DetIE: Multilingual Open Information Extraction Inspired by Object Detection

Michael Vasilkovsky[1,10], **Anton Alekseev** [3,4], **Valentin Malykh** [2,3,6,9], **Ilya Shenbin** [3], **Elena Tutubalina** [5,6,7], **Dmitriy Salikhov** [7], **Mikhail Stepnov** [7], **Andrey Chertok** [7,8], **Sergey Nikolenko** [3,9,10]

[1] Skolkovo Institute of Science and Technology, Moscow, Russia
[2] Huawei Noah's Ark lab, Moscow, Russia
[3] St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, St. Petersburg, Russia
[4] St. Petersburg State University, St. Petersburg, Russia
[5] HSE University, Moscow, Russia
[6] Kazan Federal University, Kazan, Russia
[7] Sber AI, Moscow, Russia
[8] Artificial Intelligence Research Institute, Moscow, Russia
[9] ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia
[10] Neuromation OU, Tallinn, Estonia

Abstract

State of the art neural methods for open information extraction (OpenIE) usually extract triplets (or tuples) iteratively in an autoregressive or predicate-based manner in order not to produce duplicates. In this work, we propose a different approach to the problem that can be equally or more successful. Namely, we present a novel single-pass method for OpenIE inspired by object detection algorithms from computer vision. We use an order-agnostic loss based on bipartite matching that forces unique predictions and a Transformer-based encoder-only architecture for sequence labeling. The proposed approach is faster and shows superior or similar performance in comparison with state of the art models on standard benchmarks in terms of both quality metrics and inference time. Our model sets the new state of the art performance of 67.7% F1 on CaRB evaluated as OIE2016 while being 3.35x faster at inference than previous state of the art. We also evaluate the multilingual version of our model in the zero-shot setting for two languages and introduce a strategy for generating synthetic multilingual data to fine-tune the model for each specific language. In this setting, we show performance improvement of 15% on multilingual Re-OIE2016, reaching 75% F1 for both Portuguese and Spanish languages. Code and models are available at https://github.com/sberbank-ai/DetIE.

Introduction

Extracting structured information from raw texts is a key area of research in natural language processing (NLP). It has a core set of well-defined basic problems: relation extraction, named entity recognition (NER), slot filling, and so on, each defining a specific view on the perception and analysis of textual data. In this work, we follow the paradigm of open information extraction (OpenIE) that represents texts from an arbitrary domain as a set of (subject, relation, object) triplets (Yates et al. 2007). OpenIE methods do not rely on a pre-defined ontology schema and are trained to be domain-agnostic, so they can be used in many downstream NLP tasks: multi-document question answering and summarization (Fan et al. 2019), event schema induction (Balasubramanian et al. 2013), fact salience (Ponza, Del Corro, and Weikum 2018), word embedding generation (Stanovsky, Dagan, and Mausam 2015), and more. Using triplets as graph edges, OpenIE systems serve as a core component for unsupervised knowledge graph construction (Mausam 2016); high-quality OpenIE systems can output an open knowledge graph even without further post-processing.

Historically, OpenIE systems were purely statistical or rule-based, often consisting of several components such as PoS (part-of-speech) tagging or syntax parsing, so errors tended to accumulate. Recently, end-to-end neural network systems for OpenIE have begun to outperform their non-neural counterparts. There exist two paradigms in neural OpenIE: *sequence labeling* (Stanovsky et al. 2018; Roy et al. 2019; Kolluru et al. 2020a) and *sequence generation* (Cui, Wei, and Zhou 2018; Kolluru et al. 2020b), each with their own merits and drawbacks. To avoid duplicates, in both paradigms triplets are usually extracted iteratively either in the autoregressive (Kolluru et al. 2020a,b) or predicate-based manner (Ro, Lee, and Kang 2020). During training, autoregressive methods predict triplets in a prefedined order that usually has no meaning and excessively penalizes the model. Predicate-based methods first extract all predicates and then iteratively find arguments for each, assuming that a predicate occurs in only one chain of arguments, which may not hold both in common benchmarks and in the real world.

In this work, we view OpenIE from a different perspective, as a direct set prediction problem. Our approach is inspired by one-stage anchor-based object detection models from computer vision (Liu et al. 2015; Tan, Pang, and Le 2020) that predict all bounding boxes in one forward pass and apply intersection-based matching to match predictions with the ground truth. We bridge the inter-discipline gap and bring this idea to OpenIE, demonstrating increased or equivalent performance compared to state of the art methods.

We train our model on two corpora: (1) training set of OpenIE6 (Kolluru et al. 2020a) and (2) recently released LSOIE (Solawetz and Larson 2021). For evaluation, following Kolluru et al. (2020a), we employ the CaRB test set (Bhardwaj, Aggarwal, and Mausam 2019) together with OIE2016 (Stanovsky and Dagan 2016), WiRe57 (Lechelle, Gotti, and Langlais 2019), CaRB and CaRB (1-1) evaluation scorers. Our main contributions are as follows: (1) we introduce DetIE, a novel approach for OpenIE that demonstrates improvements on common English benchmarks in both quality metrics and inference time; DetIE does not replace the entire OIE pipeline and can be combined with existing techniques such as grid constraints or coordination analysis, which, as we show, might further improve the results; (2) we investigate the language transferability of our model to other languages and obtain significant improvements on multilingual benchmarks; (3) we propose a strategy for generating multilingual synthetic data and finetuning the model for a specific language.

