CycleOIE: A Low-Resource Training Framework For Open Information Extraction

Zhihong Jin[1], Chunhong Zhang[1], Zheng Hu[1]*, Jibin Yu[1],
Ruiqi Ma[1], Qingyun Chen[2], Xiaohao Liao[2], Yanxing Zhang [3],
[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
[2]Beijing Unimonet Co., Ltd.
[3]Hebei University of Architecture

Abstract

Open Information Extraction (OpenIE) aims to extract structured information in the form of triples from unstructured text, serving as a foundation for various downstream NLP tasks. Despite the success of neural OpenIE models, their dependence on large-scale annotated datasets poses a challenge, particularly in low-resource settings. In this paper, we introduce a novel approach to address the low-resource OpenIE task through two key innovations: (1) we improve the quality of training data by curating small-scale, high-quality datasets annotated by a large language model (GPT-3.5), leveraging both OpenIE principles and few-shot examples to form *LSOIE-g principles* and *LSOIE-g examples*; (2) we propose CycleOIE, a training framework that maximizes data efficiency through a cycle-consistency mechanism, enabling the model to learn effectively from minimal data. Experimental results show that CycleOIE, when trained on only 2k+ instances, achieves comparable results to previous models trained on over 90k instances. Our contributions are further validated through extensive experiments, demonstrating the superior performance of CycleOIE and our curated LSOIE-g datasets in low-resource OpenIE as well as revealing the internal mechanisms of CycleOIE. Code and data at https://github.com/Jinsns/CycleOIE.

1 Introduction

Open Information Extraction (OpenIE) is a crucial task in information extraction, focusing on extracting structured information, typically in the form of triples, from natural language sentences (Yates et al., 2007; Angeli et al., 2015). An extraction is usually made up of one or more triples and a triple consists of a head entity, a relation entity and a tail entity (subject, relation and object syntactically). This task is crucial for various downstream NLP applications, including summarization, knowledge graph construction, and question answering over knowledge bases.

Traditionally, OpenIE models are trained on large-scale datasets but tested on smaller, manually labeled datasets. While neural models have shown strong performance, they rely heavily on massive amounts of annotated data. However, manual annotation for such tasks is both time-consuming and costly. Publicly available large-scale OpenIE datasets, such as OIE2016(Stanovsky and Dagan, 2016), IMoJIE(Kolluru et al., 2020b) and LSOIE(Solawetz and Larson, 2021), are weakly labeled, relying on automated conversion from other NLP tasks or predictions from existing OpenIE models. These weak labels introduce noise and can limit the effectiveness of models trained on these datasets. The reliance on large-scale annotated data presents a critical bottleneck, especially in scenarios where high-quality labeled data is scarce or expensive.

Despite the growing demand, low-resource OpenIE has received limited attention in the literature. Most neural models (Kolluru et al., 2020a; Vasilkovsky et al., 2022; Wei et al., 2023; Kolluru et al., 2022) are trained on a large-scale training dataset. A recent effort by Fan and He (2023) introduces the OK-IE framework, which partially addresses this issue by reducing the training data required for the second stage of their model. However, the first stage still relies on large-scale data, and the two-stage process is computationally expensive in both time and memory during inference. In light of these limitations, we propose a new approach to solve OpenIE using a one-stage seq2seq generative model. This method directly generates the extraction from the input sentence, thereby cutting inference time and memory consumption in half compared to existing two-stage generation methods(Kolluru et al., 2022; Fan and He, 2023), while maintaining competitive performance in low-

---

*Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 3372–3390
January 19–24, 2025. ©2025 Association for Computational Linguistics

resource environments.

To train a one-stage generation OpenIE model in the low-resource environment, we address two key challenges: improving the quality of the training data and enhancing the data efficiency of the training process. First, we curate small-scale, high-quality training datasets by re-annotating a subset of sentences from an existing large-scale dataset using a large language model (LLM). Specifically, we sample sentences from LSOIE and generate extractions with GPT-3.5, guided by OpenIE annotation principles and a few-shot examples. This results in two curated training datasets: *LSOIE-g principles* with over 2k instances and *LSOIE-g examples* with over 4k instances. Despite their smaller size compared to existing datasets, these curated datasets demonstrate superior quality, as confirmed by our experimental results. Second, we improve training data efficiency by implementing a cycle training framework, which generates additional training data during the training process, thereby augmenting the initial-scale dataset. We adapt this framework for OpenIE using a pair of T5(Raffel et al., 2020; Chung et al., 2022) models and name this approach CycleOIE. As depicted in Figure 1, CycleOIE consists of two cycles: Sentence-Extraction-Sentence (SES) and Extraction-Sentence-Extraction (ESE). In each cycle, data passes through two models: a Sentence-to-Extraction(S2E) model and an Extraction-to-Sentence(E2S) model. The resulting OpenIE model for inference is the S2E model, with the same inference cost as a standard T5 model. By leveraging the cycle training framework, CycleOIE augments its training data at each iteration, enabling it to effectively learn from minimal initial training data. Remarkably, when trained on only 2k+ instances, CycleOIE achieves performance comparable to models trained on large-scale datasets with over 90k instances.

In summary, this paper aims to address the challenge of low-resource OpenIE. Our key contributions are as follows:

(1) We propose a novel, one-stage seq2seq generation approach to OpenIE, achieving competitive performance even in low-resource settings.

(2) We improve the annotation quality and curate two small-scale high-quality training datasets, LSOIE-g principles and LSOIE-g examples, by leveraging an LLM to annotate sentences sampled from existing dataset.

(3) We introduce CycleOIE, a cycle training framework that allows OpenIE models to learn efficiently from minimal data, achieving state-of-the-art performance compared to models trained on 40 times more data.

(4) We conduct extensive experiments that validate the superiority of both our curated LSOIE-g and the CycleOIE framework. Additionally, we provide a detailed analysis of CycleOIE's internal mechanisms, focusing on the role of intermediate outputs in enhancing model performance.

2 Related Work

2.1 Open Information Extraction

Open Information Extraction is a fundamental NLP task aiming at extracting structured information from a given sentence. Early OpenIE methods (Yates et al., 2007; Angeli et al., 2015; Del Corro and Gemulla, 2013; Gashteovski et al., 2017) rely on linguistic expertise for sentence parsing, while recent neural methods harness deep neural networks to represent the semantics with hidden states.

Neural methods in OpenIE can be broadly categorized into sequence labeling and sequence generation methods. Sequence Labeling methods (Ro et al., 2020; Kolluru et al., 2020a) usually use an encoder (e.g., BERT) to encode the given sentence into its embedding. A sequence labeling head connected to the encoder labels each token in the sequence. On the other hand, sequence generation methods (Cui et al., 2018; Kolluru et al., 2020b, 2022; Fan and He, 2023) reframe the input and output, transforming the information extraction task into the text generation task.

