA for both IMoJIE and CIGL-OIE, which we attribute to better conjunct-boundary detection capabilities of the model. For CIGL-OIE, this gives a 2 pts increase in Wire57-C F1, compared to CalmIE's coordination analyzer (CalmIE-CA).

8 Error Analysis

We examine extractions from a random sample of 50 sentences from CaRB validation set, as output by OpenIE6. We identify three major sources of errors in these sentences:

Grammatical errors: (24%) We find that the sentence formed by serializing the extraction is not grammatically correct. We believe that combining our extractor with a pre-trained language model might help reduce such errors.

Noun-based relations: (16%) These involve introducing additional words in the relation span. Although our model can introduce *[is], [of], [from]* in relations (Section 3), it may miss some words for which it was not trained. E.g. *[in]* in *(First Security; based [in]; Salt Lake City)* for the phrase *Salt Lake City-based First Security*.

Lack of Context: (10%) Neural models for OpenIE including ours, do not output extraction context (Mausam et al., 2012). E.g. for "*She believes aliens will destroy the Earth*", the extraction *(Context(She believes); aliens; will destroy; the Earth)* can be misinterpreted without the context.

We also observe incorrect boundary identification for relation argument (13%), cases in which coordination structure in conjunctive sentences are incorrectly split (11%), lack of coverage (4%) and other miscellaneous errors (18%).

9 Conclusion

We propose a new OpenIE system – OpenIE6, based on the novel Iterative Grid Labeling architecture, which models sequence labeling tasks with overlapping spans as a 2-D grid labeling problem. OpenIE6 is 10x faster, handles conjunctive sentences and establishes a new state of art for OpenIE. We highlight the role of constraints in training for OpenIE. Using the same architecture, we achieve a new state of the art for coordination parsing, with a 12.3 pts improvement in F1 over previous analyzers. We plan to explore the utility of this architecture in other NLP problems. OpenIE6 is available at `https://github.com/dair-iitd/openie6` for further research.

Acknowledgements

We thank the anonymous reviewers for their suggestions and feedback. Mausam is supported by IBM AI Horizons Network grant, an IBM SUR award, grants by Google, Bloomberg and 1MG, Jai Gupta Chair Fellowship and Visvesvaraya faculty award by Govt. of India. We thank IIT Delhi HPC facility for compute resources. Soumen was partly supported by a Jagadish Bose Fellowship and an AI Horizons Network grant from IBM.

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 of the Asian Federation of Natural Language Processing (ACL), 2015*, pages 344–354.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731, Seattle, Washington, USA. Association for Computational Linguistics.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007.

Open information extraction from the web. In *International Joint Conference on Artificial Intelligence (IJCAI), 2007*, volume 7, pages 2670–2676.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. 2019. CaRB: A Crowdsourced Benchmark for OpenIE. 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), 2019*, pages 6263–6268.

Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and H. V. Jagadish. 2019. Open information extraction from question-answer pairs. 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 2294–2305, Minneapolis, Minnesota. Association for Computational Linguistics.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120. ACM.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of Association for Computational Linguistics (ACL), 2018*, pages 407–413.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web (WWW), 2013*, pages 355–366. ACM.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open Information Extraction: The Second Generation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 3–10. IJCAI/AAAI.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, Edinburgh, Scotland, UK.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/williamFalcon/pytorch-lightning*.

Jessica Ficler and Yoav Goldberg. 2016a. Coordination annotation extension in the penn tree bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Jessica Ficler and Yoav Goldberg. 2016b. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 23–32. The Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: minimizing facts in open information extraction. In *Association for Computational Linguistics (ACL), 2017*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Prachi Jain and Mausam. 2016. Knowledge-guided linguistic rewrites for inference rule verification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 86–92.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020. Generalizing natural language analysis through span-relation representations. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seattle, U.S.A.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 311–316. Association for Computational Linguistics.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seattle, U.S.A.

William Léchelle, Fabrizio Gotti, and Philippe Langlais. 2018. Wire57 : A fine-grained benchmark for open information extraction. In *LAW@ACL*.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), 2016*, pages 4074–4077. AAAI Press.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, 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. Association for Computational Linguistics.

Sanket Vaibhav Mehta, Jay Yoon Lee, and Jaime G. Carbonell. 2018. Towards semi-supervised learning for deep semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4958–4963. Association for Computational Linguistics.

