Open Information Extraction with Entity Focused Constraints

Prajna Upadhyay
BITS Pilani Hyderabad Campus,
Secunderabad, India
prajna.u@hyderabad.bits-pilani.ac.in

Oana Balalau and **Ioana Manolescu**
Inria and Institut Polytechnique de Paris
Palaiseau, France
first.last@inria.fr

Abstract

Open Information Extraction (OIE) is the task of extracting tuples of the form (subject, predicate, object), without any knowledge of the type and lexical form of the predicate, the subject, or the object. In this work, we focus on improving OIE quality by exploiting domain knowledge about the subject and object. More precisely, knowing that the subjects and objects in sentences are often named entities, we explore how to inject constraints in the extraction through constrained inference and constraint-aware training. Our work leverages the state-of-the-art OpenIE6 platform, which we adapt to our setting. Through a carefully constructed training dataset and constrained training, we obtain a 29.17% F1-score improvement in the CaRB metric and a 24.37% F1-score improvement in the WIRe57 metric. Our technique has important applications – one of them is investigative journalism, where automatically extracting conflict-of-interest between scientists and funding organizations helps understand the type of relations companies engage with the scientists. Our code and data are available at `https://github.com/prajnaupadhyay/openie-with-entities`

1 Introduction

Open Information Extraction (OIE) is the task of extracting triples from unstructured corpora in a domain-independent manner. A triple consists of a subject, a relation, and an object. OIE has important applications, such as question answering (Lu et al., 2019), or automatically creating or extending knowledge bases (Bhutani et al., 2019). OIE is a challenging task, with the performance of state-of-the-art models varying from 88.5% F1 score (Wang et al., 2021) to 34% (Gashteovski et al., 2021), depending on the difficulty of the benchmark.

When the named entities in a domain are known to be the subject/object of extractions, OIE should also identify relations between these entities. An important use case is automatically creating a knowledge base of relations between scientists and companies, i.e. identifying conflict-of-interest between the scientists and funding bodies, where the named entities are the names of scientists and companies, and the relation describes the conflict of interest between them. Clustering these relation phrases, such as `received a research gift from`, `received speaker fees` or `consults for` helps analyze the relationships that companies engage with the scientists. These relations are crucial to understanding scientists' positions on health issues (Oreskes and Conway, 2010) in investigative journalism. However state-of-the-art OIE models do not always retain named entities in the extractions, for example, given the sentence *"Shahrad Taheri received funding for research through a grant from Cambridge Weight Plan"*, an OIE tool (Kolluru et al., 2020a) returns ⟨`Shahrad Taheri, received, funding for research`⟩. While this extraction correctly identifies the subject of the triple, the quality of the predicate and object could be improved as follows: the extraction ⟨`Shahrad Taheri, received funding for research through a grant from, Cambridge Weight Plan`⟩ retains the second important entity (`Cambridge Weight Plan`) and is precise about the relation. Such sentences are frequent in the declarations of conflict of interest that authors add to articles in PubMed, a dataset of scientific articles on life sciences and biomedical topics.

In this work, we focus on **relation extraction, when the subject and object are named entities**. In particular, we would like to significantly improve the performance of OIE tools, such that triples as ⟨`first entity, predicate, second`

The work was done when the first author was at Inria and Institut Polytechnique de Paris.

entity⟩ are not missed or poorly extracted. To achieve this, we leverage deep learning with constraints, i.e. techniques that enforce constraints on the classifier's predictions. Constrained learning is very common in sequence-to-sequence tasks, such as relation or entity extraction, where the output should have a specific form. Constraint learning has also been successfully used in OIE. In our case, we enforce constraints on the subject, object and predicate forms, and we investigate several techniques to achieve the best result, such as constraint-aware training (Nandwani et al., 2019) and constraint inference (Lee et al., 2019). We deployed our technique within OpenIE6 (Kolluru et al., 2020a), a state-of-the-art tool for OIE.

Our salient contributions are: *i*) We extend the OpenIE6 model with entity-centric constraints; *ii*) We implement the constraints as penalties in the loss function, and as hard constraints during inference. *iii*) We show through an extensive evaluation that our method improves over the state-of-the-art; *iv*) We perform a large scale evaluation of the system, on conflict of interest declarations from PubMed bibliographical data.

2 Related Work

In the literature, the extraction of triples of the form ⟨subject, relation, object⟩ has been studied in several settings. A relation can be expressed using a surface form, i.e., the tokens present in a sentence, or a canonical form, usually introduced in a knowledge base. In the most general setting, we do not enforce any constraints on the types of the three elements, and the task is referred to as **open information extraction** (OIE). In the most restricted setting, the subject and object are entities, and the relation comes from a predefined set of relations. This task is known as **relation extraction**. Finally, **open relation extraction**, also referred to as **relation discovery**, refers to approaches that use little training (such as distant supervision, few-shot learning, or semi-supervision) or no training (unsupervised) to classify relations between entities. Some inconsistencies arise in the use of the terminology in the literature, e.g., "open relation extraction" has been also used to designate open information extraction, in (Mesquita et al., 2013).

