DocOIE: A Document-level Context-Aware Dataset for OpenIE

Kuicai Dong[1], Yilin Zhao[1], Aixin Sun[1], Jung-Jae Kim[2], Xiaoli Li[1,2]

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

{kuicai001, zhao0320}@e.ntu.edu.sg, {axsun, xlli}@ntu.edu.sg

[2] Institute for Infocomm Research, A*STAR, Singapore

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

Abstract

Open Information Extraction (OpenIE) aims to extract structured relational tuples (subject, relation, object) from sentences, and plays a critical role in many NLP applications. Existing solutions perform extraction at *sentence level*, without referring to any additional contextual information. In reality, however, a sentence typically exists as part of a document rather than standalone; we often need to access relevant contextual information around the sentence before we can accurately interpret it. As there is no document-level context-aware OpenIE dataset available, we manually annotate 800 sentences from 80 documents in two domains (Healthcare and Transportation) to form a DocOIE dataset for evaluation. In addition, we propose DocIE, a document-level context-aware OpenIE model. Our experimental results demonstrate that incorporating document-level context is helpful in improving OpenIE performance. Both the DocOIE dataset and DocIE model are available online.[1]

1 Introduction

Open Information Extraction has been a critical NLP task as it can extract structured relational tuples (*subject, relation, object*) from unstructured text. The OpenIE system is fully domain-independent, and does not need input from users. It is also highly scalable and allows fast querying mechanism (Yates et al., 2007). Therefore, OpenIE has been successfully applied to a variety of downstream NLP tasks, such as knowledge base population (Martínez-Rodríguez et al., 2018; Gashteovski et al., 2020), question answering (Khot et al., 2017), and summarization (Fan et al., 2019).

Current OpenIE methods mainly focus on extracting tuples *at sentence level*. However, in many NLP scenarios, sentences exist as part of a document rather than standalone. Given a document

Figure 1: Example sentences with ambiguity.

corpus, if we simply apply existing sentence-level OpenIE models to extract tuples, we could miss some useful and critical document-level contextual information, leading to unsatisfying results. We use the two example sentences in Fig. 1 to illustrate two types of ambiguities.

Part-of-speech Ambiguity. The word "transfers" can be a verb or a noun. Accordingly, two tuples could be extracted from the first example sentence, listed in Fig. 1. This ambiguity can be resolved by the main verb of the sentence "might not be", which is far away from "transfers", and thus is not considered by many existing OpenIE systems. However, context sentences S1, S2, and S3 in the document suggest that "data transfers" shall be considered as a noun phrase throughout this document.

Syntactic Ambiguity. The second example sentence does not have an explicit clue about the re-

---

[1] https://github.com/daviddongkc/DocOIE

lationship between "a device" and "a cellular base station".[2] Thus, existing OpenIE systems often fail to split them, but incorrectly extract the first tuple (node-B; can be; a device a cellular base station). However, context sentence S4 includes an explicit cue to the relationship between the two terms and may thus help split them.

To minimize the aforementioned ambiguities, it is clear that we should leverage document-level context. However, all existing OpenIE datasets are generated or annotated at sentence level. These datasets include standalone sentences but not their context sentences. Hence they are not suitable for evaluating context-aware tuple extraction.

We annotate a **Doc**ument-level context-aware **O**pen **I**nformation **E**xtraction (**DocOIE**) dataset. DocOIE consists of 800 expert-annotated sentences from 80 documents, where 10 sentences are randomly sampled for annotation from each of the 80 documents. To the best of our knowledge, among all OpenIE datasets as of now, DocOIE contains the largest number of *expert-annotated* sentences.[3] More importantly, DocOIE provides document-level contexts, enabling OpenIE models to take relevant contexts for accurate tuple extraction.

Furthermore, to show that document-level context is useful for OpenIE task, we develop the **Doc**ument-level context-aware Open **I**nformation **E**xtraction (**DocIE**) model. DocIE encodes a source sentence with its contextual information by using pre-trained BERT (Devlin et al., 2019). Because contextual sentences can be much longer than the source sentence, the syntactic/semantic information in source sentence might be dominated by that of the contexts. Our proposed DocIE model differentiates the source sentence and its contexts by segment tags, and adding additional transformer encoder layers only for the source sentence.

In summary, our contributions are threefold:

- We propose a new task in OpenIE to extract relational tuples with document-level contexts.
- We introduce **DocOIE**, an expert-annotated dataset for evaluating document-level OpenIE systems. DocOIE consists of 2,122 relational tuples from 800 annotated sentences, with their document-level contexts.

---

[2]The missing of a comma could be a typo in the patent document. Nevertheless, not all input sentences to OpenIE system are typo free in real applications.

[3]Expert annotation means annotations are made by a person who is familiar with the OpenIE task and OpenIE models. This term is used to distinguish annotations by crowdsourcing.

Dataset	#Sent.	Source	Annotation
OIE2016	3,200	QA-SRL	Automatic
Wire57	57	Wikipedia Newswire	Expert
CaRB	1,282	OIE2016	Crowdsourcing
CaRB	50	OIE2016	Expert
DocOIE	800	Patent	Expert

Table 1: Existing OpenIE datasets with number of sentences, sentence source, and annotation type.