To our best knowledge, low-resource OpenIE has not drawn much attention from researchers. Fan and He (2023) present this new perspective of OpenIE and propose OK-IE framework, which is the only work we've found for low-resource OpenIE. OK-IE optimizes the training of the second stage of GEN2OIE(Kolluru et al., 2022), resulting in the convergence of the second-stage model with significantly less training data. However, we find that the performance of OK-IE is limited from further improvement given more training data. Also, their two-stage generation strategy theoretically doubles the consumption of inference resources which is believed to be more precious than the training resources in real applications. Starting from these considerations, we propose CycleOIE to extend low-resource OpenIE studies.

2.2 Cycle training

Cycle training, also known as cycle-consistency training, is a training framework that use non-parallel data to train a pair of inverse models (i.e., output and input of a model could become the input and output of another in reverse.). This idea is initially introduced in the machine translation task as the term iterative back-translation(Hoang et al., 2018) to solve the challenge of lack of sentence pairs composed of source language sentences and target language sentences.

Cycle training is widely used in text generation tasks. Hoang et al. (2018); Wei et al. (2020); Dou et al. (2020) manage to use this method to overcome the scarcity of paired sentence datasets in machine translation area. Iovine et al. (2022b) introduce CycleKQR, leveraging cycle training to enhance Question Answering performance by rewriting queries into appropriate forms while retaining the semantics. Wang et al. (2023) adopt this method to data-to-text generation task and evaluate the consistency of generation. Recently, cycle training has been applied to train information extraction models. Iovine et al. (2022a) apply cycle training to address the lack of in-domain annotation data, achieving competitive performance on Named Entity Recognition (NER) tasks. With NER and many other information extraction tasks, OpenIE share a similar challenge of insufficiency of in-domain annotated data. It highlights the potential of cycle training to handle low-resource OpenIE tasks.

3 Method

3.1 CycleOIE training framework

To address the challenge of training OpenIE models in low-resource environments, we propose CycleOIE. Figure 1 illustrates the one-epoch training process. As shown, our framework introduces a new format for extractions that differs from other seq2seq approaches in OpenIE (Kolluru et al., 2022; Fan and He, 2023). A detailed explanation of this reframing can be found in Appendix A.1.

Each training epoch consists of two cycles: the Sentence-Extraction-Sentence (SES) cycle and the Extraction-Sentence-Extraction (ESE) cycle. In the SES cycle, sentences from the training dataset are used, while in the ESE cycle, extractions are employed. Each cycle involves two models, with order reversed between cycles. The Sentence-to-Extraction (S2E) model predicts extractions from input sentences, while the Extraction-to-Sentence (E2S) model reconstructs sentences from input extractions. Importantly, in each cycle, only the second model is actively trained, while the first model remains frozen, providing intermediate outputs for the second model to learn from.

```
Algorithm: CycleOIE
Input: Dataset of sentences $D_S$ extractions $D_E$
Output: S2E model $M_{S2E}$ and E2S model $M_{E2S}$
  while $M_{S2E}$ and $M_{E2S}$ have not converged do
    for every batch $S$ in $D_S$:
      Transform $S$ into $E'$ using $M_{S2E}$
      Train $M_{E2S}$ with $(E', S)$
    end for
    for every batch $E$ in $D_E$:
      Transform $E$ into $S'$ using $M_{E2S}$
      Train $M_{S2E}$ with $(S', E)$
    end for
  end while
```

Table 1: Algorithm: CycleOIE

3.2 Training with cycle-consistency loss

In each epoch of CycleOIE, we first execute ESE cycle then SES cycle. Each cycle involves two models, with the first model generating an intermediate output that is then fed into the second model. The second model in each cycle is trained with cycle-consistency loss, targeted at reconstructing the original input given the intermediate output.

In the ESE cycle, the process starts with an extraction sequence $e = \{e_1, e_2, e_3, ..., e_n\}$ fed into the frozen E2S model, which predicts an intermediate sentence $\hat{s} = \{\hat{s}_1, \hat{s}_2, \hat{s}_3, ..., \hat{s}_n\}$. This sentence is then input to the S2E model, which generates a predicted extraction $\hat{e} = \{\hat{e}_1, \hat{e}_2, \hat{e}_3, ..., \hat{e}_n\}$ aiming to match the original extraction e. The cycle-consistency loss is computed between $\hat{e}$ and e.

$$\mathcal{L}_\theta(E, \hat{E}) = -\frac{1}{|E|} \sum_{e \in E} \frac{\sum_{i<|e|} p(e_i) log\, p(\hat{e}_i)}{|e|} \tag{1}$$

Here, E represents a batch of e and $|E|$ is the batch size. θ denotes the parameter of S2E model. $|e|$ represents the sequence length of the extraction. $p(.)$ signifies the token probability, e_i and $\hat{e}_i$ are the i-th tokens in e and $\hat{e}$ respectively.

In the SES cycle, the sentence sequence $s = \{s_1, s_2, s_3, ..., s_n\}$ is input into the frozen S2E model to predict an intermediate extraction $\hat{e} = \{\hat{e}_1, \hat{e}_2, \hat{e}_3, ..., \hat{e}_n\}$. This extraction is then fed into the E2S model, which predicts a sentence $\hat{s} = \{\hat{s}_1, \hat{s}_2, \hat{s}_3, ..., \hat{s}_n\}$ with the goal of reconstructing the original sentence s. The cycle-consistency loss is computed between $\hat{s}$ and the s.

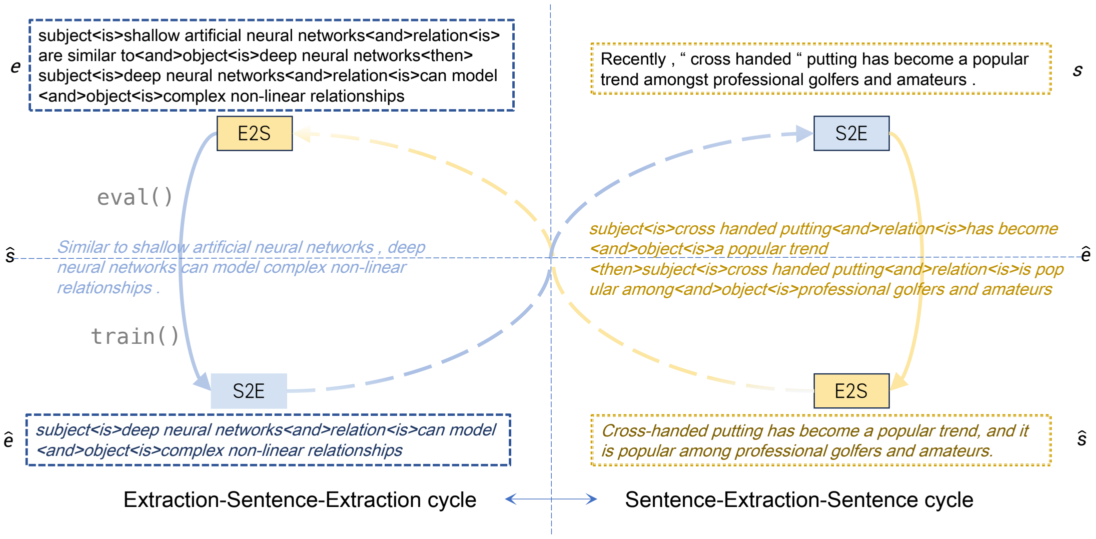
e
subject<is>shallow artificial neural networks<and>relation<is> are similar to<and>object<is>deep neural networks<then> subject<is>deep neural networks<and>relation<is>can model <and>object<is>complex non-linear relationships
E2S
eval()
ŝ
Similar to shallow artificial neural networks , deep neural networks can model complex non-linear relationships .
train()
S2E
ê
subject<is>deep neural networks<and>relation<is>can model <and>object<is>complex non-linear relationships
Extraction-Sentence-Extraction cycle
Recently , “ cross handed “ putting has become a popular trend amongst professional golfers and amateurs .
s
S2E
subject<is>cross handed putting<and>relation<is>has become <and>object<is>a popular trend <then>subject<is>cross handed putting<and>relation<is>is pop ular among<and>object<is>professional golfers and amateurs
ê
E2S
Cross-handed putting has become a popular trend, and it is popular among professional golfers and amateurs.
ŝ
Sentence-Extraction-Sentence cycle