Open Information Extraction. Open information extraction (Kolluru et al., 2020a; Etzioni et al., 2008) extracts triples from unstructured corpora in a domain-independent way. More precisely, the relations are not known beforehand and the subject and object are not required to be named entities. The state-of-the-art techniques are based on neural networks, which model the problem as a sequence labeling task (Kolluru et al., 2020a; Stanovsky et al., 2018; Cui et al., 2018). OpenIE6 (Kolluru et al., 2020a) is a neural model that achieves state-of-the-art results when compared with several other models (Del Corro and Gemulla, 2013; Gashteovski et al., 2017; Cui et al., 2018; Stanovsky et al., 2018; Roy et al., 2019; Zhan and Zhao, 2020; Kolluru et al., 2020b). Since these tools work without any domain knowledge, they might miss or extract poorly triples containing named entities. We aim to solve this problem, and our technique is trained to improve relation extraction when entities are present in the corpus.

Relation Extraction. In relation extraction (Han et al., 2020), given a sentence containing two entities, the task is to select the relation between the entities from a fixed set of relations. This is achieved via a classifier, and the challenge is in identifying relevant features for classification. Traditionally this has been achieved via hand-crafted features, such as lexical, syntactic, or semantic (Jiang and Zhai, 2007; Nguyen et al., 2007). More recently, neural models such as BERT (Devlin et al., 2018) have been very successful in relation classification (Baldini Soares et al., 2019).

Open Relation Extraction/Relation Discovery. In (Yao et al., 2011), the authors first discover relations between entities using the dependency paths between two tagged entities, and they propose an unsupervised probabilistic generative model for inducing clusters from the surface forms. In (Yu et al., 2017), surface forms of relations are first extracted by taking into account the dependency path between entities, and finally, they are mapped to canonical forms present in a KB. In (Hu et al., 2020), the authors propose a relation encoder based on BERT (Devlin et al., 2018) that computes an embedding representation of the relation based on the sentence where named entities appear, together with an adaptive clustering technique that does not require prior knowledge of the number of clusters. While some approaches(Yao et al., 2011; Yu et al., 2017) extract surface forms of relations when the arguments are entities, similar to our goal in this work, they use for this only dependency path information and do not deal with conjunctive sentences as OpenIE (Kolluru et al., 2020a). In addition, OpenIE6 has shown better performance than models using dependency parsing such as ClausIE (Kolluru et al., 2020a; Del Corro and Gemulla, 2013).

3 Problem Definition

Our goal is to extract triples from sentences that respect the guidelines detailed by the CaRB metric (Bhardwaj et al., 2019), i.e., they should be i) complete: all triples should be extracted from a sentence, ii) asserted: the triple should be implied from the sentence iii) informative: the triple should contain maximum relevant information from the sentence and iv) atomic: extraction cannot be split into multiple extractions.

Given a sentence S containing entities $E = \{e_1, ..., e_i, ..., e_n\}$, we denote by $\langle$`S, R, O`$\rangle$ a triple that is extracted from S. The CaRB rules can be customized to fit our setting as follows:

- **Complete**: For every e_i, there exists at least a triple $\langle$`S, R, O`$\rangle$ where e_i is `S` or `O`.
- **Asserted**: Each tuple must be implied by the original sentence.
- **Informative**: The extraction should contain the maximum possible information from S. For instance, from *Joe Biden is the president of the US*, an uninformative extraction is $\langle$`Joe Biden, is, the president`$\rangle$ while the informative extraction is $\langle$`Joe Biden, is the president of, US`$\rangle$.
- **Atomic**: If `S` or `O` contains e_i, then it contains only that entity and no additional tokens. If `S` or `O` contain e_i and e_j, it is always possible to create two triples $\langle$`S1, R, O`$\rangle$ and $\langle$`S2, R, O`$\rangle$, `S1` $= e_i$ and `S2` $= e_j$, similarly for `O`.

4 Entity Focused Constraints

OpenIE6 (Kolluru et al., 2020a) receives in input a sentence and outputs a list of extractions of the form $\langle$`subject, predicate, object`$\rangle$. The architecture of the model is a deep neural network that first encodes tokens using BERT (Devlin et al., 2018), and then iteratively identifies at most M extractions, i.e., calls the same architecture for each extraction for M times (Figure 1). The embeddings of the labels generated at the end of the 1st iteration are added to the embeddings of the tokens in the second iteration, and so on. This adds context so that a new extraction is generated the next time. Each token is assigned a label from $\{S$ (subject), R (relationship), O (object) or N (none)$\}$.