- We present **DocIE**, a neural OpenIE system that can leverage document-level contexts for relational tuple extraction.

2 Related Work

OpenIE Datasets. Since introduction of OpenIE task by Yates et al. (2007), the earlier systems have been mainly evaluated by using a small number of sentences, without a standardized evaluation procedure (Niklaus et al., 2018). OIE2016 (Stanovsky and Dagan, 2016) is the first large-scale dataset constructed for OpenIE tasks and comes with a standard scoring framework. In OIE2016, the gold tuples are automatically generated from a QA-SRL dataset (He et al., 2015) according to human crafted rules. Wire57 (Lechelle et al., 2019) improves the scorer and manually annotates 57 sentences as a benchmark dataset. Considering that OIE2016 dataset is noisy, Bhardwaj et al. (2019) provide a crowdsourcing dataset named CaRB. CaRB also has 50 expert-annotated sentences and a sophisticated scoring framework.

As summarized in Table 1, the number of expert-annotated sentences in these datasets remains small. Furthermore, the sentences in these datasets do not come with contextual information. In contrast, our DocOIE dataset consists of 800 expert-annotated sentences, and comes with the source documents for accurate sentence interpretation.

OpenIE Models. TextRunner (Yates et al., 2007) is the first OpenIE system, followed by Reverb (Fader et al., 2011), OLLIE (Mausam et al., 2012), Clausie (Corro and Gemulla, 2013), CSD-IE (Bast and Haussmann, 2013), Stanford OpenIE (Angeli et al., 2015), Openie4 (Mausam, 2016), Openie5[4], NESTIE (Bhutani et al., 2016), MINIE (Gashteovski et al., 2017) and

---

[4]`github:dair-iitd/openie-standalone`

Graphene (Cetto et al., 2018). We consider them as *traditional* OpenIE models. These models extract relational tuples based on handcrafted rules or statistical methods. They usually rely on prior syntactic or semantic analysis. Consequently, any error accumulated in the prior stages deteriorates model performance.

Recently, neural OpenIE systems have been developed and showed promising results (Cui et al., 2018; Zhan and Zhao, 2020; Kolluru et al., 2020a,b). Different from the traditional models, neural OpenIE models extract tuples in an end-to-end manner, not requiring prior syntactic or semantic analysis. In principle, the traditional rule-based or statistical OpenIE models do not need training. However, neural OpenIE models need a large number of training samples to learn the extraction patterns. For instance, IMOJIE (Kolluru et al., 2020b) uses about 100,000 sentences for model training. It is unrealistic and expensive to manually annotate 100,000 sentences simply for training purpose. Therefore, a common practice in learning a neural OpenIE model is to use tuples automatically extracted by traditional systems as training data, *i.e., a bootstrapping strategy*. We consider these imperfect training labels generated via bootstrapping as pseudo labels. The pseudo labels used in (Cui et al., 2018) are by Openie4, and those in (Kolluru et al., 2020b) are from multiple OpenIE systems.

To the best of our knowledge, no OpenIE models consider document-level contexts in the tuple extraction. Nonetheless, our neural model unavoidably requires extractions of pseudo labels bootstrapped from traditional models for training. To ensure reproducibility, as part of the DocOIE dataset, we also release the document IDs that are used for generating the pseudo labels.

3 DocOIE Dataset

We now present our **Doc**ument-level context-aware **O**pen **I**nformation **E**xtraction (**DocOIE**) dataset. We first introduce the data selection and collection process, and then the annotation process by two experts. Moreover, we explain our annotation consistency measurement to indicate the high-level annotation consistency in DocOIE. In summary, DocOIE consists of two datasets: *evaluation dataset* and *training dataset*.

Evaluation dataset contains 800 expert-annotated sentences, sampled from 80 documents in two domains (healthcare and transportation).

Item	#Item	Metric	Average	Min~Max
Doc	80	N_{sent}	129.85	62~218
Sent	800	L_{sent}	22.70	5~47
		N_{tuple}	2.65	1~8
Tuple	2,122	L_{sub}	3.70	1~17
		L_{rel}	3.39	1~12
		L_{obj}	3.94	0~27

Table 2: Statistics of DocOIE evaluation dataset. $N_{\{\cdot\}}$ denotes the number of units (*i.e.,* sentence or tuple); $L_{\{\cdot\}}$ denotes the length (number of words) of the unit.

Specifically, 10 sentences are sampled for annotation from each of the 40 documents in one domain. In total, 2,122 relational tuples are annotated in the 800 sampled sentences (refer to Table 2 for detailed statistics, and Table 5 for fine-grained analysis).

Training dataset contains 2,400 documents from the two domains (healthcare and transportation); 1,200 documents in each domain. All sentences from these documents are used to bootstrap pseudo labels for neural model training.[5]

3.1 Dataset Collection

OpenIE, by definition, is to extract relational tuples in open domain. Ideally, sentences/documents in DocOIE dataset shall not be restricted to any particular document type or topical domain. However, it is challenging to include all types of documents and annotate them. In fact, all existing annotations are restricted to specific types of documents like news and Wikipedia articles (Niklaus et al., 2018).