Figure 1: CycleOIE training framework. The left half side illustrates the ESE cycle while the right half side illustrates the SES cycle. For the bottom half, we mark *train()* to indicate the second model in each cycle is trained according to the cycle-consistency loss.

	Dataset	Sentences	Triples
training sets	IMoJIE	91725	190661
	LSOIE	34780	100862
	LSOIE-g$_{principles}$	2147	6820
	LSOIE-g$_{examples}$	4901	26958
test sets	CaRB	641	2715
	BenchIE	300	783

Table 2: Statistics of datasets used in experiments.

$$\mathcal{L}_\phi(S, \hat{S}) = -\frac{1}{|S|} \sum_{s \in S} \frac{\sum_{i<|s|} p(s_i) log\, p(\hat{s}_i)}{|s|} \quad (2)$$

Here S represents a batch of s and $|S|$ is the batch size. $|s|$ represents the length of text sequence of the given sentence. ϕ denotes the parameter of the E2S model, s_i and $\hat{s}_i$ are the i-th tokens in s and $\hat{s}$ respectively.

After completing both the ESE cycle and SES cycle, one epoch of CycleOIE training is completed. The pseudocode for CycleOIE training on batched data is presented in Table 1.

3.3 Instruct LLM for training data

To address the challenges of low-resource Open Information Extraction (OpenIE), we developed a novel approach to curating a small-scale, high-quality training dataset. This method leverages Large Language Models (LLMs) as intelligent annotators, offering a scalable and effective alternative to traditional automated extraction methods. Our approach can be broadly applied to enhance the quality of training data in various NLP tasks, particularly where human-like precision is desired but large-scale human annotation is infeasible.

We base our work on the recently introduced LSOIE dataset (Solawetz and Larson, 2021), which is a large-scale OpenIE dataset derived from other NLP datasets through automated scripts. While LSOIE covers a wide range of sentences, the automated nature of its tuple extraction process introduces limitations, particularly in accurately parsing the nuances of complex sentence structures. This results in lower-quality extractions that fail to capture the full richness of the data.

Recognizing this limitation, we employ LLMs to generate extractions that more closely mimic the depth and accuracy of human annotations. Specifically, we sample a subset of sentences from LSOIE and use GPT-3.5 to act as a sophisticated annotator. By crafting targeted prompts, we instruct GPT-3.5 to produce extractions that reflect a more nuanced understanding of sentence structure, effectively bridging the gap between automated and human-like annotation.

To further optimize this process, we experimented with different instructional strategies and ultimately created two distinct datasets. The first, **LSOIE-g principles**, is generated by embedding the principles of OpenIE annotation directly into the prompt. The second, **LSOIE-g examples**, takes a few-shot learning approach, incorporating multiple examples of sentence-extraction pair into a multi-turn dialogue format to guide GPT-

3.5's extraction process. Details of the instructional prompts used are provided in Appendix A.2. Dataset statistics can be found in Table 2. In the context of training our CycleOIE model, the sentence set and extraction set serve two phases of cycle training, SES and ESE independently. Combined with our high-quality annotated datasets, LSOIE-g$_{principles}$ and LSOIE-g$_{examples}$, provides a unique resource that enhances the training process by effectively simulating human-level annotation.

This innovative use of LLMs for data annotation not only improves the quality of the training data but also demonstrates a generalizable strategy for generating high-quality datasets in low-resource scenarios. By leveraging powerful language models in a structured, instructional manner, this approach can be applied to a wide range of NLP tasks where high-quality annotations are critical for performance but difficult to obtain at scale.

4 Experimental Setup

4.1 Datasets

Training sets. (1)LSOIE(Solawetz and Larson, 2021) is a large-scale OpenIE dataset converted from QA-SRL 2.0 in a similar conversion method with OIE2016(Stanovsky and Dagan, 2016). (2)IMoJIE(Kolluru et al., 2020b) constructs a high quality OpenIE dataset with Wikipedia sentences and extractions predicted by former OpenIE systems including OpenIE4(Christensen et al., 2011), ClausIE(Del Corro and Gemulla, 2013), and RNNOIE(Stanovsky et al., 2018).

Test sets. (1)CaRB(Bhardwaj et al., 2019) test set contains 641 sentences. Obtained through crowdsourcing, CaRB's annotations are generally considered as an noise reduction from OIE2016(Stanovsky and Dagan, 2016). Following Kolluru et al. (2020a); Vasilkovsky et al. (2022); Wei et al. (2023), when evaluating on CaRB test set, in addition to its own CaRB scoring function, we also use CaRB(1-1)Kolluru et al. (2020a), OIE16(Stanovsky and Dagan, 2016), Wire57(Lechelle et al., 2019) scoring function to perform a comprehensive evaluation. We also introduce a new metric, Training Data Efficiency (TDE). $TDE = 10^4 \times F1 \,/\, training\ data\ size$, where training data size is equal to the number of sentences in the training dataset. A higher TDE stands for a training data instance contributes more F1 score in average, better evaluates the method's efficiency of utilizing the training data.

(2)BenchIE(Gashteovski et al., 2022) has high quality annotations extracted by experts as well as a too strict scoring function that BenchIE judges an predicted triple to be true only when it exactly matches a gold triple in the cluster of the fact. We suspect it is the reason for few OpenIE study conduct evaluation on it. To take advantage of its high-quality annotations, we use CaRB scoring function to evaluate on BenchIE test set.

4.2 Baseline methods

Our baselines include (1)MINIE(Gashteovski et al., 2017), (2)ClausIE(Del Corro and Gemulla, 2013), (3)OpenIE4(Christensen et al., 2011), (4)OpenIE5(Saha et al., 2017; Saha and Mausam, 2018), (5)IMOJIE(Kolluru et al., 2020b), (6)IGL-OIE(Kolluru et al., 2020a), (7)CIGL-OIE(Kolluru et al., 2020a), (8)OpenIE6(Kolluru et al., 2020a), (9)DetIE(Vasilkovsky et al., 2022), (10)IOT(Wei et al., 2023), (11)GEN2OIE(Kolluru et al., 2022), (12)OK-IE(Fan and He, 2023)[1]. (1)~(4) are non-neural methods, while (5)~(12) are neural methods.