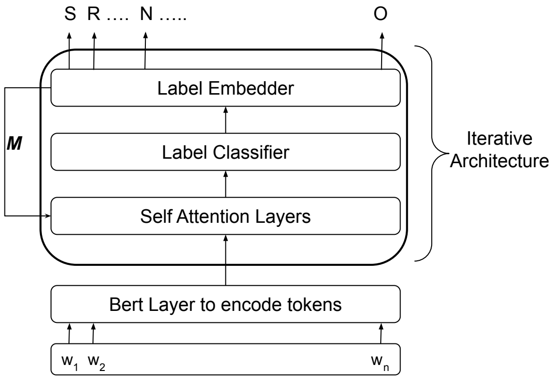

Figure 1: OpenIE6 uses the same architecture to generate embeddings for the words in M extractions, with the output of the previous extraction given as input for the next extraction

OpenIE6 constraint-aware training OpenIE6 uses constraint-aware training to infuse the model with task-related knowledge in the form of constraints. The model learns to satisfy these constraints during training without explicitly enforcing them during the inference, hence these types of constraints are typically referred to in the literature as *soft constraints*. This is achieved by adding additional penalties in the loss function, as follows:

POS Coverage (POSC). Tokens labeled as nouns, verbs, adjectives, or adverbs should be part of at least one extraction.

Head Verb Coverage (HVC). Verbs that are not *light verbs* (e.g., do, give, have, make, etc.), referred to as *head verbs*, should be present in the relation span of *a few but not too many extractions*.

Head Verb Exclusivity (HVE). The relation span of one extraction should contain *at most one head verb*.

Extraction Count (EC). The extractions having head verbs in the relation should be at least equal to the number of head verbs in the sentence.

These are **entity independent constraints**. Their full equations can be found in the OpenIE6 paper (Kolluru et al., 2020a).

Adding entity-specific constraints. We enforce additional constraints to obtain extractions satisfying our problem statement. Let $x_n^{ent} \in \{0, 1\}$ denote whether the nth token w_n belongs to some entity tagged in the sentence, and E be the set of entities. At each extraction level m, the model computes $Y_{mn}(k)$, the probability of assigning to the nth token the label $k \in \{S, R, O, N\}$ (subject, relation, object or none). We introduce the following entity-specific constraints:

1. **Entities as subject or object (ENT-ARG).** Each entity in the sentence should be present in at least a subject or object of an extraction:

$$J_{ent\_so} = \sum_{n=1}^{N} x_n^{ent} \cdot \big(1 - \max_{m \in [1,M]} \big(\max_{k \in \{S,O\}} Y_{mn}(k)\big)\big) \tag{1}$$

   The penalty is 0 when for each token belonging to an entity ($x_n^{ent} = 1$) we have $Y_{mn}(k) = 1$, that is maximum probability of being in the subject or object, for at least one extraction.

2. **Entity exclusivity (ENT-EXCL).** The subject and object should contain at most one entity each. Let $p_e(k)$, with $k \in \{S, R, O, N\}$ be the average token probability of label k in entity e, where e consists of one or more tokens. Then, we express the penalty as follows:

$$J_{ent\_exs} = \sum_{m=1}^{M} \max\big(0, \big(\sum_{e \in E} p_e(S) - 1\big)\big) \tag{2}$$

$$J_{ent\_exo} = \sum_{m=1}^{M} \max\big(0, \big(\sum_{e \in E} p_e(O) - 1\big)\big) \tag{3}$$

   The penalty is 0 when no entity is labeled as subject/object or when only one entity is labeled as such ($\sum_{e \in E} p_e(O/S)$ is 0 or 1).

3. **Entity in relation penalty (ENT-REL).** A penalty is introduced if an entity appears as a part of a relation of some extraction. This loss is directly proportional to the probability of tokens that are part of some entities and which have been labeled as part of a relation:

$$J_{ent\_rel} = \sum_{n=1}^{N} x_n^{ent} \cdot \sum_{m=1}^{M} Y_{mn}(R) \tag{4}$$

   The penalty is 0 when $Y_{mn}(R)$ is 0 for every token of an entity.

4. **Entity segmentation penalty (ENT-TOG).** A penalty is introduced if tokens describing the same entity are not labeled in the same way, for example, the first token of the entity is part of the predicate, while the rest of the tokens are part of the object. Let $w(e)$ be the set of tokens in a given entity e. Let $l_p^m(w)$ be the predicted label of a token (the label with the highest probability) at extraction m. As we are concerned with entities described by two or more tokens, the predicted label l_e^m of the entity e is the majority label of its tokens, or the label with the highest total sum of probabilities in case of a tie. For each $w \in w(e)$, we introduce a loss equivalent to $Y_{mw(l_p)}$ if $l_p^m(w) \neq l_e^m$:

$$J_{ent\_seg} = \sum_{m=1}^{M} \sum_{e \in E} \sum_{w \in w(e)} Y_{mw(l_p)}(1 - \delta_{l_p^m(w), l_e^m}) \tag{5}$$

   where δ is the Kronecker delta function.