Yatin Nandwani, Abhishek Pathak, Parag Singla, and Mausam. 2019. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems*, pages 12157–12168.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal OpenIE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39.

Marco Ponza, Luciano Del Corro, and Gerhard Weikum. 2018. Facts that matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1043–1048, Brussels, Belgium. Association for Computational Linguistics.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multi^{2}OIE: Multilingual Open Information Extraction Based on Multi-Head Attention with BERT. In *Findings of ACL: EMNLP 2020*.

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.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical OpenIE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas. Association for Computational Linguistics.

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

Gabriel Stanovsky, Mausam, and Ido Dagan. 2015. OpenIE 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.

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 (NAACL-HLT), Volume 1 (Long Papers)*, pages 885–895.

Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018. Logician: A unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 556–564.

Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Coordination boundary identification with similarity and replaceability. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 264–272.

Hiroki Teranishi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Decomposed local models for coordinate structure parsing. 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 3394–3403, Minneapolis, Minnesota. Association for Computational Linguistics.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2018. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR.

Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. 2018. Assertion-based QA with question-aware open information extraction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6021–6028. AAAI Press.

Junlang Zhan and Hai Zhao. 2020. Span Model for Open Information Extraction on Accurate Corpus. In *AAAI Conference on Artificial Intelligence, 2020*, pages 5388–5399.

A Metrics

A.1 Introduction

Designing an evaluation benchmark for an underspecified and subjective task like OpenIE has gathered much attention. Several benchmarks, consisting of gold labels and scoring functions have been contributed. While coverage and quality of gold labels of these benchmarks have been extensively studied, differences in their scoring functions is largely unexplored. We evaluate all our systems on the CaRB reference set, which has 641 sentences and corresponding human annotated extractions in both dev and test set. As the underlying gold labels, is the same, system performances differ only due to difference in design choices of these scoring functions, which we explore in detail here.

A.2 Scoring Functions of Benchmarks

OIE2016[12] creates a one-to-one mapping between (gold, system) pairs by serializing the extractions and comparing the number of common words within them. Hence the system is not penalized for misidentifying parts of an one argument in another. Precision and recall for the system are computed using the one-to-one mapping obtained, i.e. precision is (no. of system extractions mapped to gold extractions)/ (total no. of system extractions) and recall is (no. of gold extractions mapped to system extractions)/(total no. of gold extractions). These design choices have several implications (Léchelle et al., 2018; Bhardwaj et al., 2019). Overlong system extractions which are mapped, are not penalized, and extractions with partial coverage of gold extractions, which are not mapped, are not rewarded at all.

Wire57[13] attempts to tackle the shortcomings of OIE2016. For each gold extraction, a set of candidate system extractions are chosen on the basis of whether they share at least one word for each of the arguments[14] of the extraction, with the gold. It then creates a one-to-one mapping by greedily matching gold with one of the candidate system extraction on the basis of token-level F1 score. Token level precision and recall of the matches are then aggregated to get the score for the system. Computing scores at token level helps in penalizing overly long extractions.

Wire57 ignores the confidence of extraction and reports just the F1 score (F1 at zero confidence). One way to generate AUC for Wire57 is by obtaining precision and recall scores at various confidence levels by passing a subset of extractions to the scorer. However, due to Wire57's criteria of matching extractions on the basis of F1 score, the recall of the system does not decrease monotonically with increasing confidence, which is a requirement for calculating AUC.

OIE2016 and Wire57 both use one-to-one mapping strategy, due to which a system extraction, that contains information from multiple gold extractions, is unfairly penalized.

CaRB[15] also computes similarity at a token level, but it is slightly more lenient than Wire57 — it considers number of common words in (gold,system) pair for each argument of the extraction. However, it uses one-to-one mapping for precision and many-to-one mapping for computing recall. While this solves the issue of penalizing extractions with information from multiple gold extractions, it inadvertently creates another one — unsatisfactorily evaluating systems which split on conjunctive sentences. We explore this in detail in the next section.

A.3 CaRB on Conjunctive Sentences

Coordinate structure in conjunctive sentences are of two types:

- *Combinatory*, where splitting the sentence by replacing the coordinate structure with one of the conjuncts can lead to incoherent extractions. E.g. splitting "*Talks resumed between USA and China*" will give *(Talks; resumed; between USA)*.
- *Segregatory*, where splitting on coordinate structure can lead to shorter and coherent extractions. E.g. splitting "*I ate an apple and orange.*" gives *(I; ate; an apple)* and *(I; ate; an orange)*.