4.3 Details of implementing CycleOIE

CycleOIE utilizes two T5 models for the Sentence-to-Extraction (S2E) and Extraction-to-Sentence (E2S) tasks, both initialized with flan-t5-base[2](Chung et al., 2022). We extend the tokenizers of both models by adding three additional tokens. The models are trained with a learning rate of 2×10^{-5} and a batch size of 8. For intermediate output generation, such as the extraction in the SES cycle, we employ beam search with 4 beams as the default decoding strategy. Further analysis on decoding strategies is discussed in Section 5.4. The models are trained for 100 epochs, with evaluations conducted every 5 epochs. The checkpoint with the highest CaRB F1 score on the CaRB development set is selected as the final model for CycleOIE.

5 Experimental Results and Analysis

5.1 Main results

We compare our CycleOIE with baselines. As shown in Table 3, on CaRB test dataset, our unsupervised method, CycleOIE(examples), achieves the second highest AUC on CaRB, the highest F1 and AUC on CaRB(1-1), the highest F1 and AUC on OIE16-C, while CycleOIE(principles) also achieves a competitive performance compared to

---

[1] OK-IE does not provide open-source code.

[2] https://huggingface.co/google/flan-t5-base

Training dataset & size	model	CaRB			CaRB(1-1)			OIE16-C			Wire57-C	
		F1	AUC	TDE	F1	AUC	TDE	F1	AUC	TDE	F1	TDE
non-neural methods	MINIE	41.9	-	-	38.4	-	-	52.3	-	-	28.5	-
	ClausIE	45.0	22.0	-	40.2	17.7	-	61.0	38.0	-	33.2	-
	OpenIE4	51.6	29.5	-	40.5	20.1	-	54.3	37.1	-	34.4	-
	OpenIE5	48.0	25.0	-	42.7	20.6	-	59.9	39.9	-	35.4	-
IMOJIE 91725	IMoJIE	53.5	33.3	5.8	41.4	22.2	4.5	56.8	39.6	6.2	36.0	3.9
	IGL-OIE	52.4	33.7	5.7	41.1	22.9	4.5	55.0	36.0	6.0	34.9	3.8
	CIGL-OIE	54.0	35.7	5.9	42.8	24.6	4.7	59.2	40.0	6.5	36.8	4.0
	OpenIE6	52.7	33.7	5.7	46.4	26.8	5.1	65.6	48.4	7.2	**40.0**	4.4
	DetIE	52.1	36.7	5.7	40.1	24.0	4.4	56.0	38.7	6.1	36.0	3.9
	IOT	52.7	37.0	5.7	40.1	24.0	4.4	55.9	38.5	6.1	36.1	3.9
	Gen2OIE	**54.2**	33.0	5.9	43.9	23.6	4.8	60.5	40.0	6.6	37.1	4.0
	OK-IE	53.2	-	5.8	-	-	-	-	-	-	-	-
LSOIE 34781	DetIE	43.0	27.2	12.3	33.1	18.3	9.5	49.7	32.7	14.3	31.2	9.0
	IOT	42.5	27.1	12.2	32.2	17.7	9.3	53.3	36.0	15.3	30.0	8.6
LSOIE 5000	CycleOIE	40.7	25.1	81.4	33.6	18.8	67.2	51.0	33.9	102	28.8	57.6
examples 4901	CycleOIE	51.2	**39.0**	104.5	**47.4**	**33.6**	97.3	**68.7**	**57.8**	140.2	38.6	78.8
LSOIE 2266	CycleOIE	39.1	23.6	172.6	31.9	17.4	140.8	50.1	32.9	221.1	27.2	120.0
principles 2147	CycleOIE	53.1	36.3	**247.3**	43.9	27.2	**204.5**	57.3	40.0	**266.9**	39.8	**185.4**

Table 3: Overall evaluation on CaRB test. The best scores in each column are shown in bold, and second bests are underlined.

methods	P	R	F1	TDE
Multi2OIE	59.8	61.3	60.5	0.6
OpenIE6(IMOJIE)	47.8	67.1	55.9	6.1
DetIE(LSOIE)	50.4	46.3	48.3	13.9
DetIE(IMOJIE)	54.9	**64.1**	59.1	6.4
CycleOIE(principles)	**60.3**	61.3	**60.8**	**283.2**
CycleOIE(examples)	48.5	63.9	55.2	112.6

Table 4: evaluation on BenchIE$_{test}$

other baselines. As shown in Table 4, on BenchIE test dataset, CycleOIE(principles) achieves the highest F1 score and CycleOIE(examples) achieves a comparable performance. It is worth noting that in terms of the TDE metric, we are significantly ahead of the competition. That's because CycleOIE(principles) is trained merely on 2147 instances and CycleOIE(examples) is trained on 4902 instances; while recent neural methods are trained mostly on IMoJIE which has 91725 instances or LSOIE which has 34780 instances.

We notice that CycleOIE(examples) excels on CaRB(1-1) and OIE16-C while CycleOIE(principles) performs better on CaRB and Wire57. It's because CaRB and Wire57 give stronger penalty to recall. Given a sentence, GPT, instructed by few-shot examples, tends to generate more tuples per sentence. Trained on these data, CycleOIE(examples) has advantage on recall.

methods	0.9% IMOJIE	5% IMOJIE	20% IMOJIE
IMOJIE	-	36.9	39.5
OpenIE6	42.9	43.1	45.3
OK-IE	**52.9**	**53.1**	**53.3**
CycleOIE	34.2	49.8	50.3

Table 5: CaRB F1 scores of low-resource settings. With 0.9% IMOJIE training data, IMOJIE is incapable of generating effective triples. For OK-IE, the column names represent the data sizes for training **its second stage only**. Best score in the column are shown in bold, and second bests are in underlined.