Finally, the total loss can be written as:

$$J_{ent} = J + \lambda_1 J_{ent\_so} + \lambda_2(J_{ent\_exs} + J_{ent\_exo}) + \lambda_3 J_{ent\_rel} + \lambda_4 J_{ent\_seg} \tag{6}$$

where λ_* are hyperparameters, while J is the original OpenIE6 loss.

Constraints at inference. We investigate a second type of constrained learning called constraint inference. The constraints applied in this setting are hard constraints, which the model is forced to apply. The constraints are applied in the decoding phase and modify the tokens' labels (S, P, O, N). We propose three constraints inspired by the entity constraints introduced in the constraint-aware training.

1. **Entity exclusivity.** Once we have encountered one entity labeled as a subject or object in the sentence, the following entities are not allowed to receive the same label.

2. **Entity in relation.** We enforce that an entity appearing in the predicate is classified according to its second-best class probability.

3. **Entity segmentation penalty.** We enforce that all the tokens belonging to an entity be labeled with the same label.

We do not transform the constraint *entities as subject or object* in an inference constraint as it cannot be applied at the level of one existing extraction. This constraint can only be a penalty in the loss, such that it rewards sets of extractions in which all the entities are part of the arguments.

5 Experimental Evaluation

5.1 Datasets

We use the OpenIE6 data for training and validation and Pubmed data for testing. The OpenIE6 dataset consists of Wikipedia sentences, while the Pubmed data is a set of conflict of interest statements between authors and various organizations, such as those illustrated in Section 1.

Given that our focus is on improving performance when entities are present in a sentence (Section 3), and in particular, enforcing that entities are the subject or object, we need appropriate training data for the task. We are unaware of a dataset of extractions where arguments are entities, while the extraction also has the surface forms of relation. For example, FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017), two standard datasets used in relation extraction, do not contain the surface form of the relation; they only label the entire sentence as containing a particular relation.

Training data. The OpenIE6 training dataset consists of $91K$ sentences and $190K$ extractions of the form ⟨`subject, predicate, object`⟩. We tag entities in each sentence using the state-of-the-art named entity recognition tool Flair (Akbik et al., 2019). We focus on extractions of the following form: *i*) The **subject** of the extraction is exactly one entity; and *ii*) The **object** ends with an entity . We discard the extractions that do not match these constraints. In each extraction, we keep only the entity in the object and move the preceding tokens to the relationship part of the extraction. For example, one of the sentences in the original training set is *"Parmenides had a large influence on Plato, who not only named a dialogue, Parmenides, after Parmenides, but always spoke of Parmenides with veneration."* and one of the extractions is ⟨`Parmenides, had, a large influence on Plato`⟩. The extraction satisfies both the above conditions, hence we transform it to ⟨`Parmenides, had a large influence on, Plato`⟩. If the object contains only an entity, we apply the identify transformation. We refer to a sentence with at least one transformed extraction as a *clean sentence*.

We create 3 training datasets:

ORIGINAL: The original training set containing $91K$ sentences.

CLEAN: $7K$ clean sentences with their modified extractions.

MIXED: We add the remaining sentences and their extractions from the original training set to CLEAN.

Gold data. We created a gold standard dataset from Pubmed conflict-of-interest statements to be used as test data. We tagged and counted the entities with NER Flair and selected 282 sentences with a minimum of 2 entities. The maximum number of entities found in a sentence was 14.

We asked the annotators to find all the triples ⟨`S, P, O`⟩ containing those entities as arguments (in `S` or `O`). In addition, the extractions should follow the guidelines explained in Section 3 on completeness, assertion, informativeness, and atomicity. The total number of extractions obtained after annotations were 1113. One annotator annotated each sentence.

Quality of gold data. To evaluate the dataset's quality, we sampled 50 sentences from our gold sentences, and one of the authors annotated them so that we had two annotations for this set. We found the agreement by considering one annotation as gold and computing WiRE57 F1. The agreement F1 score obtained was 83, which is a high agreement.

Table 1 shows example annotations of triples for the sentence *Menno Huisman reports grants from and personal fees from Boehringer Ingelheim and Bayer Health Care.* For each CaRB property, we show the correct and incorrect extractions. An extraction of the form ⟨`Menno Huisman, reports grants from, Bayer Health Care, Germany`⟩ violates the assertion property because it adds extra information to the sentence. ⟨`Menno Huisman, reports, grants`⟩ violates the informativeness property even if it is a valid extraction because it lacks the complete second argument, i.e., `Boehringer Ingelheim`. The extraction ⟨`Menno Huisman, reports grants from, Boehringer Ingelheim and Bayer Health Care`⟩ is not atomic because the two entities in the second argument should be part of 2 extractions. If any of the four correct extractions adhering to the completeness property are missing, this property is violated.