Combinatory coordinate structures are hard to detect (in some cases even for humans). Some systems (ClausIE, CalmIE and ours) use some heuristics such as not splitting if coordinate structure is preceded by "*between*". In all other cases, coordinate structure is treated as segregatory, and is split.

The human-annotated gold labels of CaRB dataset correctly handle conjunctive sentences in most of the cases. However, we find that compared to scoring function of OIE2016 and Wire57,

[12] https://github.com/gabrielStanovsky/oie-benchmark

[13] https://github.com/rali-udem/WiRe57

[14] We refer to *subject*, *relation* and *object* as *arguments* of the extraction.

[15] https://github.com/dair-iitd/CaRB

	System 1 (P, R, F1)	System 2 (P, R, F1)
Talks resumed between USA and China Gold: (Talks; resumed; between USA and China)	(Talks; resumed; between USA) (Talks; resumed; between China) CaRB: (50.0, 66.7, 57.1) CaRB (1-1): (50.0, 66.7, 57.1)	(Talks; resumed; between USA and China) CaRB: (100, 100, 100) CaRB (1-1): (100, 100, 100)
I ate an apple and orange Gold: (I; ate; an apple) (I; ate; an orange)	(I; ate; an apple) (I; ate; an orange) CaRB: (100, 100, 100) CaRB (1-1): (100, 100, 100)	(I; ate; an apple and an orange) CaRB: (57.1, 100, **72.7**) CaRB (1-1): (53.5, 50.0, **57.1**)

Table 7: Evaluation of CaRB and CaRB (1-1) on two sentences.

CaRB over-penalizes systems for incorrectly splitting combinatory coordinate structures.

We trace this issue to the difference in mapping used for recall computation (one-to-one vs many-to-one).

Consider two systems – System 1, which splits on all conjunctive sentences (without any heuristics), and System 2, which does not. For the sentence "*I ate an apple and orange*", the set of gold extractions are {*(I; ate; an apple)*, *(I; ate; orange)*}. System 2, which (incorrectly) doe not split on the coordinate structure, gets a perfect recall score of 1.0, similar to System 1, which correctly splits the extractions (Table 7). On the other hand, when System 2 incorrectly splits extractions for the sentence "*Talks resumed between USA and China*", it is penalized on both precision and recall by CaRB, giving it a much lower score than System 2.

Due to this phenomena, we find that the gains obtained by our system on splitting the segregatory coordinate structures correctly is overshadowed by penalties of incorrectly splitting the coordinate structures. To re-affirm this, we evaluate all the systems on **CaRB(1-1)**, a variant of CaRB which retains all the properties of CaRB, except that it uses one-to-one mapping for computing recall.

We notice that our CIGL-OIE+IGL-CA shows improvements in CaRB(1-1) and other metrics which use one-to-one mapping (OIE16, Wire57) (Table 2). But it shows a decrease in CaRB score. This demonstrates that the primary reason for the decrease in performance is the many-to-one mapping in CaRB.

However, we also observe that this is not the best strategy for evaluation as it assigns equal score to both the cases — splitting a combinatory coordinate structure, and not splitting a segregatory coordinate structure (Table 7). This is also not desirable as a long extraction which is not split is better than two incorrectly split extractions. Hence, we consider that one-to-one mapping for computing recall under-penalizes splitting a combinatory coordinate structure.

Determining the right penalty in this case is an open-ended problem. We leave it to further research to design an optimal metric for evaluating conjunctive sentences for OpenIE.

B Reproducibility

Compute Infrastructure: We train all of our models using a Tesla V100 GPU (32 GB).

Hyper-parameter search: The final hyper-parameters used during train our model are listed in Table 8. We also list the search-space, which was manually tuned. We select the model based on the best CaRB (F1) score on validation set.

Validation Scores: We report the best validation scores in Table 9.

Number of parameters: The CIGL-OIE model contains 110 million parameters and IGL-CA contains 335 million parameters. The difference is because they use BERT-base and BERT-large models, respectively.