Related Work

Early open information extraction approaches such as TextRunner (Etzioni et al. 2008), ReVerb (Fader, Soderland, and Etzioni 2011), OLLIE (Schmitz et al. 2012), ClausIE (Del Corro and Gemulla 2013), and MinIE (Gashteovski, Gemulla, and Corro 2017) were mostly rule-based, used automatically generated training data and separated modules such as PoS taggers, dependency parsers, and chunkers. While they have advantages such as domain independence, errors from separate modules tend to accumulate.

Modern approaches are usually based on neural networks with either recurrent (Stanovsky et al. 2018; Cui, Wei, and Zhou 2018) or, more recently, Transformer-based architectures (Kolluru et al. 2020a,b). Neural models can be trained end to end but require labeled data, and manual relation annotation for supervised learning is extremely costly. Therefore, neural open IE partly relies on classical approaches.

Sequence labeling approaches assume a triplet to be a subset of the input sequence, either predicting spans for all three parts in the input sequence or assigning a corresponding label—subject, relation, object, or background—to every token and assembling a triplet from these labels. These models cannot change the sentence structure or introduce new auxiliary words while generating predictions.

The *RnnOIE* model (Stanovsky et al. 2018) predicts entities given ground truth predicates during learning, but predicates are extracted with a PoS tagger during inference. In order for the model to be able to extract multiple overlapping tuples for each sentence, the authors used an extended version of *BIO tagging* (beginning-inside-outside) (Ramshaw and Marcus 1999). Since then, several models have extended and improved over *RnnOIE*. In order to alleviate the lack of labeled training data, *SenseOIE* (Roy et al. 2019) augments model inputs with extractions from existing IE systems such as word embedding, part-of-speech tags, syntactic role labels, and dependency structure. *Multi^{2}OIE* (Ro, Lee, and Kang 2020) is a two-step procedure that first predicts the predicates and then the corresponding entities. This model is able to make multilingual predictions by using multilingual BERT embeddings even if it had been trained only on an English dataset. *SpanOIE* (Zhan and Zhao 2020) is also a two-step model, but unlike sequence labeling models shown above it predicts a span instead of a BIO tag for every token.

In natural language texts, predicates are often present only implicitly. To solve this, finding predicates can be viewed as a classification task (Zeng et al. 2014), but this approach is unsuitable for an open vocabulary setting. There, researchers use *sequence generation* approaches: encoder-decoder frameworks that produce triplets as sequences, typically split via special tokens. The first such model was *NeuralOIE* (Cui, Wei, and Zhou 2018), later improved in *IMoJIE* (Kolluru et al. 2020b). In IMoJIE, the next extraction is conditioned on all previously extracted tuples, which leads to more diverse tuples. Sequence generation models are heavy and have relatively low performance in both learning and inference due to autoregressive output generation. This problem was partially resolved in the *IGL-OIE* model (Kolluru et al. 2020a), where the next extraction is still conditioned on all previous extractions, but the tuples themselves are extracted in the sequence labeling fashion.

A key advantage of sequence labeling over sequence generation is that the problem is formalized as token classification, so all classification-related techniques can be applied. On the other hand, it is hard to define similarity metrics between generated text and the ground truth; existing metrics do not correlate well with human judgement (Mathur, Baldwin, and Cohn 2020; Lukasik et al. 2020), so the sequence generation approach is inherently biased.

Autoregressive generation of triplets inherent in previous methods forces the model to predict triplets in a predefined order, leading to additional arbitrary penalties. In this work, we alleviate this problem and propose an approach that is entirely novel compared to the works discussed above.

Method

We follow the sequence labeling paradigm. Given an input sequence $\{x_1 \dots x_T\}$, the goal is to predict a set S of token masks $\{\{L_{1,1} \dots L_{T,1}\} \dots \{L_{1,N} \dots L_{T,N}\}\}$, $|S| = N$, labeling each token in a mask with exactly one of $C = 4$ classes: “Background”, “Subject”, “Relation”, or “Object”. If a mask contains non-background tokens, it produces a triplet; for some applications, it is needed to ensure that subject, relation, and object tokens are all present in the mask.

Our method has two main components: a feedforward neural architecture and an order-agnostic loss function. It follows the general idea of convolutional architectures used for object detection in computer vision: it (1) makes mutually-aware predictions in a single pass and (2) matches them with the ground truth via intersection-based matching during training. The latter both encourages relevant predictions to be closer to the ground truth and serves to discard irrelevant predictions and duplicates. This type of architectures has been used in one-stage object detection, especially in the family of *single-shot detectors* (SSD) (Liu et al. 2015) and their later developments with feature pyramids, e.g., RetinaNet (Lin et al. 2017). In computer vision, SSD uses a

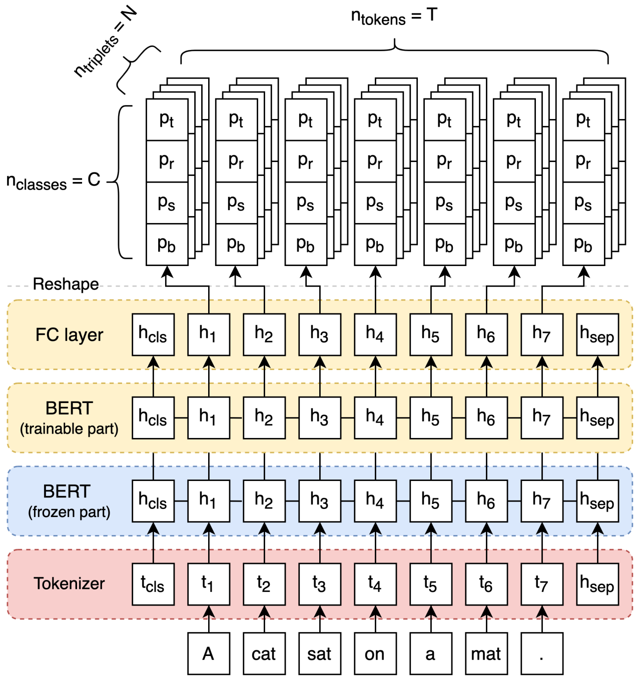
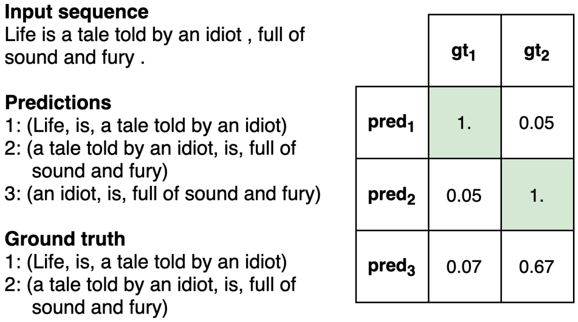