5.2 Models

We experimented with the following models:

`OpenIE6`. This is the default OpenIE6 model.

`OpenIE6(ECTR)`. OpenIE6 model with entity constraint training (ECTR), as in Section 4.

	Correct	Incorrect
Completeness	⟨`Menno Huisman, reports grants from, Boehringer Ingelheim`⟩, ⟨`Menno Huisman, reports grants from, Bayer Health Care`⟩, ⟨`Menno Huisman, reports personal fees from, Boehringer Ingelheim`⟩, ⟨`Menno Huisman, reports personal fees from, Bayer Health Care`⟩	If any of the extractions is missing
Assertion	⟨`Menno Huisman, reports grants from, Bayer Health Care`⟩	⟨`Menno Huisman, reports grants from, Bayer Health Care, Germany`⟩
Informativeness	⟨`Menno Huisman, reports grants from, Boehringer Ingelheim`⟩	⟨`Menno Huisman, reports, grants`⟩
Atomic	⟨`Menno Huisman, reports grants from, Boehringer Ingelheim`⟩	⟨`Menno Huisman, reports grants from, Boehringer Ingelheim and Bayer Health Care`⟩, ⟨`Menno Huisman, reports grants from and personal fees from, Boehringer Ingelheim`⟩

Table 1: Examples of correct and incorrect annotations for the 4 CaRB properties

`OpenIE6(ECTR, ECIN)`. To the trained model `OpenIE6(ECTR)`, we add constraints at inference in the evaluation of the test data.

`OpenIE6(ECIN)`. To the trained model `OpenIE6`, we add constraints at inference in the evaluation of the test data.

We note that the models use a different coordinate boundary model than the one in the OpenIE6 paper. We retrained the coordinate boundary model using a newer Huggingface Transformers library version (Wolf et al., 2020) for compatibility with our code. However, we could not reproduce the accuracy, obtaining 83.3 instead of 85.4. A better coordinate boundary model would positively impact performance, both with and without constraints.

Parameters. The model's training consists of two phases, a warm-up phase, where the training is done without constraints, and a constrained training part. The warm-up training was done for 30 epochs, and the constrained training was done for 15 epochs. During constrained training, all constraints had equal weights. The learning rate was set to $5e{-}06$. `BERT-base-cased` model was used with two iterative layers. We repeat the experiments with 6 different random seeds for the network initialization, and we average the results. We run our code on a 32GB GPU.

Baselines We implement four baselines.

`ConnectingPhrase`. This simple technique returns the phrase connecting the two entities in a sentence as the relation between them. It comprises the following steps:

1. We first use the coordinate boundary detection model (available with OpenIE6 code). Coordinate boundary detection models (Saha and Mausam, 2018; Kolluru et al., 2020a) split a conjunctive sentence into smaller parts. For example, the sentence *"Adrian Brown and Shahrad Taheri received funding for research through a grant from Cambridge Weight Plan."* is split into:

   (a) *"Adrian Brown received funding for research through a grant from Cambridge Weight Plan."*

   (b) *"Shahrad Taheri received funding for research through a grant from Cambridge Weight Plan."*

   This is crucial to improve the recall.

2. Next, we label the entities in sentences obtained using Flair (Akbik et al., 2019).

3. For each consecutive pair of entities e_i, e_{i+1} in the sentence, we return an extraction containing as subject e_i, as predicate the phrase connecting the entities, and as object e_{i+1}.

4. We filter the extractions by removing the ones whose predicates do not contain a token labeled as a verb by a part-of-speech parser. The final set of extractions is obtained at the end of this step.

`DependencyPath`. We follow the same steps as in `ConnectingPhrase`, except that in 3. above, we return as the predicate the tokens on the dependency path between entities e_i and e_{i+1}.

`PostprocessedOpenIE6`. We run the original OpenIE6 tool and post-process its output as follows: we tag entities in `subject` and `object` of the extractions, and then we modify extractions, in the same manner as when we created the CLEAN dataset (Section 5.1), and leave unchanged the ones not satisfying our conditions.

`FilteredOpenIE6`. We remove the extractions from `PostprocessedOpenIE6` that were not modified according to the procedure used for generating the CLEAN dataset.