5.2 Low-resource results

For a relatively fair comparison to other low-resource settings, which are trained on IMOJIE, we use IMOJIE training data to cycle train a pair of T5. Table 5 demonstrates the CaRB F1 scores of the low-resource settings tested on CaRB test set. IMOJIE method cannot generate effective extractions with 0.9% IMOJIE training data. IMOJIE and OpenIE6 keep relatively low F1 scores but effective improvements with the increment of training data size. OK-IE keeps the highest F1 scores, but has no significant improvement with the increment of training data size. Please note that for the OK-IE model, Table 5 only represents the data volume used by its second-stage model. The first-stage model is inherited from GEN2OIE and is trained

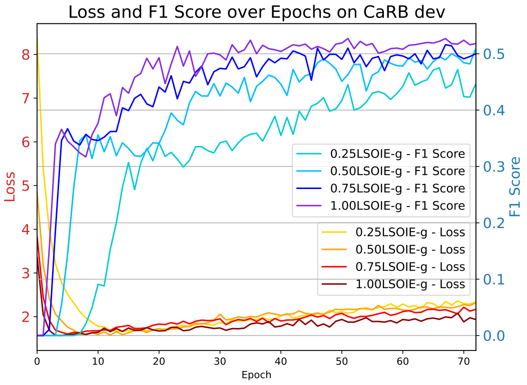

Figure 2: Loss and F1 score of the S2E model with the growth of epochs.

on the full IMOJIE dataset. Therefore, its low-resource training pertains only to the second stage. This data suggests that its impressive performance may rely more heavily on the capabilities of the first-stage model. Our CycleOIE, which has only one-stage generation, are trained on totally low-resource training data. It gets the lowest F1 score when trained on 0.9% IMOJIE data, but its performance increases significantly when the training data size increases to 5%. Then it gets almost convergence at this data size. With 20% of IMOJIE data, which contains 18345 instances, CycleOIE's performance still has some distance with the performance demonstrated in Table 3. This may indicates our curated dataset *LSOIE-g principles* and *LSOIE-g examples* can train models more efficiently, at least for our CycleOIE.

We further examine the impact of training on varying sizes of the *LSOIE-g principles*. The training trends are depicted in Figure 2 which tracks the S2E model's loss θ and the CaRB F1 score on the CaRB dev set across each epoch of CycleOIE. Notably, training with 100% LSOIE-g principles leads to the fastest decline in loss, which then plateaus at its lowest level, while the F1 score remains elevated, starting from approximately the 20th epoch. Additionally, the F1 score's fluctuations diminish as CycleOIE is trained on larger datasets. Interestingly, even with just 0.25 *LSOIE-g principles*–comprising 536 sentences–CycleOIE surpasses a 40.0 F1 score, outperforming CycleOIE trained on 0.9% IMOJIE, which includes 825 sentences. These results highlight the distinct advantages and effectiveness of our meticulously curated dataset.

	CaRB		
	P	R	F1
baseline 1: sft on LSOIE	0.573	0.339	0.426
baseline 1 → sft on LSOIE-g	0.654	0.418	0.51
baseline 1 → CycleOIE	0.694	0.424	**0.526**
baseline 2: sft on LSOIE+IMoJIE	0.402	**0.453**	0.426
baseline 2 → sft on LSOIE-g	**0.704**	0.415	0.523
baseline 2 → CycleOIE	0.694	0.428	**0.530**

Table 6: Cycle training versus sft. When we conduct sft on LSOIE-g, the sentence and the corresponding extraction are paired up to make an parallel dataset. In this table, "+" expresses that we add up data of two existing datasets to construct the training dataset for this test. "→" expresses that we conduct a two-stage training.

5.3 Comparing cycle training to sft

CycleOIE benefits from both the training data and the training framework. We have previously discussed the efficacy of our curated LSOIE-g dataset. Here, we delve into the advantages of cycle training over supervised fine-tuning (SFT) in the context of training generative models for Open Information Extraction (OpenIE).

As illustrated in Table 6, we initially established a baseline model via supervised fine-tuning (SFT) on LSOIE. Subsequently, we pursued two distinct training approaches with this baseline. The first approach involved further SFT on LSOIE-g, while the second approach entailed implementing cycle training on LSOIE-g. To establish a more robust baseline, we also enriched the training data by combining LSOIE with IMOJIE. Although the F1 score remained unchanged, the precision and recall metrics became more aligned. We conducted identical training experiments on this enhanced baseline. The experimental outcomes reveal that training with LSOIE-g yields substantial enhancements to the baseline model, with cycle training further augmenting performance.

5.4 Analysing the intermediate generation

As highlighted in related literature, cycle training has been assessed across some tasks for its performance. However, limited research has explored the underlying mechanisms to elucidate its data efficiency. In supervised fine-tuning of a seq2seq model, a sentence and its corresponding annotated extraction are treated as a pair of supervisory data to fine-tune the model. While cycle training pairs supervisory data with the intermediate generation and the input data. As optimization progresses, the

	mode	consistency	variation
num_beams=4	10	11	9
top_p=0.3	13	13	5
top_p=0.5	12	12	5
top_p=0.7	4	4	11
top_p=0.9	3	2	14

Table 7: Characteristics data of the case. *Mode* stands for the most frequent generation. *Consistency* stands for the intermediate sentence is consistent with the input extraction. *Variation* stands for the number of different intermediate sentences.

	CaRB		
	P	R	F1
num_beams=4	61.6	41.2	49.4
top_p=0.3	64.5	**41.6**	50.6
top_p=0.5	**65.3**	**41.6**	**50.8**
top_p=0.7	57.5	41.0	47.9
top_p=0.9	49.0	37.8	42.7

Table 8: The CycleOIE performance trained on 50%LSOIE-g principles, with different top_p value when generating the intermediate output. We also demonstrate the performance of our default setting, beam search(num_beams=4), to be a baseline to compare with.

intermediate generations may diverge from those of the preceding epoch, suggesting that cycle training acts as a form of data augmentation strategy within the training process. We posit that the presence of more varied intermediate generations, which effectively equates to an increase in the volume of training data, leads to enhanced performance.

We use neuclus (top_p) sampling(Holtzman et al., 2020), which is widely used as the decoding strategy for generative models, to control the diversity of intermediate generations, resulting in varied outcomes. Given our focus on low-resource environments, we conduct this experiment using 50%*LSOIE-g principles*. We maintain the same top_p value for the both the S2E and E2S model during the generation of intermediate outputs. Table 8 indicates that performance is higher with top_p values of 0.3 or 0.5 compared to our default setting, which uses beam search (num_beams=4) as the decoding strategy. However, when higher top_p values are used, performance declines. This is further illustrated in Figure 3, where higher top_p values correspond with lower loss but also lower F1 scores and increased F1 score fluctuations.

We manually count the characteristic data of the intermediate sentences generated by the E2S model in ESE cycle and present the findings in Table 7. This demonstrates that as the top_p value increases, the number of variations increases, indicating greater diversity in intermediate sentences. This aligns with our hypothesis. However, as sentences become more diverse, their consistency with the input extraction is not guaranteed by the E2S model, which can negatively impact training. The detail of the case is provided in Appendix A.3.

While higher top_p values ranging from 0.70 to 0.95 have been shown to generate text that is more human-like (Holtzman et al., 2020) and enhance the text quality (Cui et al., 2021), it is recommended to use a lower range of top_p values, between 0.3 and 0.5, for generating consistent intermediate outputs in CycleOIE.

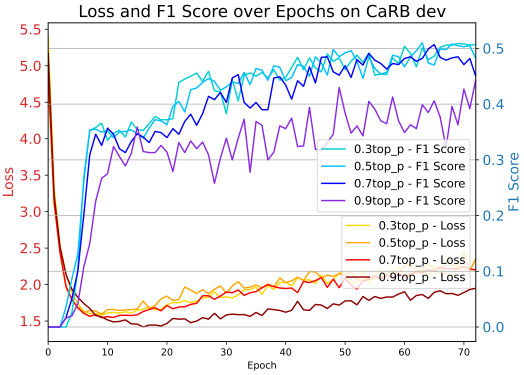

Figure 3: Loss and F1 score of the S2E model with the growth of epochs, evaluated on CaRB dev, different top-k when generating middle output.