Figure 1: Model architecture.



Input sequence
Life is a tale told by an idiot , full of sound and fury .

Predictions
1: (Life, is, a tale told by an idiot)
2: (a tale told by an idiot, is, full of sound and fury)
3: (an idiot, is, full of sound and fury)

Ground truth
1: (Life, is, a tale told by an idiot)
2: (a tale told by an idiot, is, full of sound and fury)

	gt_1	gt_2
$pred_1$	1.	0.05
$pred_2$	0.05	1.
$pred_3$	0.07	0.67

Figure 2: Sample matching: predictions vs. ground truth; $pred_3$ is not matched and will be penalized.

set of predefined *anchor boxes* that represent default bounding box predictions for each position in the feature map; the same network predicts both class labels and refined positions for each anchor box, usually on several different scales, and the network is trained in an end-to-end fashion with a single loss function. In DetIE, the counterparts of anchor boxes are masks representing *possible triplets*.

Model

Our model aims to extract a large predefined number of probability masks from a single text fragment, each corresponding to a possible triplet. For a given sequence of tokens, the model produces a three-dimensional tensor of probabilities p (Fig. 1) of shape (T, N, C), where T corresponds to the number of tokens, N is a pre-defined number of possible extracted triplets, and C is the number of classes designated above. Since N serves as an upper bound on the number of predictions, it should be large enough to cover all possible triplets in a document. However, larger values of N exacerbate the class imbalance discussed in the next section. We found $N = 20$ to be sufficient for our datasets.

The output at position (t, n, c) is the probability of a token t to belong to class c in mask n. The actual prediction is obtained from the probabilities by taking the $\arg\max$ over classes for each token and filtering out background-only masks (i.e., if the maximal mask contains at least one non-background token, we extract it as a triplet).

The core of the architecture is a pre-trained BERT encoder as implemented in the *HuggingFace* library (Devlin et al. 2019; Wolf et al. 2020). We map its output to $N \times C$ channels for every input token with a fully-connected layer, and then reshape the obtained tensor to the shape of p from above. The value of N does not affect the performance significantly since it only scales the last layer in our model linearly, leaving the most computationally intensive part—the BERT encoder—intact.. In the BERT architecture, we unfreeze several top layers to capture inter-token dependencies relevant for OpenIE. However, we provide an alternative version of our model with fully frozen BERT and an additional Transformer on top. This model has slightly worse performance but may be beneficial in the multilingual setting since it cannot lose multilingual information by unfreezing BERT.

Order-Agnostic Loss

To solve the issue of the predefined ordering of triplets, we design a special loss function with a bijective match between each predicted mask and the nearest ground truth triplet. Let N and M be the number of masks and ground truth triplets respectively; we choose M most relevant predictions and encourage them to match the ground truth exactly with the cross-entropy loss. To find the best matching, we calculate the $N \times M$ IoU (intersection-over-union) matrix between each probability mask and one-hot representation of the ground truth. We then maximize the sum of IoUs over all matches with the Hungarian algorithm (Kuhn 1955); see Fig. 2 for an example. During training, we do not compute exact labels predicted by our model to avoid thresholding, but rather directly calculate the average smooth IoU between predicted probability masks and labels as follows: $\mathrm{IoU}_{nm} = \frac{I_{nm}}{U_{nm}}$, where $I_{nm} = \sum_{t,c} p_{tnc} l_{tmc}$; $U_{nm} = \sum_{t,c} p_{tnc} + \sum_{t,c} l_{tmc} - I_{nm}$, p is the probability tensor predicted by the model, and l is the one-hot tensor of ground truth labels. The main drawback of this approach is that as the number of relations increases, the proportion of background (non-relation) tokens rapidly grows as well. We have taken measures against this induced class imbalance, discarding the background class in the IoU and reweighting non-background classes. We have also experimented with the focal loss instead of cross-entropy, but it did not yield any improvements, only shifted the precision-recall tradeoff.

Setup and Datasets

Experimental Setup

We implement our model in pytorch_lightning (Falcon 2019) with Hydra configuration framework (Yadan 2019). We use

Split	Dataset	# sentences	# tuples
Train	IMoJIE	91,725	190,661
	LSOIE	34,780	100,862
	Synth	10,000	41,645
Test	LSOIE	7,900	17,459
	CaRB	641	2,715
	MultiOIE2016	595	1,508

Table 1: Dataset statistics; the MultiOIE2016 and Synth numbers are given for each language.

Prob	Action
0.1	Concatenating the triplet and adding '.'
0.2	Triplet + conjunction + triplet.
0.35	Joining 3–5 concatenated triplets with ','
0.35	Joining 2–9 concatenated triplets with '.'

