Mirror: A Universal Framework for Various Information Extraction Tasks

Anonymous EMNLP submission

Abstract

The variety of information extraction tasks and data formats make it hard to share common knowledge among those tasks. This causes wastes to some extent and adds difficulties to build complex pipeline applications in real scenarios. Recent studies formulate IE tasks as a triplet extraction problem. However, such format does not support multi-span and n-ary extraction tasks, leading to weak versatility. To this end, we reorganize IE datasets into a unified format and propose a universal framework for various IE tasks, namely Mirror. We regard IE tasks as a multi-span cyclic graph extraction problem, and devise a non-autoregressive graph decoding algorithm to extract all spans in a single step. This graph structure is flexible, and it supports span-only MRC, label-only classification, and label-span mixed information extraction tasks. We manually construct a corpus containing 62 datasets for model pretraining, and experiments on 30 datasets across 8 tasks show that our model has good compatibilities and achieves SOTA performances under few-shot and zero-shot settings. The code, model weights and data will be publicly available at GitHub.

1 Introduction

Information Extraction (IE) is a fundamental task in Natural Language Processing (NLP), which aims to extract structured information from unstructured text (Grishman, 2019), such as Named Entity Recognition (NER), Relation Extraction (RE), Event Extraction (EE), etc. However, each IE task is usually isolated with specific data structures and delicate models, which makes it difficult to share knowledge across tasks (Lu et al., 2022; Josifoski et al., 2022).

In order to unify the data formats and take advantage of common features between different tasks, there are two main routes in recent studies. The first is to utilize generative pretrained language models (PLMs) to generate the structured information directly. Lu et al. (2022) and Paolini et al. (2021) structure the IE tasks as a sequence-to-sequence problem, and use generative models to predict the structured information autoregressively. However, such methods cannot provide the exact positions of the structured information, which is important for NER and fair evaluations (Hao et al., 2023). Besides, the generation-based methods are usually slow, and it consumes huge resources to train on large-scale datasets (Wang et al., 2022). The second is to apply the extractive PLMs, which is way more faster to train and inference. USM takes the IE tasks into a triplet prediction problems via semantic matching (Lou et al., 2023). However, such method is limited in a small range of triplet-based tasks, and not suitable for multi-span and n-ary IE tasks.

To extend the universal IE system into more

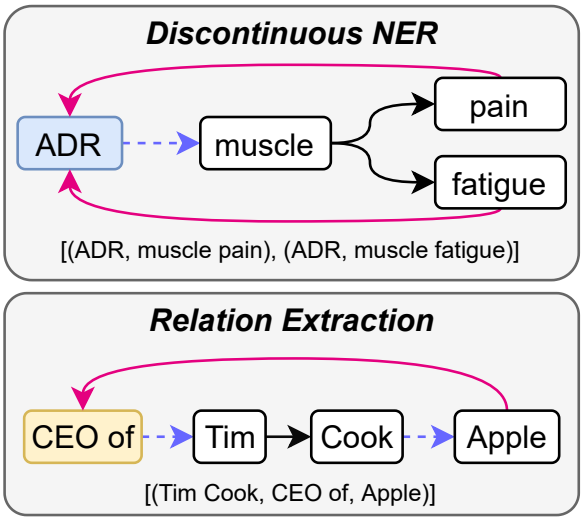

Figure 1: Multi-span cyclic graph for discontinuous NER and RE tasks (best viewed in color). The spans are connected by three types of edges, including ***consecutive connections***, dotted ***jump connections*** and ***tail-to-head connections***. *ADR* in discontinuous NER denotes the entity label of Adverse Drug Reaction.

Model	TANL	UIE	DeepStruct	USM	Mirror
PLM	T5-base	T5-large	GLM	RoBERTa	DeBERTa-v3
#Params	220M	770M	10B	large 372M	large 434M
Decoding	AR	AR	AR	NAR	NAR
Indexing	✗	✗	✗	✓	✓
Multi-span	✗	○	○	○	✓
N-ary	✗	✗	✗	✗	✓
Cls.	✗	✗	✗	✗	✓
MRC	✗	✗	✗	○	✓

Table 1: Comparisons with other systems. AR denotes the auto-regressive decoding while NAR is the non-autoregressive decoding strategy. Indexing means whether the model could provide exact information positions. ○ indicates the model supports the task theoretically, but the implementation is not available. Multi-span means the model supports multi-span extraction, e.g. the discontinuous named entity recognition task. N-ary denotes the ability of n-ary tuple extraction. Cls. represents the classfication task support. MRC stands for the machine reading comprehension task support.

tasks, we propose *Mirror*, a new IE framework that can be applied in multi-span extraction, n-ary extraction, machine reading comprehension (MRC) and even classification tasks. As examplified in Figure 1, we formulate IE tasks into a unified multi-slot tuple extraction problem, and transform those tuples into multi-span cyclic graphs. This graph structure is rather flexible and scalable. It can be applied to span-only MRC tasks, label-only classification tasks, and label-span mixed IE tasks. Mirror takes schemas as part of the model inputs, and this benefits few-shot and zero-shot tasks naturally.