6 Conclusion

In this paper, we present solutions to the low-resource OpenIE challenge from two perspective. For higher training data quality, we curate high-quality training datasets, LSOIE-g, using GPT-3.5, demonstrating the effectiveness of leveraging large language models for data annotation. For higher data efficiency, We introduce CycleOIE, a training framework that uses cycle-consistency to efficiently learn from limited data. Despite training on just 2k+ instances, CycleOIE achieved performance comparable to models trained on 90k+ datasets. Our approach highlights the potential of combining high-quality data and efficient training to solve OpenIE in low-resource environments.

Limitations

While CycleOIE demonstrates strong performance in low-resource settings, there are areas where further optimization could be pursued. One limitation is the relatively longer training time compared to simpler fine-tuning approaches, as observed in our experiments (Figure 2 and 3). Although the models converge by the 30th epoch, this process could be more efficient.

For future work, we see potential in exploring ways to streamline the training process, particularly by leveraging advances in large language models. Integrating these models more deeply into the workflow could further enhance the performance and efficiency of our approach, while maintaining its strengths in low-resource environments.

Acknowledgments

We would like to express our gratitude to the anonymous reviewers and the area chair for their valuable feedback. This work was supported by the Beijing Association of Higher Education Project MS2023151, Project 5203204043 and Project A2024091.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Sangnie Bhardwaj, Samarth Aggarwal, and 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.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the Sixth International Conference on Knowledge Capture*, K-CAP '11, page 113–120, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Danxin Cui, Dominique Mariko, Estelle Labidurie, Hugues de Mazancourt, and Patrick Paroubek. 2021. A sequence to sequence transformer data logic experiment. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 9–18, Lancaster, United Kingdom. Association for Computational Linguistics.

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.

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

Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. Dynamic data selection and weighting for iterative back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5894–5904, Online. Association for Computational Linguistics.

Zhiyuan Fan and Shizhu He. 2023. Efficient data learning for open information extraction with pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13056–13063, Singapore. 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, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. 2022. BenchIE: A framework for multi-faceted fact-based open information extraction evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4472–4490, Dublin, Ireland. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Andrea Iovine, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022a. Cyclener: An unsupervised training approach for named entity recognition. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2916–2924, New York, NY, USA. Association for Computing Machinery.

Andrea Iovine, Anjie Fang, Besnik Fetahu, Jie Zhao, Oleg Rokhlenko, and Shervin Malmasi. 2022b. CycleKQR: Unsupervised bidirectional keyword-question rewriting. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11875–11886, Abu Dhabi, United Arab Emirates. 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.

Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, and Mausam. 2022. Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multiˆ2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117, Online. Association for Computational Linguistics.

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

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

Jacob Solawetz and Stefan Larson. 2021. LSOIE: A large-scale dataset for supervised open information extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2595–2600, Online. Association for Computational Linguistics.

Gabriel Stanovsky 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.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salikhov, Mikhail Stepnov, Andrey Chertok, and Sergey Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11412–11420.

Zhuoer Wang, Marcus Collins, Nikhita Vedula, Simone Filice, Shervin Malmasi, and Oleg Rokhlenko. 2023. Faithful low-resource data-to-text generation through cycle training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2847–2867, Toronto, Canada. Association for Computational Linguistics.

Hao-Ran Wei, Zhirui Zhang, Boxing Chen, and Weihua Luo. 2020. Iterative domain-repaired back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5884–5893, Online. Association for Computational Linguistics.

Kaiwen Wei, Yiran Yang, Li Jin, Xian Sun, Zequn Zhang, Jingyuan Zhang, Xiao Li, Linhao Zhang, Jintao Liu, and Guo Zhi. 2023. Guide the many-to-one assignment: Open information extraction via IoU-aware optimal transport. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4971–4984, Toronto, Canada. Association for Computational Linguistics.

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.

A Appendix

A.1 Task reframing

As shown in Table 9, we seek to reframe OpenIE task as a seq2seq problem with least additional tokens. Previous OpenIE tasks like GEN2OIE(Kolluru et al., 2022) and OK-IE(Fan and He, 2023) use additional tokens to format the output sequences. For example, <*s*> and <*/s*> are used to bracket the subject entity, <*r*> and <*/r*> are used to bracket the relation entity, <*o*> and <*/o*> are used to bracket the object entity. In this way, when N entity types are required to be extracted, 2N additional tokens are needed to format the sequence. In our designed format, we use natural language tokens whose representation is learned in the pre-training stage, to help construct the format, reducing the addition tokens in need. With only three additional tokens, our designed format can format extractions with N entity types no matter with the value of N. Formally, the token <*is*> is always following the token *subject*, or *relation*, or *object*, conditioning the part of speech to be generated. <and> is used to split entities in a triple while <then> is used to separate triples from an extraction.

sentence	Earlier today , Thailand 's Prime Minister Yingluck Shinawatra formally dissolved the country 's parliament and called for new elections .
extraction (our format)	subject <is> the country 's parliament <and> relation <is> dissolved <and> object <is> Earlier today Thailand 's Prime Minister Yingluck Shinawatra <then> subject <is> Thailand 's Prime Minister Yingluck Shinawatra <and> relation <is> called <and> object <is> Earlier today new elections
extraction (format of GEN2OIE and OK-IE)	<s> the country 's parliament </s> <r> dissolved </r> <o> Earlier today Thailand 's Prime Minister Yingluck Shinawatra </o> <s> Thailand 's Prime Minister Yingluck Shinawatra </s> <r> called </r> <o> Earlier today new elections </o>

Table 9: <*is*>, <*and*>, <*then*> is added to the tokenizer to format the extractions.

A.2 Details of constructing LSOIE-g

Instruction demonstrated in Table 10 is used to generate *LSOIE-g principles* extractions. We give 6 annotation rules and each rule is provided with an example. Intructions demonstrated in Table 11 is used to generate *LSOIE-g examples* extractions. We provide 5-shot examples, structured for 5-turn chat, and input the sentence to annotate in the user's input in the 6th turn.

Ragarding the volumn of our low-resource training dataset, we believe that a quantity equal or double that of the test set is appropriate. The LSOIE$_{wiki}$ dev and test set have 2266 and 2402 sentences respectively. We directly use LSOIE$_{wiki}$ dev sentences to curate LSOIE-g principles, since we have opted not to evaluate on the LSOIE dev or test set due to the belief that the annotation quality, being automatically derived, is not as robust as that of human-annotated datasets. After filtering some invalid generation, there remains 2147 effective sentence-extraction pair. We attempted to construct additional training data by sampling sentences from the LSOIE$_{wiki}$ train, yet observed no significant improvement in performance. Consequently, we have set the size of our training data at 2,147 instances.