Table 2: Templates and their probabilities for *Synth*.

the pretrained *bert-base-multilingual-cased* BERT by *HuggingFace* (Wolf et al. 2020) for both English and multilingual benchmarks. During training, performance is measured as token-wise macro F1-score between predicted masks and ground truth labels. The best checkpoint is monitored along epochs. In case of IMoJIE data, 10% of the samples are selected as validation. In case of LSOIE, the validation was performed on the test split of the dataset. Our best model was trained with Adam optimizer with learning rate 5e-4 and weight decay 1e-6, batch size 32, unfreezing 4 top layers of BERT, $N = 20$ detections, matching based on IoU similarity metrics, and doubled weights of non-background classes. Typical training time until the best model is reached is about 1.5 hours on an NVIDIA Tesla V100 GPU. Inference speed is measured in sentences processed per second on a well-known set of 3,200 sentences (Stanovsky et al. 2018). We refer to results by Kolluru et al. (2020a) and assess the performance (Table 3) in a similar setting: batches of 32 processed on a single NVIDIA Tesla V100 GPU and 6 cores Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz without any additional model optimizations.

Datasets

Dataset statistics are summarized in Table 1. First, we use the recent **LSOIE** (*Large-Scale Open Information Extraction*) dataset (Solawetz and Larson 2021). For a fair comparison, we have also trained models on the dataset used by Kolluru et al. (2020a,b), called **IMoJIE** below. Another reason for training on two different datasets is that LSOIE differs in its annotation scheme from popular evaluation datasets such as OIE2016 (Stanovsky and Dagan 2016) and CaRB (Bhardwaj, Aggarwal, and Mausam 2019); e.g., in LSOIE auxiliary verbs such as "is" or "was" are (intentionally) not included into predicates, while CaRB adds more context into predicates than other OIE systems. We also experimented with adding *Wikidata*-based synthetic sentences during training (see below). Thus, in our experiments we use crowdsourced (LSOIE), prediction-based (IMoJIE), and synthetic data. All these approaches are discussed in detail below.

Wikidata-based synthetic sentences. For this dataset, called **Synth** below, we have used *Wikidata* (Vrandečić and Krötzsch 2014) to devise a simple sentence generation strategy. We lexicalize *Wikidata* triplets, joining the subject, object, and relation (predicate) by white spaces, e.g., `(Albert Einstein, is, physicist)` becomes "Albert Einstein is physicist". These phrases are used to generate sentences, e.g., "Albert Einstein is physicist while Amelia Mary Earhart is pilot." These sentences obviously have poor and highly standardized grammatical structure, but we have found that extra data of this quality leads to better relation extraction on multilingual benchmarks (see below). Sentences for *Synth* are generated with a set of templates selected with fixed probabilities (Table 2). *Wikidata* is a very rich data source, with nearly 9,000 different properties (predicates) that are highly imbalanced in the data, so *Synth* is also imbalanced w.r.t. predicates.

We used *Stanza* (Qi et al. 2020) for tokenization and PoS tagging. Tags are used for filtering aimed to create more grammatical sentences. Since synthetic sentences are simplistic, our model does not need too many examples of each type, although it important to have typological diversity in the samples. We generate data in two languages, Spanish and Portuguese.

LSOIE. We have used the LSOIE dataset (Solawetz and Larson 2021) prepared based on QA-SRL 2.0 (FitzGerald et al. 2018) that expands the scope of OIE2016 and AW-OIE (Stanovsky et al. 2018). The dataset can be converted to several different formats[1]; we have concatenated the **lsoie_(wiki|science)_*.conll** subsets as sources for our training and test data. Most datapoints in LSOIE are single sentences with a tuple of extractable subsequences; relations in LSOIE are N-ary. One of the elements usually represents a predicate (`P-B`, `P-I`), others are arguments (`A[N]-B`, `A[N]-I`). Since we are interested only in subject-relation-object triplets, we have removed the datapoints that do not have a predicate and at least two arguments. Then, predicates were converted into "relation" ("rel"/"pred"), arguments `A0-⋆` were converted into "subject" ("source", "arg1"), and arguments `AN-⋆` (for $N > 0$) were combined into a single argument defined as "object" ("target", "arg2").

IMoJIE (dataset). To provide a fair comparison with state of the art models such as OpenIE6 (Kolluru et al. 2020a) and IMoJIE (Kolluru et al. 2020b), we also used the large-scale dataset they were trained on[2] for training DetIE. The dataset consists of tuples extracted from *Wikipedia* sentences via OpenIE4 (Christensen, Soderland, and Etzioni 2011), ClausIE (Del Corro and Gemulla 2013), and RnnOIE (Stanovsky et al. 2018) and filtered with a *Score-and-Filter* technique proposed by Kolluru et al. (2020b). IMoJIE data is labelled for sequence *generation*: each sentence is assigned with a set of string tuples representing triplets. We find that the triplets are typically a combination of different pieces of the input sentence, so we apply a heuristic algorithm to retrieve them as masks for sequence labeling. The

[1] https://github.com/Jacobsolawetz/large-scale-oie

[2] https://github.com/dair-iitd/imojie

algorithm iteratively finds longest common substrings of a sentence and a triplet, converts them into labeled spans and excludes, until either sentence or triplet are exhausted; since spans are excluded, (S, R, O) masks are guaranteed to be disjoint. This method fails only if some tokens (e.g., "is") are understood implicitly and do not occur in the sequence. Similarly to Kolluru et al. (2020a), to cover such cases we append tokens "[is]", "[from]", and "[to]" to the end of a sequence, both in the converted IMoJIE and during prediction.

Next we describe our evaluation datasets: CaRB, LSOIE, and MultiOIE2016. We do not use two popular datasets, OIE2016 and WiRe57; the flaws of OIE2016 were discussed in (Zhan and Zhao 2020; Bhardwaj, Aggarwal, and Mausam 2019), while WiRe57 is almost 2x smaller than CaRB test set: only 57 sentences with 343 manually extracted facts.