Document Type Selection In building DocOIE, we focus on formally written documents and leave it for future work to explore other kinds of documents. We select the type of formal documents with four criteria: (i) **Adequacy:** as a document-level context-aware dataset, each document shall have a reasonable number of sentences to provide sufficient context. (ii) **Consistency:** each document shall focus on a central topic. In such a way, sentences within the same document are correlated to and consistent with each other, which helps derive proper context. (iii) **Informativeness:** a document is considered informative if it contains informative entities like technical concepts, relations,

---

[5]Only document IDs are included in DocOIE, for document collection at http://patft.uspto.gov/

and events. Intuitively, OpenIE models are more useful for extracting factual tuples in informative documents. (iv) **Syntactic Variety:** sentences in these documents shall vary in syntactic structures. Such variety facilitates thorough evaluation of OpenIE models under different scenarios.

Patent Document Collection   After taking all factors into consideration, we choose to collect patent documents from PatFT.[6] Each patent document elaborates one specific invention in reasonable length, providing sufficient contexts to annotators. They are rich in informativeness by nature, and the documents contain rich syntactic structures.

Through PatFT search engine, patent documents can be retrieved by keywords. We have two considerations for keyword selection: (i) **Magnitude:** as part of DocOIE, a large number of documents shall be available for training neural OpenIE models. Hence, the keywords shall lead to sufficient patent documents. (ii) **Diversity:** the collected patent documents are expected to be diversified in inventors, organizations, filed date, etc., to avoid fixed patterns, hence to ensure diversity of our dataset.

As the result, we choose three broad and non-technical keywords: "healthcare", "traffic", and "transportation", to collect documents in two broad domains, *healthcare* and *transportation*. Reported in Table 3, 42,514 and 32,256 documents are collected in healthcare and transportation respectively. Documents in each domain are contributed by more than 40,000 inventors from over 8,000 cities, and the filed dates range in several decades.

We clean these documents by removing non-textual components in them. Then, by length (in number of words) the shortest 10% and longest 10% documents are removed to avoid extremely short/long documents in our dataset. The remaining documents form the corpus from which we sample (i) documents for annotation, and (ii) training documents for bootstrapping pseudo labels.

3.2   DocOIE Evaluation Dataset Selection

To ensure annotation quality and consistency, we choose to follow expert annotation scheme instead of crowdsourcing adopted in CaRB (Bhardwaj et al., 2019). As we discussed in Section 1, sentences exist as part of a document rather than standalone. To gain an accurate interpretation of a sentence, the annotator needs to read a few surround-

Patents	Healthcare	Transportation
#Document	42,514	32,256
#Inventor	74,266	42,286
#Organization	10,888	7,919
#City	10,493	8,095
Filed year	1999~2020	1970~2020

Table 3:   Properties of collected patents in healthcare and transportation domains.

ing sentences, or even the entire document, for relevant contexts. Hence, the choices of labelling one sentence or multiple sentences per document incur different costs.

To be able to cover a reasonable number of documents and also balance the annotation workload, we choose to randomly sample 10 sentences per document from 80 documents for annotation. Recall that the average number of sentences per document is 101.78 (refer to Table 2). The 10 sentences annotated in a document can be used to evaluate context-aware OpenIE at 10 different positions in this document. In this way, our annotation covers 80 documents with considerable diversity.

In summary, we randomly selected 80 documents (40 in each domain) from the documents collected in Section 3.1. Then we randomly selected 10 sentences from each document. These 80 documents, along with 800 expert-annotated sentences form the DocOIE *evaluation dataset*.

3.3   Annotation Consistency Measurement

The annotation was performed by two OpenIE experts (both are authors of this paper) with reference to existing annotation processes (Stanovsky and Dagan, 2016; Bhardwaj et al., 2019). The dataset was annotated in three stages.

In the first stage, the two annotators practiced annotations independently on 100 sentences among the 800 sentences. Then they cross-validated the annotation results, discussed them to resolve disagreements, and updated annotation policy.

In the second stage, the two experts independently annotated another 100 sentences among the remaining 700 sentences. These two sets of annotations are used for measuring annotation consistency. Because it is not straightforward to evaluate annotation agreement by measures like Kappa coefficient, we adopted the evaluation scorer proposed by CaRB (Bhardwaj et al., 2019). The scorer performs matching at tuple level instead of lexical

Consistency	Precision	Recall	F1
A←B	90.7	92.4	91.6
B←A	84.6	92.0	88.2
Average	87.7	92.2	89.9

Table 4: Annotation consistency estimated between annotators A and B. A←B indicates evaluation of A's annotations with B's annotations as ground truth.

Analysis	Healthcare	Transportation
Sentence semantic analysis (%)		
Conjunction	61.00	61.50
Terminology	60.00	42.75
Dependent Clause	29.00	41.50
Tuple semantic analysis (%)		
Negative Polarity	3.26	5.16
Possibility	12.17	8.74
Under-specificity	8.80	6.49

Table 5: Fine-grained statistical analysis for 800 annotated sentences and 2,122 tuples. Percentages refer to the respective domain: Healthcare and Transportation.

level. Specifically, we score one expert's annotations by treating the other's annotations as ground truth. Among the tuple matching strategies in CaRB, we used the default binary lenient tuple matching, to estimate the consistency between the two annotators. Reported in Table 4, the two annotators reach high-level agreement in annotations with an average F1 of 89.9%.