For LSOIE-g examples, we sample 5000 sentences from LSOIE$_{wiki}$ train to annotate with GPT-3.5. After filtering, there remains 4901 effective instances. We sampled a proportion from these instances to train CycleOIE. The performance keeps improving with the increment of training data size. So we have set the size of LSOIE-g examples at 4901 instances.

A.3 Case study on intermediate generations

In Table 12, 13, 14, 15, 16, we demonstrate the intermediate sentences generated by the E2S model in ESE cycle with different decoding strategy. The extraction input to the E2S model, which consisting 3 triples, is: *(A massive fire || is feared to have killed || residents) ; (A massive fire || is feared to have destroyed || more than one hundred homes); (A massive fire || is feared to have occurred in || the Blue Mountains region of New South Wales, Australia)*

We observe that with higher top_p value, the intermediate outputs become more diversified. However, this change may be harmful because it will generate sentences not consistent with the input. For example, comparing Table 13 and 16, we ob-

System	You are an expert annotator in Information Extraction.
User	Following the annotation principles, extract information from the given sentence below. Annotation principles: (1) Each triple should represent two entities or concepts, and the verb-mediated relation between them. For example, from the input sentence "Michael Jordan, who is a former basketball player, was born in Brooklyn.", there are three entities and concepts— Michael Jordan, former basketball player and Brooklyn—which are related as follows: {"subject": "Michael Jordan", "relation": "is", "object": "former basketball player"} and {"subject": "Michael Jordan", "relation": "was born in", "object": "Brooklyn"}. (2) When possible, if an extraction is in passive voice, the annotator should place its active voice equivalent into the appropriate fact synset. For instance, consider the sentence "The ball was kicked by John."; then, the fact synset should contain the following triples: {"subject": "[The] ball", "relation": "was kicked by", "object": "John"} {"subject": "John", "relation": "kicked", "object": "[The] ball"} (3) Extractions that indicate attribution of another core piece of information should be placed in separate fact synset, because they indicate a separate piece of information with separate predicate. For example, the core information of the sentence "Conspiracy theorists say that Barack Obama was born in Kenya." is that Barack Obama was born in Kenya. the triple {"subject": "Barack Obama", "relation": "[was] born in", "object": "Kenya"}—in one fact synset, and the triples indicating attribution— {"subject": "Conspiracy theorists", "relation": "say that", "object": "Barack Obama was born in Kenya"}—in another. (4) The annotator should not extract incomplete clauses, i.e., triples that lack crucial piece of information. Suppose there is the input sentence "He was honored by the river being named after him". The following triple should not be manually extracted: ("He"; "was honored by"; "[the] river"), but the following triples should be: {"subject": "He", "relation": "was honored by [the] river being named after", "object": "him"} and {"subject":"[the] river", "relation": "being named after", "object": "him"}. (5) The annotator should not allow for conjunctive phrases to form an argument (i.e., subject or object). Such arguments should be placed into separate extractions (and in separate fact synsets). Consider the sentence "Michael Jordan and Scottie Pippen played for Chicago Bulls.". The annotator should manually extract the following triples: {"subject":"M. Jordan", "relation": "played for", "object": "Chicago Bulls"} {"subject":"S. Pippen", "relation": "played for", "object": "Chicago Bulls"} The annotator should not, however, extract {"subject": "M. J. and S. P.", "relation": "played for", "object": "Chicago Bulls"}. (6)We focus on explicit extractions, which means that every word in the extracted triple must be present in the original input sentence. Therefore, implicit extractions—i.e., extractions that contain inferred information with words not found in the sentence—are not considered. One example implicit extraction is {"subject": "Michael Jordan", "relation": "be", "object": "Prof."} from the input sentence "Prof. Michael Jordan lives in USA.", where the triple infers that Michael Jordan is professor without being explicitly indicated in the sentence (i.e., the word "be" is not present in the input sentence, it is inferred) Given sentence: "*{INPUT SENTENCE}*" Don't miss any fact. Format these facts into compact JSON format {"subject": "", "relation": "", "object": ""}. 1 compact JSON in 1 line. Don't generate any token outside the most outer bracket{ } of JSON.

Table 10: Instruct gpt to annotate LSOIE-g sentences with OpenIE annotation principles. *{INPUT SENTENCE}* should be replaced with the sentence to annotate.

System	You are an expert in Information Extraction. Please extract triples from the given sentence. Please write optional tokens in square brackets [and] . Link subject, predicate, and object in a triple with –> . Please write one triple in one line.
User	He served as the first Prime Minister of Australia and became a founding justice of the High Court of Australia .
Assistant	He –> served as –> [the] [first] Prime Minister [of Australia] He –> served as [the] [first] Prime Minister of –> Australia He –> became –> [a] [founding] justice of [the] High Court [of Australia] He –> became [a] [founding] justice of –> [the] High Court [of Australia] He –> became [a] [founding] justice of [the] High Court of –> Australia
User	Graner handcuffed him to the bars of a cell window and left him there , feet dangling off the floor , for nearly five hours .
Assistant	Graner –> handcuffed –> him Graner –> left him [there] for –> [nearly] five hours Graner –> handcuffed him to –> [the] bars [of a cell window] feet –> dangling off –> [the] floor Graner –> handcuffed him to [the] bars of –> [a] [cell] window Graner –> left –> him feet –> dangling off [the] floor for –> nearly five hours
User	It deals with cases of fraud in relation to direct taxes and indirect taxes , tax credits , drug smuggling , and money laundering , cases involving United Nations trade sanctions , conflict diamonds and CITES .
Assistant	It –> deals with cases of fraud in [relation to] –> direct taxes It –> deals with cases of fraud in [relation to] –> indirect taxes It –> deals with cases of fraud in [relation to] –> tax credits It –> deals with cases of fraud in [relation to] –> drug smuggling It –> deals with cases of fraud in [relation to] –> money laundering It –> deals with cases involving –> United Nations trade sanctions It –> deals with cases involving –> conflict diamonds It –> deals with cases involving –> CITES It –> deals –> with cases of fraud in [relation to] money laundering It –> deals with cases of –> fraud in [relation to] direct taxes cases –> involving –> United Nations trade sanctions cases –> involving –> conflict diamonds cases –> involving –> CITES
User	Because of this association , St. Michael was considered to be the patron saint of colonial Maryland , and as such was honored by the river being named for him .
Assistant	St. Michael –> was considered [to be] –> [the] patron saint [of colonial Maryland] St. Michael –> was considered [to be] [the] patron saint –> of colonial Maryland St. Michael –> was honored by –> [the] river being named for him [the] river –> [being] named for –> him St. Michael –> was considered [to be] [the] patron saint [of colonial Maryland] Because of –> [this] association
User	The show was designed to appear as if the viewer was channel surfing through a multi-channel wasteland , happening upon spoof adverts , short sketches , and recurring show elements .
Assistant	[The] viewer –> was happening upon –> spoof adverts [The] viewer –> was happening upon –> short sketches [The] viewer –> was happening upon –> recurring show elements [the] viewer –> was channel surfing through –> [a] [multi-channel] wasteland [The] show –> was designed to appear as if [the] viewer was channel surfing through –> [a] multi-channel wasteland [The] show –> was designed to appear as if [the] viewer was happening upon –> spoof adverts [The] show –> was designed to appear as if [the] viewer was happening upon –> short sketches [The] show –> was designed to appear as if [the] viewer was happening upon –> recurring show elements
User	*{INPUT SENTENCE}*