We conduct extensive experiments on 30 datasets from 8 tasks, including NER, RE, EE, Aspect-based Sentiment Analysis (ABSA), multi-span discontinuous NER, n-ary hyper RE, MRC and classfication. To enhance the few-shot and zero-shot abilities, we manually collect 62 datasets into a whole corpus for model pretraining. Our Mirror shows good compatibility across different tasks and datasets, and achieves competitive results on few-shot and zero-shot settings.

Our contributions are summarized as follows:

- We propose a unified schema-guided multi-slot extraction paradigm, which is capable of span-only MRC, label-only classification and label-span mixed information extraction tasks.
- We propose Mirror, a universal non-autoregressive framework that transforms multiple tasks into a multi-span cyclic graph.
- We conduct extensive experiments on 30 datasets from 8 tasks, and the results show that our model achieves competitive results on single-tasks, and outperforms previous SOTA systems on few-shot and zero-shot settings.

2 Related Work

2.1 Multi-task Information Extraction

Multi-task IE is a popular research topic in recent years. The main idea is to use a single model to perform multiple IE tasks. IE tasks could be formulated as different graph structures. Li et al. (2022) formulate flat, nested, and discontinuous NER tasks as a graph with next-neighboring and tail-to-head connections. Maximal cliques also have been used to flat & discontinuous NER tasks (Wang et al., 2021) and trigger-available & trigger-free event extractions (Zhu et al., 2022). DyGIE++ takes NER, RE and EE tasks as span graphs, and apply iterative propagation to enhance spans' contextual representations (Wadden et al., 2019). OneIE uses the similar graph structures with global constraint features (Lin et al., 2020).

In addition to explicit graph-based multi-task IE systems, generative language models are also been widely used. Yan et al. (2021b) and Yan et al. (2021a) add special index tokens into BART (Lewis et al., 2020) vocabulary to help perform various NER and ABSA tasks and obtain explicit span positions. TANL (Paolini et al., 2021) apply T5 (Raffel et al., 2020) to generate texts with special enclosures as the predicted information. GenIE (Josifoski et al., 2022) and DeepStruct (Wang et al., 2022) share a similar idea to generate subject-relation-object triplets, and DeepStruct extends the

model size to 10B with GLM as the backbone (Du et al., 2022).

2.2 Schema-guided Information Extraction

In schema-guided IE systems, schemas are input as an guidance signal to help the model extracting target information. UIE (Lu et al., 2022) categorize IE tasks into span spotting and asscociating elementary tasks and devise a linearized query language. Fei et al. (2022) introduces the hyper relation extraction task to represent complex IE tasks like EE, and utilize external parsing tools to enhance the text representations.

While the above methods use flexible generative language models, they cannot predict exact positions, which brings ambiguity when evaluating. Besides, large generative language models are usually slow to train and inference, and requires tons of computing resources. USM (Lou et al., 2023) utilizes BERT-family models to extract triplets non-autoregressively. USM regards IE tasks into a unified schema matching task and use a label-text matching model to extract triplets. However, these methods cannot extend more information extraction tasks, such as multi-span discontinuous NER, and n-ary information extractions.

3 Mirror

3.1 Unified Data Interface

To make the model able to handle different IE tasks, we propose a unified data interface for the input. As shown in Figure 2, there are three parts in the model input: the *instruction*, the *schema labels*, and the *text*. The instruction is composed of a leading token `[I]` and a natural language sentence. The leading token indicates the instruction part while the sentence tells the model what it should do. For example, the instruction of NER could be *Please identify any possible entities in the given text and label them with the following types*. The instruction is the question in Machine Reading Comprehension (MRC) and Question Answering (QA) datasets. For each task, we manually design a set of instructions, and randomly pick one of them for each training sample. The number of instructions for each IE task is listed in Table 2.

The schema labels are task ontologies that used for schema-guided extraction. This part is consists of special token labels (`[LM]`, `[LR]` and `LC`) and corresponding label texts. Among the special tokens, `[LM]` denotes the label of mentions (or event types),

Task	#Dataset	#Instruction	#Instance
Cls		42	
MRC		42	
NER		42	
RE		53	
EE		40	

Table 2: Pretraining dataset statistics.