Based on the high-level annotation consistency, in the third stage, each expert independently annotated 300 sentences from the remaining 600 sentences. The annotations are then validated by the other expert, and annotation disagreements are resolved through discussion.

3.4 Analysis of DocOIE Evaluation Dataset

To understand the difficulty of DocOIE, we provide an analysis of the annotated sentences and tuples in DocOIE, similar to (Gashteovski et al., 2019).

Sentence-level Analysis. We evaluate the complexity of a sentence on whether it contains conjunction word, terminology mention, and dependent clause. Consider the following sentence as an example: *"Though depicted as a distinct step, it may be performed as part of the VAD or ASR*

processes." *"VAD processes"* and *"ASR processes"* are terminology mentions. The conjunction word *"or"* in the sentence connects these two terminologies. In addition, dependent clause refers to the subordinate clause "Though depicted as a distinct step". Table 5 reports the percentages of sentences that contain at least one conjunction word, terminology mention, and dependent clause, respectively.

Tuple-level Analysis. Consider an example sentence *"They may be implemented as instructions stored on a machine-readable medium."* Two tuples can be extracted from this sentence: (1) *"they ; may be implemented as ; instructions"*, and (2) *"instructions ; are stored on ; a machine-readable medium"*.

We analyze a tuple on whether it is a certainty or merely a possibility. The relation *"may be implemented as"* indicates tuple (1) a possibility because of the modal verb *"may"*. Then, negative polarity refers to certainty negation, which can be indicated by words such as *"not"* or *"no"*. Some tuples are under-specified in either tuple subject or object. The main reason for under-specificity is the lack of co-reference information, where we need additional context to obtain a coherent meaning. In tuple (1), its subject *"they"* is under-specified because co-reference information is needed to resolve what *"they"* refers to.[7] Table 5 reports the percentages of tuples that are under each of the categories.

3.5 DocOIE Training Dataset

Besides the 80 documents for expert annotations, we further sample 2,400 documents randomly (1,200 in each domain) from the documents collected in Section 3.1 to create DocOIE training dataset. The 1,200 documents in each domain contain around 120,000 sentences, which is sufficient for pseudo label generation, required by neural OpenIE models.

4 Pseudo Label by Bootstrapping

Following the common practice (Kolluru et al., 2020b; Cui et al., 2018; Zhao et al., 2020), we generate pseudo labels by bootstrapping with traditional OpenIE models. Before we run these models on the DocOIE training dataset, we evaluate their performances on the DocOIE evaluation dataset, to select the models which can generate better quality pseudo labels.

---

[7]Co-reference is not annotated in DocOIE.

OpenIE Model	AUC	Prec	Rec	F1
Healthcare Domain				
Reverb	35.4	**79.9**	42.8	55.8
Stanford	16.5	11.0	29.7	16.1
Clausie	22.1	38.8	53.8	45.1
OpenIE4	35.4	59.5	<u>55.1</u>	57.2
OpenIE5	29.1	53.6	50.5	52.0
Rev+Oie4	**36.8**	<u>75.8</u>	47.7	**58.6**
Oie4+Rev	<u>35.8</u>	59.6	**55.3**	<u>57.4</u>
Transportation Domain				
Reverb	29.3	**79.1**	36.3	49.7
Stanford	15.7	13.2	27.8	17.9
Clausie	18.0	36.2	48.4	41.4
OpenIE4	29.2	52.8	<u>51.2</u>	52.0
OpenIE5	25.0	50.9	43.8	47.1
Rev+Oie4	**31.0**	<u>74.2</u>	42.4	**54.0**
Oie4+Rev	<u>30.1</u>	53.4	**52.7**	<u>53.0</u>

Table 6: Performance of OpenIE models on DocOIE evaluation dataset. The best scores are in boldface and second best scores are underlined.

We evaluate the models by using CaRB scorer (Bhardwaj et al., 2019). Table 6 reports the performance of five independent OpenIE models: Reverb (Fader et al., 2011), Clausie (Corro and Gemulla, 2013), Stanford OpenIE (Angeli et al., 2015), OpenIE4 (Mausam, 2016) and OpenIE5[8]. In addition to these five models, we also evaluated two combinations of Reverb and OpenIE4. With **Rev+Oie4**, Reverb is the main system and if Reverb fails to extract any tuples from a sentence, we complement the extraction by using Openie4. Similarly, **Oie4+ Rev** uses OpenIE4 as the main system, and the extractions are complemented by Reverb.

All the evaluated models show consistent performance in both domains. By F1 score, both Reverb and OpenIE4 are the best performing individual models and their combinations lead to the best and second best F1 scores in both domains. Accordingly, by applying OpenIE4, Reverb, and their combinations, the number of sentences and tuples-extracted from the DocOIE training dataset are reported in Table 7.[9] Note that, a sentence is not counted if it has no extracted tuples, which leads to the different sentence number.