Table 11: Instruct gpt to annotate LSOIE-g sentences with few-shot OpenIE examples. *{INPUT SENTENCE}* should be replaced with the sentence to annotate. The few-shot examples have been structured for multi-turn conversations.

serve that when num epoch is 80, the fact of the intermediate sentence generated with top_p = 0.3 is consistent with the given extraction while the latter (top_p = 0.9) includes an additional fact, "*NSW Fire Commissioner Adrian O'Farrell said*", not mentioned in the given extraction. This example illustrates the result shown in Figure 3 that even as the epoch approaches 80, there remains a significant gap in performance between the setting of top_p=0.9 and the others.

num epoch	intermediate output
0	The Blue Mountains region of New South Wales, Australia, has been ravaged by a massive fire.
5	A massive fire is feared to have destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
10	A massive fire is feared to have killed more than one hundred homes.
15	A massive fire is feared to have destroyed more than one hundred homes.
20	A massive fire is feared to have occurred in the Blue Mountains region of New South Wales, Australia.
25	A massive fire is feared to have killed residents and destroyed object more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
30	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
35	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
40	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
45	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
50	A massive fire is feared to have killed residents and destroyed and object more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
55	A massive fire is feared to have killed residents and destroyed hundred homes " in the Blue Mountains region of New South Wales, Australia.
60	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
65	A massive fire is feared to have killed residents and destroyed more than one hundred homes ", in the Blue Mountains region of New South Wales, Australia.
70	A massive fire is feared to have killed residents and destroyed more than one hundred homes ".
75	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
80	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.

Table 12: Intermediate sentences generated by E2S model(num_beams=4) in ESE cycle .

num epoch	intermediate output
0	The fire was feared to have occurred in a flooded area of the Blue Mountains region of New South Wales, Australia.
5	A massive fire is feared to have killed more than one hundred homes, and is feared to have destroyed more than one hundred homes.
10	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
15	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
20	A massive fire is feared to have killed residents and destroyed object more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
25	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
30	A massive fire is feared to have killed residents and destroyed object more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
35	A massive fire is feared to have killed residents and destroyed object to more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
40	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
45	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
50	A massive fire is feared to have killed residents and destroyed more than one hundred homes, in the Blue Mountains region of New South Wales, Australia.
55	A massive fire is feared to have killed residents and destroyed more than one hundred homes, in the Blue Mountains region of New South Wales, Australia.
60	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
65	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
70	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
75	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
80	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.

Table 13: Intermediate sentences generated by E2S model (top_p=0.3) in ESE cycle.

num epoch	intermediate output
0	The fire raged on the outskirts of the Blue Mountains region of New South Wales, Australia, and was quickly extinguished.
5	A massive fire, which was feared to have killed at least one hundred homes, is feared to have destroyed more than one hundred homes.
10	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
15	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
20	A massive fire is feared to have killed residents and destroyed object more than one hundred homes.
25	A massive fire is feared to have killed residents and destroyed object more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
30	A massive fire is feared to have killed residents and destroyed object more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
35	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
40	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.
45	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
50	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
55	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
60	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
65	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
70	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
75	A massive fire is feared to have killed residents and destroyed more than one hundred homes " in the Blue Mountains region of New South Wales, Australia.
80	A massive fire is feared to have killed residents and destroyed more than one hundred homes in the Blue Mountains region of New South Wales, Australia.

Table 14: Intermediate sentences generated by E2S model (top_p=0.5) in ESE cycle .

num epoch	intermediate output
0	A massive fire was feared to have destroyed objectet ” more than one hundred homes, and the blaze was not the only one that sparked the fire.
5	A massive fire, believed to have destroyed more than one hundred homes, is feared to have killed more than one hundred residents.
10	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
15	A massive fire is feared to have killed residents and destroyed one hundred homes.
20	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
25	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
30	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
35	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
40	A massive fire is feared to have killed residents and destroyed more than one hundred homes ”.
45	A massive fire is feared to have killed residents and destroyed more than one hundred homes ”.
50	A massive fire is feared to have killed residents and destroyed more than one hundred homes ” in the Blue Mountains region of New South Wales, Australia.
55	A massive fire is feared to have killed residents and destroyed more than one hundred homes,” Australian Fire and Rescue Service -LRB- FRT -RRB- said in a statement.
60	A massive fire is feared to have killed residents and destroyed more than one hundred homes,” Australian Fire Commissioner David Jones said in a statement.
65	A massive fire is feared to have killed residents and destroyed more than one hundred homes,” Australian Fire and Rescue Service said in a statement.
70	A massive fire is feared to have killed residents and destroyed more than one hundred homes, ” Australian Fire Service -LRB- AFP -RRB- said in a statement.
75	A massive fire is feared to have killed residents and destroyed more than one hundred homes,” Australian Geological Survey Deputy Chief Richard Bowyer said.
80	A massive fire is feared to have killed residents and destroyed more than one hundred homes,” Australian Fire Commissioner Len McCluskey said.

Table 15: Intermediate sentences generated by E2S model (Top_p=0.7) in ESE cycle.

num epoch	intermediate output
0	A massive fire was feared to have occurred in a mountain, a mountain.
5	A fire in the Blue Mountains region of New South Wales has claimed the lives of at least 145 people and destroyed more than 100 homes.
10	A massive fire is feared to have occurred in the Blue Mountains region of New South Wales, Australia.
15	A massive fire is feared to have killed residents, and destroyed more than one hundred homes.
20	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
25	A massive fire is feared to have killed residents, destroyed more than one hundred homes and destroyed more than one hundred businesses in the Blue Mountains region of New South Wales, Australia.
30	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
35	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
40	A massive fire is feared to have killed residents, and destroyed more than one hundred homes.
45	A massive fire is feared to have killed residents and destroyed more than one hundred homes, in the Blue Mountains region of New South Wales, Australia.
50	A massive fire is feared to have killed residents and destroyed more than one hundred homes, despite warnings from businesses across the Blue Mountains region.
55	A massive fire is feared to have killed residents and destroyed more than one hundred homes, " said the Australian Fire Brigade.
60	A massive fire is feared to have killed residents and destroyed more than one hundred homes.
65	A massive fire is feared to have killed residents and destroyed more than one hundred homes, in the Blue Mountains region of New South Wales, Australia.
70	A massive fire is feared to have killed residents and destroyed more than one hundred homes, making it the deadliest fire in Australian history.
75	A massive fire is feared to have killed residents and destroyed more than one hundred homes. "
80	A massive fire is feared to have killed residents, and destroyed more than one hundred homes, " NSW Fire Commissioner Adrian O'Farrell said.

Table 16: Intermediate sentences generated by E2S model (top_p=0.9) in ESE cycle.