CaRB (test). This evaluation dataset[3] is a subset of sentences from the OIE2016 dataset (Stanovsky and Dagan 2016) re-annotated via *Amazon Mechanical Turk* crowdsourcing. Annotators selected (arg1; rel; arg2) triplets and annotated time and location attributes if possible (Bhardwaj, Aggarwal, and Mausam 2019).

LSOIE (test). We have used a combination of test sets of $\text{LSOIE}_{\text{wiki}}$ and $\text{LSOIE}_{\text{science}}$.

MultiOIE2016 (test). This dataset is based on Re-OIE2016 (Zhan and Zhao 2020), a version of OIE2016 tailored for sequence tagging (unlike generation-oriented CaRB). Ro, Lee, and Kang (2020) extended it to Spanish and Portuguese; the number of sentences and tuples is the same for each language.

Evaluation Metrics

Following Kolluru et al. (2020a), we score model predictions against CaRB reference extractions, evaluated using the schemes introduced in OIE2016, WiRe57, CaRB, and OpenIE6 and discussed in detail below. Note that DetIE predicts a set of probabilities per token rather than confidence scores, but probabilities can be converted to "confidence" in many ways, e.g. aggregated max probability per argument or average $\log$ of probabilities.

OIE2016. We use a scheme proposed for evaluation by Stanovsky and Dagan (2016)[4], who compare systems in terms of precision and recall; the crucial step is to match predicted and ground truth extractions. A prediction is matched with the ground truth if they agree on the grammatical head of the elements (arguments and predicate).

WiRe57. Lechelle, Gotti, and Langlais (2019) introduced a new scoring procedure[5] that penalizes overly long extractions and assigns a token-level prediction quality score to all gold-prediction pairs for each sentence. Unlike the OIE2016 scorer, it considers all pairs of extractions. First, a predicted tuple is considered *possibly matching* a reference tuple from the same sentence if they share at least one token from each relation, argument 0 and argument 1 (with triplets, as in our case, this means at least one token in common for every corresponding element of the tuple). Then precision, recall, and F1 scores are computed for all possibly matching pairs of predicted t and reference g tuples as follows:

$$\begin{aligned}
\text{prec}(t,g) &= \tfrac{1}{|t|}\textstyle\sum_k |t^{(k)} \cap g^{(k)}|,\\
\text{rec}(t,g) &= \tfrac{1}{|g|}\textstyle\sum_k |t^{(k)} \cap g^{(k)}|,\\
\text{F1}(t,g) &= \tfrac{2\text{prec}(t,g)\text{rec}(t,g)}{\text{prec}(t,g)+\text{rec}(t,g)},
\end{aligned}$$

where $t^{(k)}$ is the bag of words/tokens representation of the kth part of the tuple and $|t|$ and $|g|$ are the numbers of tokens in the corresponding tuples.

Having computed the scores for all possibly matching pairs, we greedily construct the best matching. Overall system performance is measured by micro-averaged precision and recall. For more details we refer to (Lechelle, Gotti, and Langlais 2019) and the original implementation[6].

CaRB. This is a crowdsourced OIE2016 re-annotation initiative (Bhardwaj, Aggarwal, and Mausam 2019) already mentioned above; we use the evaluation scheme which here is different from the predecessors. CaRB uses filtering of stopwords and makes use of binary model predictions. CaRB scorer is not greedy; it constructs a matching table for all gold-predicted pairs, computes and averages maximum overall recall in each row, thus matching gold tuples with the best extraction. For precision, extractions are matched with gold annotations in a 1-1 fashion, then averaged as well. CaRB scorer uses only matches with at least one common word in the relation field. All higher order arguments (beyond (arg1; rel; arg2) triplets) are appended to the last argument. Illustrations of this single-to-many approach and a more detailed discussion are given in (Bhardwaj, Aggarwal, and Mausam 2019) and reference code[7].

CaRB (1-1). Used by Kolluru et al. (2020a), this evaluation scheme retains CaRB's similarity computation but uses a one-to-one mapping for both precision and recall, similar to OIE16 and Wire57.

Experiments and Results

Models and Systems in Comparison

We compare the proposed Det-IE model on the LSOIE and CaRB datasets with the following non-neural models: (1) MinIE (Gashteovski, Gemulla, and Corro 2017), (2) ClausIE (Del Corro and Gemulla 2013), (3) OpenIE4[8] (Christensen, Soderland, and Etzioni 2011), (4) OpenIE5[9] (Saha, Pal et al. 2017; Saha et al. 2018); and the following neural models: (5) IMoJIE (Kolluru et al. 2020b), (6) NeuralOIE (Cui, Wei, and Zhou 2018), (7) RnnOIE (Stanovsky et al. 2018), (8) SenseOIE (Roy et al. 2019), (9) SpanOIE (Zhan and Zhao 2020), (10) CIGL-OIE, (11) OpenIE6 (CIGL-OIE + IGL-CA)[10] (Kolluru et al. 2020a). SpanOIE is a span-based model. IMoJIE and NeuralOIE are generative models. RnnOIE, SenseOIE, CIGL-OIE, and OpenIE6 are sequence labeling models.