[8] github:dair-iitd/openie-standalone

[9] The number of sentences may vary because of different text pre-processing and sentence segmentation strategies adopted. The number of tuples extracted may vary due to different versions of OpenIE tools.

OpenIE model	Healthcare		Transportation	
	#Sent	#Tuple	#Sent	#Tuple
OpenIE4	117k	263k	111k	258k
Oie4+Rev	121k	268k	114k	262k
Reverb	103k	146k	97k	141k
Rev+Oie4	121k	181k	114k	173k

Table 7: Number of sentences and tuples extracted by Reverb, OpenIE4 and their combinations. The sentence is not included if it has no tuples extracted.
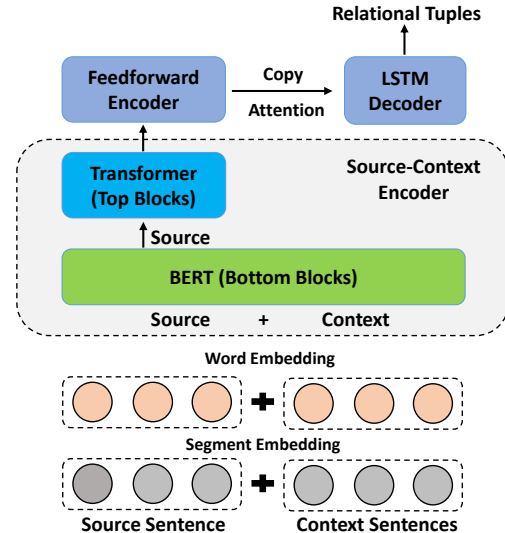

Figure 2: The architecture of DocIE.

5 DocIE Model

In this section, we present the proposed **Doc**ument-level context-aware **O**pen **I**nformation **E**xtraction model, named **DocIE**. As shown in Fig. 2, DocIE mainly consists of two parts: source-context encoder, and encoder-decoder.

Document-level Context Formally, we denote a document as $D = \{s_1, s_2, \ldots, s_N\}$ consisting of N sentences. The source sentence s_i is the input sentence that relational tuples are extracted from. Given source sentence s_i, we regard its surrounding sentences $c_i = \{s_{i-t}, \ldots, s_{i-1}, s_{i+1}, \ldots, s_{i+t}\}$ as contextual sentences, where t represents the context window size. The larger t is, the more document-level context c_i covers.

Source-Context Encoder The source-context encoder is inspired by a recent work (Ma et al., 2020) which adopts Flat-Transformer to incorporate context into source sentence, for machine translation. In DocIE, our encoder consists of (i) bottom blocks which take the concatenation of source sen-

tence and context sentences as input, and (ii) top blocks which take only the representation of the source sentence from the bottom blocks as input.

In our implementation, we use BERT (Devlin et al., 2019) as the bottom blocks to perform semantic interactions between source sentence s and context c. We first project both s and c into embedding space by summing their word embedding and segment embedding, *i.e.,* $e_s = E(s) + S(s)$ and $e_c = E(c) + S(c)$. Here, E is the trainable word embedding matrix, and S is the trainable segment embedding matrix. The segment embedding is to distinguish words in source sentence from words in context sentences. They are initialized to 0 and 1 for words in source and context sentences respectively. Then we concatenate e_s with e_c as $[e_s; e_c]$ as the input to the source-context encoder.

$$h_1[s; c] = \text{BERT}([e_s; e_c]) \qquad (1)$$

BERT, with multiple layers of transformers, merges source sentence information and its contextual information. We use the last hidden state $h_1[s; c]$ of BERT as the representation of the two concatenated input sequences.

On top of the BERT blocks, we add Transformer as top blocks (Vaswani et al., 2017) to prepare the source sentence representation for the following encoder-decoder. The source sentence representation is obtained by truncating the latter (context sentences representation) $h_1[c]$ from $h_1[s; c]$. Therefore only the former (source sentence representation) $h_1[s]$ is kept.

$$h_2[s] = \text{Transformer}(h_1[s]) \qquad (2)$$

Encoder-Decoder The encoder-decoder generation module follows CopyAttention (Cui et al., 2018) which casts OpenIE task as a sequence-to-sequence generation task with copying mechanism. The encoder-decoder framework represents a variable length input sequence in the encoder and uses it in the decoder to generate output sequence. In our encoder-decoder framework, attention mechanism (Bahdanau et al., 2015) is used to align the encoder hidden state with the decoder hidden state, jointly maximizing the log probability of output tuples, conditioned on the input sentence. Meanwhile, since tuple arguments and relation are normally sub-spans of the input sentence, additional copying mechanism (Gu et al., 2016) is applied. It helps copy words directly from the input sentence to the output tuples.

6 Experiments

We evaluate DocIE and compare its results with two baseline neural OpenIE models, CopyAttention+BERT and IMOJIE (Kolluru et al., 2020b). Kolluru et al. (2020b) report that CopyAttention+BERT is a strong baseline. Meanwhile, DocIE adopts CopyAttention (Cui et al., 2018) as its encoder-decoder module. Hence CopyAttention+BERT can be considered as the base model, from which DocIE adds context modelling.