`[LR]` denotes the label of relations (or argument roles), and `[LC]` denotes the label of classes. `[LC]` token is designed for classification tasks when pretraining.

The text part is the input text that the model should extract information from. It is composed of a leading token (`[TL]` or `[TP]`) and a natural language sentence. If the leading token is `[TL]`, the model should link labels from schema labels to spans in the text. While the `[TP]` token indicates the target spans are only in the text, and the model should extract information from the text without schema labels. The `[TP]` label is used in the pretraining stage to make the model able to extract information in MRC tasks without schema. In classification tasks when pretraining, the model should not extract anything from the text part. So we add a special background area with a leading token `[B]` to distinguish from extractive texts.

With the above three parts, we can formulate classification, extractive MRC, multi-choice MRC, and IE tasks into a unified data interface, and the model can be trained in a unified way even the model is not based on generative language models.

3.2 Multi-slot Tuple and Multi-span Cyclic Graph (MCG)

We formulate IE tasks as a unified multi-span extraction problem. For the flat NER task, the model is expected to extract a tuple like: "((entity label position,), (span start position, span end position))". As shown in Figure 1 and the top right of Figure 2, we formulate IE tasks into a unified multi-span cyclic graph, and regard labels as the leading tokens in schema labels. There are three types of connections in the graph: the ***consecutive*** connection, the ***jump*** connection, and the ***tail-to-head*** connection.

The consecutive connection is used to **spans in the same entity**. For an entity that has multiple tokens, the consecutive connection connects from

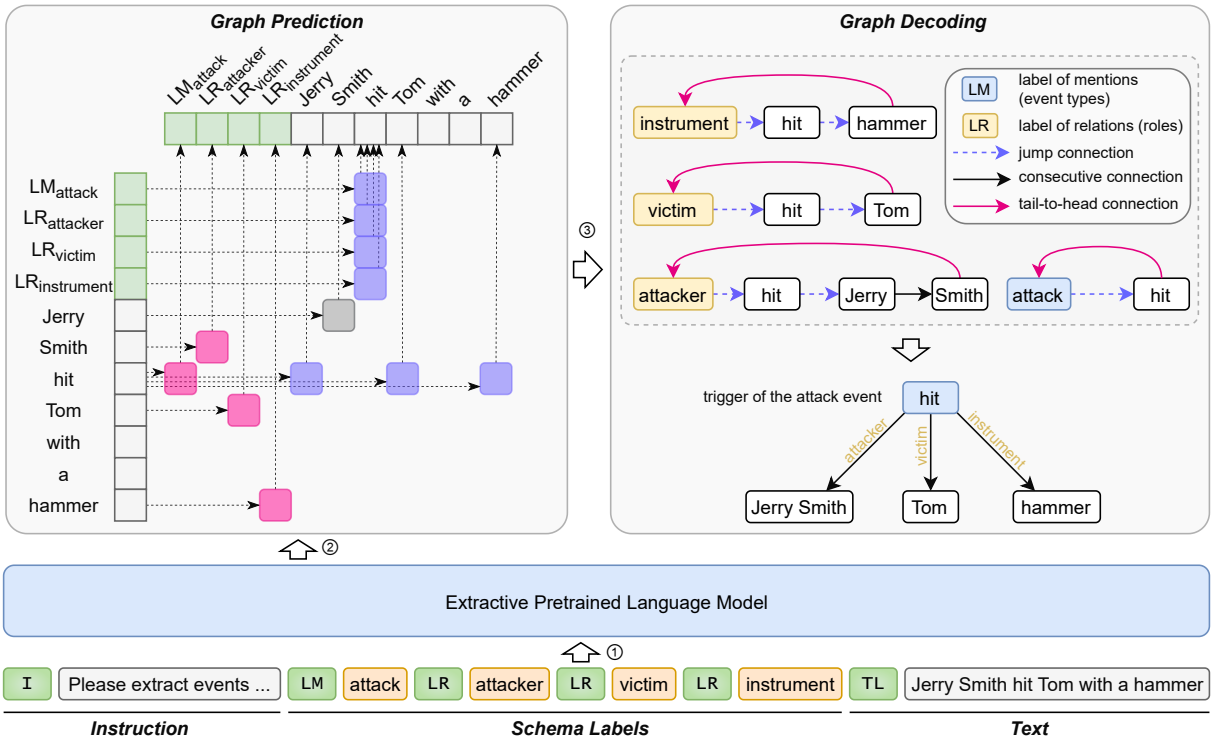

Figure 2: Model framework (best viewed in color).