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

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

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

[6] https://github.com/rali-udem/WiRe57/blob/master/code/wire_scorer.py

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

[8] https://github.com/allenai/openie-standalone

[9] https://github.com/dair-iitd/OpenIE-standalone

[10] https://github.com/dair-iitd/openie6

Model	CaRB evaluation schemes							Speed
	CaRB		CaRB(1-1)		OIE16-C		Wire57-C	(sent./sec)
	F1	AUC	F1	AUC	F1	AUC	F1	
MinIE (Gashteovski, Gemulla, and Corro 2017)	41.9	-	38.4	-	52.3	-	28.5	8.9
ClausIE (Del Corro and Gemulla 2013)	45.0	22.0	40.2	17.7	61.0	38.0	33.2	4.0
OpenIE4 (Christensen, Soderland, and Etzioni 2011)	51.6	29.5	40.5	20.1	54.3	37.1	34.4	20.1
OpenIE5 (Saha, Pal et al. 2017; Saha et al. 2018)	48.0	25.0	42.7	20.6	59.9	39.9	35.4	3.1
SenseOIE (Roy et al. 2019)	28.2	-	23.9	-	31.1	-	10.7	-
SpanOIE (Zhan and Zhao 2020)	48.5	-	37.9	-	54.0	-	31.9	19.4
RnnOIE (Stanovsky et al. 2018)	49.0	26.0	39.5	18.3	56.0	32.0	26.4	149.2
NeuralOIE (Cui, Wei, and Zhou 2018)	51.6	32.8	38.7	19.8	53.5	37.0	33.3	11.5
IMoJIE (Kolluru et al. 2020b)	53.5	33.3	41.4	22.2	56.8	39.6	36.0	2.6
IGL-OIE (Kolluru et al. 2020a)	52.4	33.7	41.1	22.9	55.0	36.0	34.9	142.0
CIGL-OIE (Kolluru et al. 2020a)	**54.0**	35.7	42.8	24.6	59.2	40.0	36.8	142.0
OpenIE6 (Kolluru et al. 2020a)	52.7	33.7	**46.4**	26.8	65.6	48.4	**40.0**	31.7
DetIE$_{\text{LSOIE}}$ (ours)	43.0	27.2*	33.1	18.3*	49.7	32.7*	31.2	**708.6**
DetIE$_{\text{IMoJIE}}$ (ours)	52.1	**36.7***	40.1	24.0*	56.0	38.7*	36.0	**708.6**
DetIE$_{\text{LSOIE}}$ (ours) + IGL-CA from OpenIE6	39.6	26.7*	36.3	22.7*	63.3	47.9*	33.5	112.2
DetIE$_{\text{IMoJIE}}$ (ours) + IGL-CA from OpenIE6	47.3	35.1*	43.1	**29.3***	**67.7**	**54.0***	37.8	112.2

Table 3: Comparison on CaRB test set with scoring schemes from CaRB (Bhardwaj, Aggarwal, and Mausam 2019), CaRB (1-1) (Kolluru et al. 2020a), OIE2016 (Stanovsky and Dagan 2016), WiRe57 (Lechelle, Gotti, and Langlais 2019). Results for all models except DetIE are cited from (Kolluru et al. 2020a). Best results are shown in bold; second best, underlined. DetIE does not provide confidence scores, so ROC-AUC values are approximated from a single TPR-FPR point.

Model	F1	AUC
OllIE (Mausam et al. 2012)	36.8	16.7
ReVerb (Fader, Soderland, and Etzioni 2011)	36.8	16.9
OpenIE4	54.6	32.3
OpenIE5	49.5	25.8
CIGL-OIE	59.7	48.0
OpenIE6 (CIGL-OIE + IGL-CA)	51.6	32.7
DetIE$_{\text{IMoJIE}}$	55.7	44.9*
DetIE$_{\text{IMoJIE}}$ (ours) + IGL-CA	45.9	41.7*
DetIE$_{\text{LSOIE}}$	**71.4**	**61.3***
DetIE$_{\text{LSOIE}}$ + IGL-CA	58.7	55.9*

Table 4: Comparison on combined LSOIE test sets (Solawetz and Larson 2021) with the original CaRB evaluation scheme (Bhardwaj, Aggarwal, and Mausam 2019). Best results are shown in bold; second best, underlined. DetIE does not provide confidence scores, so ROC-AUC values are approximated from a single TPR-FPR point.

In experiments with monolingual data, we additionally preprocess raw sentences in the test set using a coordination analysis (CA) model IGL-CA following OpenIE6 approach (Kolluru et al. 2020a)[11] which considers CA as a grid labeling problem and is trained on the coordination-annotated Penn Treebank (Ficler and Goldberg 2016). We first apply CA to sentences in the test set and then apply DetIE to the resulting "simplified" texts, attributing the extractions to the corresponding original sentences with neither post-filtering nor rescoring.

We also compare DetIE with three systems on MultiOIE2016: rule-based multilingual systems (1) ArgOE (Gamallo and Garcia 2015) and (2) PredPatt (White et al. 2016) and (3) neural BERT-based Multi^{2}OIE system (Ro, Lee, and Kang 2020).

Lang.	Model	F1	Prec.	Rec.
EN	ArgOE	43.4	56.6	35.2
	PredPatt	53.1	53.9	52.3
	Multi^{2}OIE	69.3	66.9	71.7
	DetIE$_{\text{IMoJIE}}$ (ours)	78.7	85.4	69.9
	DetIE$_{\text{IMoJIE+Synth}}$ (ours)	**79.3**	**87.1**	**72.8**
ES	ArgOE	39.4	48.0	33.4
	PredPatt	44.3	44.8	43.8
	Multi^{2}OIE	60.2	59.1	61.2
	DetIE$_{\text{IMoJIE}}$ (ours)	73.2	83.7	65.0
	DetIE$_{\text{IMoJIE+Synth}}$ (ours)	**75.0**	**85.6**	**66.8**
PT	ArgOE	38.3	46.3	32.7
	PredPatt	42.9	43.6	42.3
	Multi^{2}OIE	59.1	56.1	62.5
	DetIE$_{\text{IMoJIE}}$ (ours)	74.7	85.1	66.6
	DetIE$_{\text{IMoJIE+Synth}}$ (ours)	**75.0**	**86.0**	**69.4**

Table 5: Binary extraction performance on Multi-OIE2016 (Zhan and Zhao 2020) measured with CaRB's evaluation scheme. Results for models other than DetIE are cited from (Ro, Lee, and Kang 2020).