6.1 Neural Baseline Models

We first evaluate the two neural baseline models trained with the pseudo labels listed in Table 7. The evaluation is conducted on the DocOIE evaluation dataset with CaRB scorer.

Reported in Table 8, CopyAttention+BERT outperforms IMOJIE in most settings by both measures: AUC and F1. In general, for both models, pseudo labels by Rev+Oie4 (and also Reverb) lead to better results in healthcare domain. Pseudo labels by Oie4+Rev (and also OpenIE4) generate better results in transportation domain. During our annotation of the 800 sentences, we observe that sentences in transportation domain tend to contain slightly more conjunctions (*e.g.,* multiple conjunctions in one sentence) and thus have more coordinating structures than those in healthcare. OpenIE4 system generally extracts more tuples than Reverb (refer to Table 7) and provides higher recall. Therefore, extractions in transportation domain with more conjunctions may better match the tuples extracted by OpenIE4.

Based on this set of results, in our following experiments, we use pseudo labels by Rev+Oie4 for healthcare domain, and pseudo labels by Oie4+Rev for transportation domain.

6.2 DocIE Against Baselines

In this section, we evaluate DocIE against sentence-level OpenIE systems. We refer DocIE without the top transformer layer as "DocIE w/o transformer" and DocIE as "DocIE w transformer" for clarity. The context window size of DocIE is set to 5 for healthcare domain and 4 for transportation domain.

Table 9 summarizes the experiment results. For easy comparison, the results of the best traditional OpenIE baselines (refer to Table 6) and neural OpenIE models (refer to Table 8) are replicated here. Observe that DocIE w transformer achieves the best AUC and F1 in both domains. Its variant, DocIE

Neural OpenIE	Pseudo labels	Healthcare				Transportation			
		AUC	Prec	Rec	F1	AUC	Prec	Rec	F1
CopyAttention+BERT	OpenIE4	38.6	54.4	**51.6**	52.9	38.5	54.3	**57.6**	55.9
	Oie4+Rev	40.4	57.1	50.4	53.5	38.3	55.3	56.9	**56.1**
	Reverb	43.7	77.8	46.4	58.1	36.9	70.5	42.2	52.8
	Rev+Oie4	**46.8**	**77.9**	48.6	**59.8**	**40.3**	**72.1**	43.9	54.6
IMOJIE	OpenIE4	36.2	73.0	**47.7**	57.7	35.7	62.9	48.8	55.0
	Oie4+Rev	34.1	69.5	46.7	55.9	**35.8**	63.5	**49.2**	**55.5**
	Reverb	38.5	79.2	45.6	57.9	33.2	77.3	39.2	52.0
	Rev+Oie4	**39.7**	**80.1**	46.4	**58.7**	33.0	**77.4**	39.6	52.4

Table 8: Neural baseline models trained with different pseudo labels. The best scores of *each model* are in boldface.

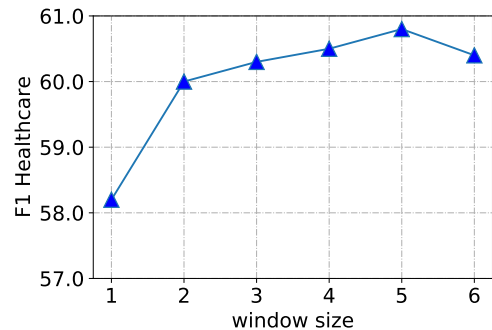| System	Healthcare				Transportation			
	AUC	Prec	Rec	F1	AUC	Prec	Rec	F1
Rev+Oie4	36.8	75.8	47.7	58.6	31.0	**74.2**	42.4	54.0
Oie4+Rev	35.8	59.6	**55.3**	57.4	30.1	53.4	52.7	53.0
CopyAttention+BERT	46.8	<u>77.9</u>	48.6	59.8	<u>38.3</u>	55.3	56.9	56.1
IMOJIE	39.7	**80.1**	46.4	58.7	35.8	<u>63.5</u>	49.2	55.5
DocIE w/o transformer	<u>47.1</u>	76.2	49.9	<u>60.3</u>	**38.5**	55.8	<u>57.0</u>	<u>56.4</u>
DocIE w transformer	**47.4**	74.4	<u>51.3</u>	**60.8**	**38.5**	56.0	**57.5**	**56.9**

Table 9: Results of DocIE and baselines. The best scores are in boldface and second best scores are underlined.

w/o top transformer, is the second best performer and outperforms all sentence-level models.

The experiment results suggest that incorporating document-level context is helpful in improving OpenIE. On the other hand, we remark that DocIE is trained by pseudo labels produced by traditional OpenIE models which do not consider document-level context. The potential of utilizing document-level context is yet to be fully realized.

6.3 Impact of Context Window Size

The setting of window size determines the number of context sentences to be considered. We evaluated the range from 1 to 6 and plot F1 scores against window size changes in Fig. 3. Observe that the optimal window size for healthcare domain is 5, and the number is 4 for transportation. Better F1 scores are observed along the increase of context sentence window, till 4 or 5. In general, 8~10 surrounding sentences (window size 4 or 5) provide sufficient context for sentence understanding. Small window size might not provide sufficient context, and a large window size might introduce noise and dominate source representation learning.