the first token to the last token. As shown in Figure 2, "Jerry" connects to "Smith". If there is only one token in an entity, the consecutive connection is not used. For example, entities in "muscle pain and fatigue" contains two entities "muscle pain" and "muscle fatigue". The consecutive connection is used to connect from "muscle" to "pain", and "muscle" to "fatigue". The jump connection connects **different slots** in a tuple. Schema labels and spans from texts are in different slots, so they are connected in jump connections. In addition, the head entity and the tail entity of a relation triplet are in different slots, so they are also connected in jump connections. The tail-to-head connection helps **locate the start & end boundaries**, and forms a cycle in the graph. It connects from the last token of the last slot to the first token of the first slot in a tuple.

3.3 Model Structure

With the unified data interface and the multi-span cyclic graph, we propose a unified model structure for IE tasks. Similar to Yu et al. (2020), we use biaffine attention to obtain the adjacency matrix of the multi-span cyclic graph. For token representations $\mathbf{H}$, we get the adjacency matrix by Equation 1.

$$\left.\begin{matrix}\\ \odot \\ \\ \end{matrix}\right\rangle_{=\infty} \mathbf{HW}_{\text{biaffine}}\mathbf{H}^{\top}+\mathbf{HW}_{\text{linear}}+\mathbf{W}_{\text{linear}}^{\top}\mathbf{H}^{\top}+\mathbf{W}_{\text{bias}} \quad (1)$$

4 Experiments

4.1 Experiment Setup

4.2 Datasets

4.3 Main Results

4.4 Few-shot Results

4.5 Zero-shot Results

4.6 Analysis on Training Strategies

4.7 Ablation Study

Limitations

Content input length and model compatibility. Multi-turn result modification. Laborious data cleaning and format unification.

Ethics Statement

All datasets are publicly available without further annotation, and the NLU tasks are traditional tasks in natural language processing communities. So there are no ethical issues.

Task	Datasets	TANL	UIE	DeepStruct	InstructUIE	USM	UniEX	RexUIE	Mirror
NER	ACE04	-	86.89	-	-	87.62	87.12	87.25	87.63
	ACE05	84.90	85.78	86.9	86.66	87.14	87.02	87.23	86.81
	CoNLL03	91.70	92.99	93	92.94	93.16	92.65	93.67	92.83
RE	ACE05	63.70	66.06	66.8	-	67.88	66.06	64.87	66.70
	CoNLL04	71.40	75.00	78.3	78.48	78.84	73.40	78.39	71.48
	NYT	-	93.54	93.3	90.47	94.07	-	94.55	91.66
	SciERC	-	36.53	-	45.15	37.36	38.00	38.37	36.08
Event	ACE05-Tgg	68.40	73.36	69.8	77.13	72.41	74.08	75.17	69.95
	ACE05-Arg	47.60	54.79	56.2	72.94	55.83	53.92	59.15	48.12
	CASIE-Tgg	-	69.33	-	67.80	71.73	71.46	73.01	69.19
	CASIE-Arg	-	61.30	-	63.53	63.26	62.91	63.87	55.09
ABSA	14-res	-	74.52	-	-	77.26	74.77	77.46	77.20
	14-lap	-	63.88	-	-	65.51	65.23	66.41	62.84
	15-res	-	67.15	-	-	69.86	68.58	70.84	68.82
	16-res	-	75.07	-	-	78.25	76.02	77.20	75.53

Table 3: Main results.

	P	R	F1
Discontinuous NER: CADEC			
BART-NER	70.08	71.21	70.64
W2NER	74.09	**72.35**	**73.21**
Mirror	**74.83**	67.88	71.19
N-ary Tuples: HyperRED			
CubeRE	66.39	67.12	66.75
RexUIE	-	-	**75.20**
Mirror	70.88	64.05	67.29

Table 4: Results on multi-span and n-ary inforamtion extraction tasks. The best results are in **bold**, and the second best results are underlined.