Evaluation metrics. Several evaluation metrics have been proposed to evaluate the performance of an OpenIE system. **WiRe57** (Lechelle et al., 2019) is a one-to-one matching metric, in which each system extraction is matched to exactly one gold extraction. Given a sentence, a system extraction matches a gold extraction if they share at least one word from each of the relation, subject, and object. Two extractions are compared by computing the token level recall and precision between the gold subject and system subject, respectively, the predicates and objects. Precision is the percentage of system words found in the gold extraction. The recall is the percentage of gold words in the systems' predictions. The system extractions are matched one-to-one to gold extraction in decreasing order of $F1$-score. **CaRB** (Bhardwaj et al., 2019) is a many-to-one matching metric in which several gold extractions can be matched to one system extraction when computing the recall. This avoids penalizing a system if one extraction would better correspond to two or more golden extractions, as is the case, for instance, in ⟨`Adrian Brown; has received travel grants from; Cambridge Weight Plan and Oxford University`⟩ (note that there should have been two triples extracted here, each with a different object). Precision is computed by matching system extractions one-to-one to gold extractions, decreasing order of precision score. Hence, we will penalize the extraction above when computing precision, as one gold extraction will not be matched.

We report both metrics, however, WiRe57 is more in line with our task as it respects the atomicity constraint in Section 3, given that it does not reward system triples with several entities in one argument.

6 Results and Discussion

Evaluation. In Table 2 we show the results on the test data, measuring both CaRB and WiRe57. We use the different training datasets that we introduced and the different training constraints. When training without any entity constraints, the training dataset can make a significant difference, as we observe `OpenIE6` trained on CLEAN has a more than 26% increase in CaRB F1 than `OpenIE6` trained on the ORIGINAL dataset. In addition, adding entity constraints further improves the results as shown by the models `OpenIE6(ECTR)` which has the best WiRe57 score for all CLEAN, MIXED and ORIGINAL models. The smallest improvement is for the model trained with the ORIGINAL dataset, as in this case the training data may be in conflict with the constraints, having for example several entities in one argument. `OpenIE6(ECIN)` improves upon `OpenIE6`, with a significant increase in the precision of the WiRe57 metric, which is expected given the hard constraints are being forced on the triples. However,`OpenIE6(ECTR)` has a more significant improvement than `OpenIE6(ECIN)` according to WiRe57 (the metric aligned with our problem statement, as explained in Section 5.2), showing that it is more important to have soft constraints, which are rewarding good extractions during training and hence obtaining a better extraction model. Combining soft and hard constraints gives the best model, `OpenIE6(ECTR, ECIN)`. Regarding the baselines, `PostprocessedOpenIE6` and `FilteredOpenIE6` have good precision but lower recall than our top-performing models, showing the importance of the constraint learning and adapted training datasets.

Ablation study. We perform an ablation study to evaluate the importance of the entity constraints added during the training. We take our best performing model, `OpenIE6(ECTR)` trained on the CLEAN dataset, and we train it with 1, 2, or 3 constraints at a time. Table 3 shows the results obtained on our test set. When we add just one constraint, as expected, the constraint ENT-ARG enforces the highest WiRe57 recall, as it has learned to penalize extractions where entities may be missing from the arguments. However, this model has the lowest precision, due to the fact it allows more than one

Method	Training data	CaRB P	CaRB R	CaRB F1	WiRe57 P	WiRe57 R	WiRe57 F1
OpenIE6	CLEAN	75.07	62.52	67.97	66.15	**45.79**	54.04
OpenIE6(ECTR)	CLEAN	<u>80.76</u>	61.77	<u>69.95</u>	73.05	<u>45.37</u>	<u>55.95</u>
OpenIE6(ECIN)	CLEAN	76.05	<u>63.64</u>	69.65	70.91	44.80	54.89
OpenIE6(ECTR, ECIN)	CLEAN	79.85	63.09	**70.46**	<u>74.88</u>	45.36	**56.48**
OpenIE6	MIXED	52.23	50.69	51.60	43.30	37.41	40.04
OpenIE6(ECTR)	MIXED	57.83	55.22	56.38	49.60	38.27	43.20
OpenIE6	ORIGINAL	43.01	39.75	41.29	32.45	31.81	32.11
OpenIE6(ECTR)	ORIGINAL	41.61	40.90	41.19	33.29	31.78	32.48
PostprocessedOpenIE6	ORIGINAL	59.25	52.52	55.62	43.82	42.49	43.12
FilteredOpenIE6	ORIGINAL	**85.01**	41.44	55.78	**81.01**	30.77	44.57
DependencyPath	-	58.51	57.54	58.02	59.65	36.94	45.62
ConnectingPhrase	-	58.23	**70.63**	63.84	58.45	45.31	51.04

Table 2: Model comparison on the test dataset. Best values are in bold and second best are underlined.