Results

Tables 3 and 4 report the quality metrics and performance comparisons across all metrics for the CaRB and LSOIE datasets respectively (DetIE and IGL-CA inference times were estimated separately, on an NVIDIA Tesla V100). In Table 3, IGL-CA is a pretrained coordination analysis model by Kolluru et al. (2020a), used as discussed above. Table 3 shows that variations of DetIE improve upon the state of the art for almost all evaluation schemes on CaRB. For LSOIE,

[11] https://github.com/dair-iitd/openie6

as expected, DetIE$_{\mathrm{LSOIE}}$ performs best on LSOIE test set (Table 4) since it was trained on data with the same annotation principles. Note, however, that even training on a different dataset (IMoJIE) allows DetIE models to be among the best-performing ones. Our hypothesis is that our training scheme being able to capture multiple relations at once allows the underlying Transformer architecture to better use its ability to perceive relations in a text for extraction.

Results for MultiOIE2016 are presented in Table 5. Here, the DetIE$_{\mathrm{IMoJIE}}$ model trained on IMoJIE significantly outperforms previous approaches. The margin in the F1 evaluation metric reaches 15.6% on the Portuguese part of the dataset, while for English it is 8.6%. Interestingly, Multi^{2}OIE shows better performance on English in terms of recall, although the gap between it and DetIE is only 1.8%.

Effectively, training on IMoJIE makes DetIE a zero-shot model for Spanish and Portuguese, since IMoJIE is collected entirely in English. Thus, we decided to fuse training data with *Synth* for our model; the result is shown as DetIE$_{\mathrm{IMoJIE+Synth}}$. Synthetic data adds another 1.8% of F1 for Spanish, 0.6% for English, and 0.3% for Portuguese. This model outperforms previous state of the art w.r.t. all metrics, including recall on English.

Note that Multi^{2}OIE is a two-stage approach, thus it is run at least twice for each sentence: one time to extract predicates and possibly several times for the predicates, one for each. In contrast, DetIE is a single-shot model; we extract all relations at once, only capping at the number of possible extractions[12]. We believe that this is the main reason why our model is so much faster during inference.

Discussion and Error Analysis

We have analyzed the outputs of DetIE$_{\mathrm{IMoJIE}}$ on a random sample of 100 sentences from the CaRB validation set (Table 6). According to our analysis, DetIE is prone to aggregating conjunctions and comparisons into a single triplet (Sent. #1, #2), which explains the improvements from coordination analysis in Table 3. Occasionally, we observed incorrect prediction of triplets in sentences with coreference (Sent. #3). Since our model uses a single pass for relation extraction, we hypothesise that it could be applied to whole passages of text (multiple sentences) and extract relations which permeate the limits of a single sentence. Thus, the coreference task could be done alongside with relation extraction in end-to-end manner; we leave this for further work.

The performance of DetIE on CaRB varies widely across training sets, which is expected since CaRB is only a test set and has its own markup scheme, different from schemes used in training datasets. There is no conventional gold standard training set for OpenIE: IMoJIE was obtained with an advanced bootstrapping scheme, OpenIE6 (SotA) was trained on it, and LSOIE is the latest published dataset of size suitable for neural models but it differs significantly, hence the difference in performance.

We hypothesise that one-hot-encoded PoS and/or dependencies head labels or appending the dependency head's embedding to each token's embedding could have improved the results. We believe so since many early OpenIE models did heavily rely on syntax (e.g, OpenIE5, OpenIE6), but it remains to be tested in further work.

Sent. #1	Males had a median income of $ 28,750 versus $ 16,250 for females.
Gold	(Males; had a median income of; $ 28,750) (females; had a median income of; $ 16,250)
DetIE	(Males; had; a median income of $ 28,750 versus $ 16,250 for females)
Sent. #2	Hapoel Lod played in the top division during the 1960s and 1980s, and won the State Cup in 1984.
Gold	(Hapoel Lod; played in; the top division; during the 1960s) (Hapoel Lod; played in; the top division; during the 1980s) (Hapoel Lod; won; the State Cup; in 1984)
DetIE	(Hapoel Lod; played; in the top division during the 1960s and 1980s) (Hapoel Lod; won; the State Cup in 1984)
Sent. #3	A spectrum from a single FID has a low signal-to-noise ratio, but fortunately it improves readily with averaging of repeated acquisitions.
Gold	(A spectrum from a single FID; has; a low signal-to-noise ratio) (signal-to-noise ratio; improves readily with averaging of; repeated acquisitions)
DetIE	(A spectrum from a single FID; has; a low signal-to-noise ratio) (it; improves; readily with averaging of repeated acquisitions)

Table 6: Sample sentences with gold annotations and relations predicted by DetIE$_{\mathrm{IMoJIE}}$.

Conclusion

We have introduced a novel DetIE model for the OpenIE task; it is based on the ideas of single-shot object detection in computer vision and extracts multiple triplets in a single pass. The proposed model is atomic and can be used as a part of OIE pipelines that may include coordination analysis, rescoring, syntactic chunks collapsing etc. Our approach outperforms existing state of the art on the LSOIE dataset and performs at least on par or better for every considered evaluation scheme on CaRB. The DetIE model is 5x faster than previous state of the art in terms of inference speed.