C Manual Comparison

The set of extractions from both the systems, CIGL-OIE and OpenIE6 were considered for a random 100 conjunctive sentences from the validation set. We identify a conjunctive sentence, based on the predicted conjuncts of coordination analyzer. The annotators are instructed to check if the extraction has well formed arguments and is implied by the sentence.

A screenshot of the process is shown in Figure 5.

Hyperparameters	Best Values	Grid Search
Training:		
Batch Size	24	{16,32,24}
Optimizer	AdamW	{AdamW, Adam}
Learning Rate	2×10^{-5}	$\{1 \times 10^{-3}, 2 \times 10^{-4}, 5 \times 10^{-5}\}$
Model:		
Iterative Layers	2	{1,2,3}
λ_{posc}	3	{0.1, 1, 3, 5, 10}
λ_{hvc}	3	{0.1, 1, 3, 5, 10}
λ_{hve}	3	{0.1, 1, 3, 5, 10}
λ_{ec}	3	{0.1, 1, 3, 5, 10}

Table 8: Hyperparameter settings.

System	CaRB		CaRB(1-1)		OIE16-C		Wire57-C
	F1	AUC	F1	AUC	F1	AUC	F1
IMoJIE	55.2	35.2	43.1	23.4	59.0	42.5	38.7
IGL-OIE	53.4	32.7	41.8	22.0	56.8	36.6	36.9
CIGL-OIE	**55.2**	**35.5**	43.9	23.9	62.3	42.4	39.1
CIGL-OIE + IGL-CA (OpenIE6)	53.8	35.0	**47.5**	**27.7**	**67.7**	**51.9**	**42.4**

Table 9: Evaluation of OpenIE systems on validation set

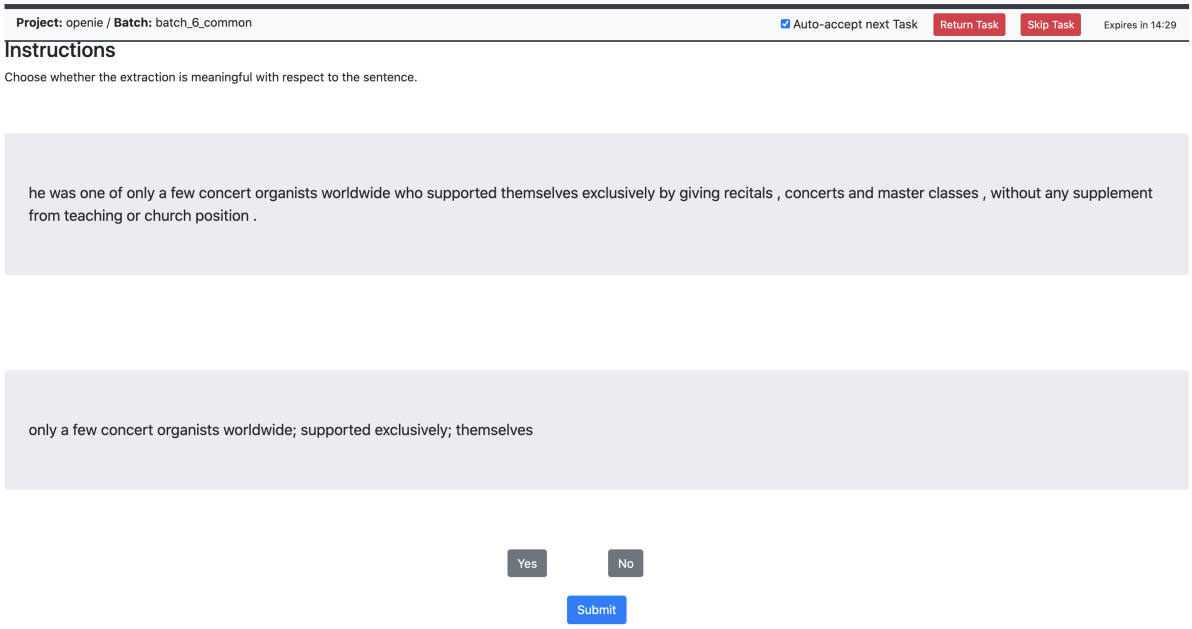

Figure 5: Process for manual comparison. Each extraction from both the systems are presented to the annotator in a randomized order. The annotator checks if the extraction can be inferred from the original sentence and marks it accordingly.