Constraints	CaRB P	CaRB R	CaRB F1	WiRe57 P	WiRe57 R	WiRe57 F1	Violations ENT-ARG	Violations ENT-EXCL	Violations ENT-REL	Violations ENT-TOG
∅ (OpenIE6)	75.07	62.52	67.97	66.15	45.79	54.04	32.44	1.59	16.30	1.11
{ENT-ARG}	64.58	<u>64.02</u>	63.70	58.21	**47.52**	52.24	**25.01**	4.90	**4.87**	**0.56**
{ENT-EXCL}	78.65	60.99	68.67	68.98	44.83	54.32	33.59	**1.30**	15.65	1.35
{ENT-REL}	78.18	62.24	69.10	70.50	<u>46.27</u>	55.73	28.72	2.68	8.77	1.18
{ENT-TOG}	75.39	62.77	68.28	67.15	45.51	54.20	32.51	1.48	15.57	1.13
{ENT-ARG, ENT-EXCL}	78.06	61.18	68.28	67.32	45.48	54.22	32.09	1.56	14.59	1.12
{ENT-ARG, ENT-REL}	74.23	59.20	65.79	62.86	46.13	53.04	<u>26.94</u>	4.60	<u>5.45</u>	0.89
{ENT-ARG, ENT-TOG}	67.85	**64.24**	65.54	62.17	46.17	52.82	27.77	3.98	7.98	<u>0.82</u>
{ENT-EXCL, ENT-REL}	80.22	61.97	69.89	72.93	45.17	55.77	32.27	<u>1.41</u>	10.35	1.36
{ENT-EXCL, ENT-TOG}	78.56	61.45	68.86	69.63	44.93	54.59	33.05	1.43	13.90	1.35
{ENT-REL, ENT-TOG}	74.38	61.53	66.93	66.60	43.21	52.31	32.33	3.34	9.29	1.74
EC \ ENT-ARG	79.12	62.70	69.90	**73.20**	45.03	55.75	32.15	1.49	9.64	1.32
EC \ ENT-EXCL	75.48	61.28	67.47	65.99	45.46	53.70	27.75	4.49	6.76	0.94
EC \ ENT-REL	**80.89**	60.48	69.19	68.75	45.74	54.89	31.72	1.58	12.31	1.13
EC \ ENT-TOG	79.98	62.20	<u>69.94</u>	72.76	45.43	<u>55.92</u>	31.26	1.59	9.62	1.22
EC (OpenIE6(ECTR))	<u>80.76</u>	61.77	**69.95**	<u>73.05</u>	45.37	<u>55.95</u>	31.39	1.63	9.77	1.30

Table 3: Ablation study with models trained on the CLEAN dataset. We report CaRB, WiRe57, and the percentage of entity constraints violations on the test set.

entity in one argument. Removing the constraint from the set, EC \ ENT-ARG, gives us the highest precision. A combination of ENT-EXCL and ENT-REL performs the best among the models that were trained with 2 constraints, which is expected since the models trained with ENT-EXCL and ENT-REL were the top-2 performing models when trained individually. Enforcing only ENT-TOG does not bring important improvements, and training with the whole EC is slightly better than when training with EC \ ENT-TOG. Hence, ENT-TOG could be removed without a significant drop in quality.

For a complete analysis, we also compute the percentage of violations in the extractions (Table 3). For ENT-ARG, we count as a violation every entity that is not found in at least one extraction, and we divide by the total number of entities in the test set. For ENT-EXCL, we count a violation for each subject or object with more than one entity and normalize by twice the number of extractions. For ENT-REL, a violation is a relation containing an entity, normalized by the number of extractions. Finally, for ENT-TOG, a violation is an entity in extraction with more than one tag (S,O,R,N), normalized by the number of extractions containing an entity. We observe that ENT-ARG is violated the most, followed by ENT-REL. When enforcing ENT-ARG, we obtain the best results for 3 out of 4 constraints. This does not result, however, in the best $F1$ score, showing the importance of minimizing violations of type ENT-EXCL. ENT-ARG and ENT-EXCL have competing goals: ENT-ARG enforces the occurrence of entities in arguments, but ENT-EXCL does not allow more than one entity in an argument. So, whenever ENT-ARG is enforced with ENT-EXCL, we see an increase in the number of ENT-ARG or ENT-EXCL violations. Finally, when comparing the model with no entity constraints, `OpenIE6`, with the model enforcing all 4 constraints, `OpenIE6(ECTR)`, we observe a more significant difference in the violations ENT-ARG and ENT-REL, the constraints that are more frequently violated.

Quality of Named Entity Recognition on Pubmed. We sampled and annotated 50 test set sentences, taking care to keep the words together in long named entities, such as "Oregon Health and Science University Center for Embryonic Cell and Gene Therapy". We obtained 88% F1 score for the NER model Flair (Akbik et al., 2019) , in line with the performance of the model on Ontonotes (Weischedel et al., 2017) and CONLL (Tjong Kim Sang and De Meulder, 2003).