Moreover, DetIE has shown excellent performance in the zero-shot cross-lingual setting, exceeding existing state of the art for Spanish and Portuguese by 13% and 15% respectively. We have also introduced a technique for multilingual synthetic data generation and used it to generate additional training data that further improved the results (by 1.8% in Spanish and 0.3% in Portuguese).

As a first step for future work, our method may benefit from enrichment with PoS tags, syntactic information (e.g., `deprel` tags), and traditional neural sequence labeling layers (e.g., CRF). We also plan to experiment with other possible improvements in the model architecture.

[12] It serves as a hyperparameter of our model; we used 100 possible extractions as a large upper bound for a single sentence.

Acknowledgements

The work of Sergey Nikolenko was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated November 2, 2021 No. 70-2021-00142. The work of Elena Tutubalina was supported by a grant from the President of the Russian Federation for young scientists-candidates of science (MK-3193.2021.1.6). We are grateful to Daria Grischenko and Maxim Eremenko for their highly valuable help with organizing this work.

References

Balasubramanian, N.; Soderland, S.; Mausam; and Etzioni, O. 2013. Generating Coherent Event Schemas at Scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1721–1731. Seattle, Washington, USA: Association for Computational Linguistics.

Bhardwaj, S.; Aggarwal, S.; and Mausam, M. 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)*, 6263–6268. Hong Kong, China: Association for Computational Linguistics.

Christensen, J.; Soderland, S.; and Etzioni, O. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, 113–120.

Cui, L.; Wei, F.; and Zhou, M. 2018. Neural Open Information Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 407–413. Melbourne, Australia: Association for Computational Linguistics.

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

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12): 68–74.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, 1535–1545.

Falcon, W. A. e. a. 2019. PyTorch Lightning. GitHub.

Fan, A.; Gardent, C.; Braud, C.; and Bordes, A. 2019. Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs. *CoRR*, abs/1910.08435.

Ficler, J.; and Goldberg, Y. 2016. Coordination Annotation Extension in the Penn Tree Bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 834–842.

FitzGerald, N.; Michael, J.; He, L.; and Zettlemoyer, L. 2018. Large-Scale QA-SRL Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2051–2060.

Gamallo, P.; and Garcia, M. 2015. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, 711–722. Springer.

Gashteovski, K.; Gemulla, R.; and Corro, L. d. 2017. Minie: minimizing facts in open information extraction. Association for Computational Linguistics.

Kolluru, K.; Adlakha, V.; Aggarwal, S.; Mausam; and Chakrabarti, S. 2020a. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. arXiv:2010.03147.

Kolluru, K.; Aggarwal, S.; Rathore, V.; Chakrabarti, S.; et al. 2020b. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5871–5886.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Lechelle, W.; Gotti, F.; and Langlais, P. 2019. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, 6–15.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2015. SSD: Single Shot MultiBox Detector. *CoRR*, abs/1512.02325.

Lukasik, M.; Jain, H.; Menon, A. K.; Kim, S.; Bhojanapalli, S.; Yu, F.; and Kumar, S. 2020. Semantic Label Smoothing for Sequence to Sequence Problems. arXiv:2010.07447.

Mathur, N.; Baldwin, T.; and Cohn, T. 2020. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics. arXiv:2006.06264.

Mausam. 2016. Open Information Extraction Systems and Downstream Applications. In *IJCAI*.

Mausam; Schmitz, M.; Bart, R.; Soderland, S.; and Etzioni, O. 2012. Open Language Learning for Information Extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Ponza, M.; Del Corro, L.; and Weikum, G. 2018. Facts That Matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1043–1048.

Brussels, Belgium: Association for Computational Linguistics.

Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; and Manning, C. D. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Ramshaw, L. A.; and Marcus, M. P. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, 157–176. Springer.

Ro, Y.; Lee, Y.; and Kang, P. 2020. Multiˆ 2OIE: Multilingual Open Information Extraction based on Multi-Head Attention with BERT. *arXiv preprint arXiv:2009.08128*.

Roy, A.; Park, Y.; Lee, T.; and Pan, S. 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)*, 728–737. Hong Kong, China: Association for Computational Linguistics.

Saha, S.; Pal, H.; et al. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 317–323.

Saha, S.; et al. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2288–2299.

Schmitz, M.; Soderland, S.; Bart, R.; Etzioni, O.; et al. 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*, 523–534.

Solawetz, J.; and Larson, S. 2021. LSOIE: A Large-Scale Dataset for Supervised Open Information Extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2595–2600. Online: Association for Computational Linguistics.

Stanovsky, G.; and Dagan, I. 2016. Creating a Large Benchmark for Open Information Extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (to appear). Austin, Texas: Association for Computational Linguistics.

Stanovsky, G.; Dagan, I.; and Mausam. 2015. Open IE as an Intermediate Structure for Semantic Tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 303–308. Beijing, China: Association for Computational Linguistics.

Stanovsky, G.; Michael, J.; Zettlemoyer, L.; and Dagan, I. 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)*, 885–895. New Orleans, Louisiana: Association for Computational Linguistics.

Tan, M.; Pang, R.; and Le, Q. V. 2020. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

White, A. S.; Reisinger, D.; Sakaguchi, K.; Vieira, T.; Zhang, S.; Rudinger, R.; Rawlins, K.; and Van Durme, B. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1713–1723.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Yadan, O. 2019. Hydra - A framework for elegantly configuring complex applications. Github.

Yates, A.; Banko, M.; Broadhead, M.; Cafarella, M.; Etzioni, O.; and Soderland, S. 2007. TextRunner: Open Information Extraction on the Web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 25–26. Rochester, New York, USA: Association for Computational Linguistics.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2335–2344.

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