(a) Healthcare
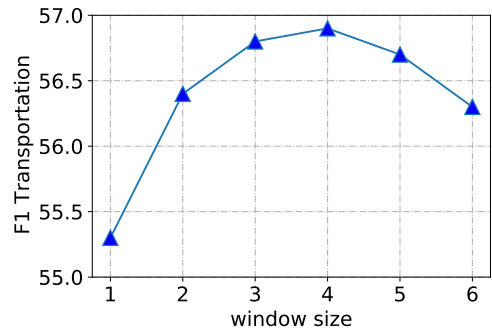
(b) Transportation

Figure 3: F1 with varying window sizes, on both domains.

6.4 Case Study

We use the two example sentences shown in Fig. 1 as a case study, to illustrate the differences between DocIE and the sentence-level neural OpenIE baselines: CopyAttention+BERT and IMOJIE.

For Sentence 1, CopyAttention+BERT incorrectly recognizes the word "transfers" as a verb, thus extracting an incorrect tuple (data ; transfers to ; a single target terminal). IMOJIE, however, completely misses the key phrase "data transfers" and extracts an incorrect tuple (a single target terminal ; using ; the invention). Only DocIE manages to extract the correct tuple (data transfers to a single target terminal ; using ; the invention).

For Sentence 2, there is no explicit clue about the relationship between "a device" and "a cellular base". Both CopyAttention+BERT and IMOJIE treat "a device a cellular base station" as a whole and mistakenly generate a tuple (Node-B ; can be ; a device a cellular base station having beam-forming antennas). In contrast, DocIE successfully splits "a device a cellular base station" by referring to surrounding context and extracts the correct tuple (Node-B ; can be ; a device). However, DocIE fails to infer the inter-relationship between "a device" and "a cellular base". Accordingly, another correct tuple (a device ; is such as ; a cellular base station) is not extracted.

Results of the two example sentences show the improvements made by DocIE after leveraging contextual information for tuple extraction.

6.5 Error Analysis

Similar to the error analysis performed in (Kolluru et al., 2020b), we examine tuples extracted by DocIE from 50 randomly selected sentences in DocOIE. We identify the following major error types. (i) **Incompleteness:** In 28% sentences, DocIE fails to cover at least one key phrase in either arguments or relation. Missing key phrases result in incomplete information extraction. (ii) **Incorrect Boundary:** 27% extractions misinterpret the syntactic meaning of the sentence, leading to incorrect boundary of arguments and relation. (iii) **Redundant Extractions:** 15% sentences contain redundant extractions; that is, the same relational fact is extracted multiple times from a sentence or phrase. (iv) **Grammatical Errors:** 13% extractions are not grammatically correct. Most grammatical errors are contributed by the incorrect verb form used in tuple relation.

6.6 Implementation

We implement DocIE using the AllenNLP framework[10] in Pytorch 1.4. Pre-trained BERT[11] is fine-tuned at learning rate 2×10^{-5} to get contextualized word embeddings. The learning rate for the other modules is set to 1×10^{-4}. The input dimension, projection dimension, feedforward hidden dimension, number of layers, and number of attention heads of top transformer encoder are set to 768, 256, 3072, 2, and 8, respectively. The hidden dimension, and word embedding dimension of the LSTM-decoder are set to 256 and 100 respectively.

7 Conclusion

In this paper, we propose to consider document-level contextual information for OpenIE task. We contribute DocOIE, the first document-level context-aware OpenIE dataset. It consists of 800 expert-annotated sentences from 80 documents. The documents are carefully selected and the annotations are completed by experts with high-level annotation consistency.

With the help of DocOIE, we conduct evaluation of neural OpenIE models and demonstrate that incorporating document-level context is helpful in improving OpenIE performance through DocIE. As a baseline for document-level context-aware OpenIE, DocIE achieves promising results compared with all sentence-level OpenIE models. Our future works are in two main directions. One is to research on more effective context-aware OpenIE models, and the other is to investigate the possibility of not relying on pseudo labels.

Acknowledgments

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages

---

[10]https://github.com/allenai/allennlp
[11]https://huggingface.co/transformers/model_doc/bert.html

344–354, Beijing, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hannah Bast and Elmar Haussmann. 2013. Open information extraction via contextual sentence decomposition. In *2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, September 16-18, 2013*, pages 154–159. IEEE Computer Society.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Nikita Bhutani, H. V. Jagadish, and Dragomir Radev. 2016. Nested propositions in open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Austin, Texas. Association for Computational Linguistics.

Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. Graphene: Semantically-linked propositions in open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2300–2311, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366. International World Wide Web Conferences Steering Committee / ACM.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. 2020. On aligning OpenIE extractions with knowledge bases: A case study. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 143–154, Online. Association for Computational Linguistics.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. OPIEC: an open information extraction corpus. In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Vancouver, Canada. 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)*, pages 3748–3761, Online. 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.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, Online. Association for Computational Linguistics.

José-Lázaro Martínez-Rodríguez, Ivan López-Arévalo, and Ana B. Ríos-Alvarado. 2018. Openie-based approach for knowledge graph construction from text. *Expert Syst. Appl.*, 113:339–355.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077. IJCAI/AAAI Press.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

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

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9523–9530. AAAI Press.