Evaluation on the CaRB dataset. `OpenIE6` has been evaluated on the CaRB dataset (Bhardwaj et al., 2019). We evaluate our constrained models to investigate their performance on this standard benchmark, see Table 4. Note that annotating guidelines for CaRB were not the same as for our Pubmed test data: there might be more than one entity in the arguments of a relation. This inherently limits the quality of our results. However, we show that the constrained models trained on the MIXED and ORIGINAL datasets have competitive performance with the original `OpenIE6` model while performing much better on our test data, as shown in Table 2. As expected, the models trained on the CLEAN dataset perform the worst, as they have seen only extractions with entities in the arguments; to achieve the best results, the user should choose a model considering the nature of the dataset. We note that the results for the `OpenIE6` model are slightly lower than those reported in the original paper because of the coordinate boundary model, as mentioned in Section 5. The conjunctive model is a core OpenIE6 component; gains in its precision would likely improve performance, both with and without constraints.

Conflicts of interest in PubMed. We analyse the extractions by `OpenIE6(ECTR)` and the original `OpenIE6` model on a larger PubMed dataset consisting of $170K$ sentences. Table 5 shows the number of extractions (**#ext**), extractions containing one entity in the subject and object (**#ext1**), containing a "Person" entity in subject and "Organization" entity in the object (**#ext2**), and the number of sentences processed by the model per second (**speed**). `OpenIE6(ECTR)` finds more interesting triples where a conflict of interest relation is expressed between a person and an organization entity, compared to the original `OpenIE6`. Also, our model processes more sentences per second compared to the original `OpenIE6`. This is because `OpenIE6` generates more extractions per sentence, however, even with more extractions, the model retrieves fewer conflicts of interest relations between a person and an organization.

Method	Training data	CaRB F1	WiRe57 F1
OpenIE6(ECTR)	CLEAN	24.44	11.59
OpenIE6	CLEAN	27.76	12.42
OpenIE6(ECTR)	MIXED	50.16	38.15
OpenIE6	MIXED	50.27	38.59
OpenIE6(ECTR)	ORIGINAL	**50.70**	**39.28**
OpenIE6	ORIGINAL	<u>50.60</u>	<u>39.15</u>

Table 4: Model comparison on the CaRB dataset. Best values are in bold and second best are underlined.

	OpenIE6(ECTR)	OpenIE6
#sen	170298	
#ext	233081	**564877**
#ext1	**138188** (59.29%)	117795 (20.85%)
#ext2	**106232** (45.58%)	92152 (16.31%)
speed	**87.41**	56.14

Table 5: Comparison of extractions from a larger dataset of PubMed conflict of interest statements.

Conclusion. We presented an approach that significantly improves OIE when the input sentence contains entities while being competitive on a standard OIE benchmark. Finally, we showed that our method is much better suited for a real use case, as it extracts high-quality triples from PubMed.

Acknowledgments We thank Rémi Goujot for manually annotating the PubMed dataset used to test our model, during his high school internship. This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011011614R2). The authors were partially funded by the ANR-20-CHIA-0015 project and by the Hi!PARIS Center.

7 Limitations

We identify the following limitations affecting our proposed methods:

- The performance of our models is impacted by the quality of the named entity recognition tool, as well as the performance of the conjunctive model.
- Training OpenIE6 with more constraints requires around 3h/epoch, while the model with the original constraints requires half this time.
- Users trying our tool, but also the original OpenIE model, should have the computational possibility of using the BERT-based model, the main component of OpenIE6. We plan to release trained models based of smaller language models.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

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, 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.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

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, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. 2021. Benchie: Open information extraction evaluation based on facts, not tokens. *arXiv preprint arXiv:2109.06850*.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. SelfORE: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Rochester, New York. Association for Computational Linguistics.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020a. Openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3748–3761. Association for Computational Linguistics.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020b. 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.

William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. WiRe57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, Florence, Italy. Association for Computational Linguistics.

Jay Yoon Lee, Sanket Vaibhav Mehta, Michael Wick, Jean-Baptiste Tristan, and Jaime Carbonell. 2019. Gradient-based inference for networks with output constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4147–4154.

Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. 2019. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. SIGIR'19, page 105–114, New York, NY, USA. Association for Computing Machinery.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Seattle, Washington, USA. Association for Computational Linguistics.

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

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Relation extraction from wikipedia using subtree mining. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, page 1414–1420. AAAI Press.

N. Oreskes and E.M. Conway. 2010. *Merchants of Doubt: How a Handful of Scientists Obscured the Truth on Issues from Tobacco Smoke to Global Warming*. Bloomsbury Publishing.

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.

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

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238.

Ralph M. Weischedel, Eduard H. Hovy, Mitchell P. Marcus, and Martha Palmer. 2017. Ontonotes : A large training corpus for enhanced processing.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dian Yu, Lifu Huang, and Heng Ji. 2017. Open relation extraction and grounding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 854–864, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.