Hongyan Zhao, Ru Li, Xiaoli Li, and Hongye Tan. 2020. Cfsre: Context-aware based on frame-semantics for distantly supervised relation extraction. *Knowledge-Based Systems*, 210:106480.

DocOIE: A Document-level Context-Aware Dataset for OpenIE (Supplementary Material)

A DocOIE Annotation Agreement

This section complements Section 3.3 Annotation Consistency Measurement.

We take the sentence shown in Table 10 as an example, to demonstrate how the two experts clarified inconsistency and updated annotation policy. In blue-colored tuples, Expert A broke down a long argument *"nerve grafts in peripheral nerve tissue engineering"* into an additional tuple *(nerve grafts; are in; peripheral nerve tissue engineering)*, while Expert B chose to contain the descriptive part in this argument. After discussion, they decided to follow Expert B's practice to ensure complete argument without losing descriptive information. Meanwhile, as shown in red-colored tuples, the two experts disagreed on the inner relation of the argument "aft tissue-derived ECM modified tissue engineered nerve grafts". Expert A considered "aft tissue-derived ECM" should be modified by "tissue engineered nerve grafts" but Expert B thought "nerve grafts" shall be engineered by "aft tissue-derived ECM modified tissue". This inconsistency is resolved by referring to the relevant contexts in the document: "tissue engineered nerve grafts" is a terminological phrase and there exists an action of "ECM modification" performing on "tissue engineered nerve grafts". The annotation agreed by both experts are shown in the last row in Table 10.

B Extraction Results

Table 11 lists the tuples extracted by DocIE and the sentence-level neural OpenIE baselines: Copy-Attention+BERT and IMOJIE. This table complements Section 6.4 Case Study.

2388

Source Sentence	*The ANG meets the basic requirements for nerve grafts in peripheral nerve tissue engineering, and becomes aft tissue-derived ECM modified tissue engineered nerve grafts.*
Annotation by Expert A	(the ANG ; meets the basic requirements for ; nerve grafts) (nerve grafts ; are in ; peripheral nerve tissue engineering) (the ANG ; becomes ; tissue engineered nerve grafts) (tissue engineered nerve grafts ; is modified by ; aft tissue-derived ECM)
Annotation by Expert B	(the ANG ; meets the basic requirements for ; nerve grafts in peripheral nerve tissue engineering) (the ANG ; becomes ; aft tissue-derived ECM modified tissue engineered nerve grafts) (nerve grafts ; is engineered by ; aft tissue-derived ECM modified tissue)
Document Context	...another group is using plain *tissue engineered nerve grafts* (without *ECM modification*) for repairing the sciatic nerve defects of rats....
Annotation in Agreement	(the ANG ; meets the basic requirements for ; nerve grafts in peripheral nerve tissue engineering) (the ANG ; becomes ; aft tissue-derived ECM modified tissue engineered nerve grafts) (tissue engineered nerve grafts ; is modified by ; aft tissue-derived ECM)

Table 10: An example of inconsistent annotations between two annotators, and the annotation in agreement.

Sentence 1	*In some circumstances, data transfers to a single target terminal using the invention might not be significantly faster than conventional download methods.*
Annotated tuples	(data transfers to a single target terminal ; use ; the invention) (data transfers to a single target terminal using the invention ; might not be significantly faster than ; conventional download methods)
DocIE	(data transfers to a single target terminal ; using ; the invention) (data transfers to a single target terminal using the invention ; might not be ; significantly faster than conventional download methods)
CopyAttention+BERT	(data ; transfers to ; a single target terminal) (a single target terminal ; using ; the invention) (data transfers to a single target terminal ; might not be ; significantly faster than conventional download methods)

IMOJIE	(a single target terminal ; using ; the invention)
	(data transfers to a single target terminal using the invention ; might not be ;
	    significantly faster than conventional download methods)
	(data transfers to a single target terminal using the invention ;
	    might not be significantly faster than conventional download methods ;)
	(data transfers to a single target terminal using the invention ; might not be significantly faster ;)

Sentence 2	*Node-B can be a device a cellular base station having beam-forming antennas that serves various sectors of a cell.*
Annotated tuples	(Node-B ; can be ; a device) (a device ; is such as ; a cellular base station) (a cellular base station ; has ; beam-forming antennas) (Node-B ; serves ; various sectors of a cell)
DocIE	(Node-B ; can be ; a device) (a cellular base station ; having ; beam-forming antennas) (beam-forming antennas ; serves ; various sectors of a cell)
CopyAttention+BERT	(Node-B ; can be ; a device a cellular base station having beam-forming antennas) (a cellular base station ; having ; beam-forming antennas that serves various sectors of a cell) (beam-forming antennas ; serves ; various sectors of a cell)
IMOJIE	(Node-B ; can be ; a device a cellular base station having beam-forming antennas) (a cellular base station ; having ; beam-forming antennas) (beam-forming antennas ; serves ; various sectors of a cell)

Table 11: Tuples extracted respectively by DocIE, CopyAttention+BERT, and IMOJIE from the two example sentences in Fig. 1.