	Model	1-shot	5-shot	10-shot	Avg.
NER CoNLL03	UIE	57.53	75.32	79.12	70.66
	USM	71.11	83.25	84.58	79.65
	RexUIE	**86.57**	**89.63**	**90.82**	**89.07**
	Mirror	77.50	82.73	84.48	81.57
RE CoNLL04	UIE	34.88	51.64	58.98	48.50
	USM	36.17	53.20	60.99	50.12
	RexUIE	**43.80**	**54.90**	**61.68**	**53.46**
	Mirror	34.66	52.23	58.68	48.52
Event Trigger ACE05	UIE	42.37	53.07	54.35	49.93
	USM	40.86	55.61	58.79	51.75
	RexUIE	**56.95**	64.12	**65.41**	**62.16**
	Mirror	49.50	**65.61**	60.68	58.60
Event Arg ACE05	UIE	14.56	31.20	35.19	26.98
	USM	19.01	36.69	42.48	32.73
	RexUIE	**30.43**	41.04	**45.14**	**38.87**
	Mirror	23.46	**48.32**	41.90	37.89
ABSA 16res	UIE	23.04	42.67	53.28	39.66
	USM	30.81	52.06	58.29	47.05
	RexUIE	37.70	49.84	60.56	49.37
	Mirror	**67.06**	**73.51**	**68.70**	**69.76**

Table 5: Few-shot results. The best results are in **bold**, and the second best results are underlined.

References

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. In *NeurIPS*.

Ralph Grishman. 2019. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692.

Peng Hao, Wang Xiaozhi, Yao Feng, Zeng Kaisheng, Hou Lei, Li Juanzi, Liu Zhiyuan, and Shen Weixing. 2023. The Devil is in the Details: On the Pitfalls of Event Extraction Evaluation. ArXiv:2306.06918 [cs].

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics:*

Model	Movie	Restaurant	AI	Literature	Music	Politics	Science	Avg.
Davinci	0.84	2.94	2.97	9.87	13.83	18.42	10.04	8.42
ChatGPT	41.00	**37.76**	**54.40**	54.07	**61.24**	59.12	**63.00**	**52.94**
USM	37.73	14.73	28.18	**56.00**	44.93	36.10	44.09	37.39
InstructUIE	**63.00**	20.99	49.00	47.21	53.61	48.15	49.30	47.32
Mirror	40.96	20.02	51.13	44.80	60.63	**61.19**	53.65	47.48
Upper Bound	85.94	83.30	65.72	67.93	78.25	75.92	70.96	75.43

Table 6: Zero-shot NER results. The best results are in **bold**, and the second best results are underlined. The upper bound is the Mirror performance where these zero-shot NER training sets are included in the pretraining phase.

Dataset	Training	NER CoNLL03	RE CoNLL04	Event Trigger ACE05	Event Arg ACE05	ABSA 16res	Avg.
Included in PT	Multi-task FT						
	Single-task FT	92.83	71.48	69.95	48.12	75.53	71.58
Excluded in PT	Multi-task FT	91.84	72.39	68.24	47.73	75.31	71.10
	Single-task FT	92.45	73.70	71.01	52.57	75.55	73.06

Table 7: Analysis on training strategy.

Human Language Technologies, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10965–10973. AAAI Press.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. *CoRR*, abs/2301.03282.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.*, 21:140:1–140:67.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

	NER CoNLL03	RE CoNLL04	Event Trigger ACE05	Event Arg ACE05	ABSA 16res	Avg.
Mirror - Pretrain - Pretrain & Instruction						

Table 8: Ablation study.

Yucheng Wang, Bowen Yu, Hongsong Zhu, Tingwen Liu, Nan Yu, and Limin Sun. 2021. Discontinuous named entity recognition as maximal clique discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 764–774, Online. Association for Computational Linguistics.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021a. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Yuan, and Min Zhang. 2022. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4552–4558. International Joint Conferences on Artificial Intelligence Organization. Main Track.

A Dataset Statistics

test

B Multi-span Cyclic Graph Decoding

This is a section in the appendix.

Figure 3: Python Code - find_paths_from_adj_mat

```python
def find_paths_from_adj_mat(adj_mat: torch.Tensor) -> list[tuple[int]]:
    assert adj_mat.shape[0] == adj_mat.shape[1] and len(adj_mat.shape) == 2

    paths = []
    self_loops = set()
    adj_map = defaultdict(set)
    rev_adj_map = defaultdict(set)
    # current -> next
    for c, n in adj_mat.detach().nonzero().tolist():
        # self-loop
        if c == n:
            self_loops.add(c)
        else:
            adj_map[c].add(n)
            # reversed map
            rev_adj_map[n].add(c)
    for self_loop_node in self_loops:
        paths.append((self_loop_node,))

    def track(path: tuple[int], c: int):
        visited: set[tuple[int]] = set()
        stack = [(path, c)]
        while stack:
            path, c = stack.pop()
            if c in adj_map:
                for n in adj_map[c]:
                    if (c, n) in visited:
                        continue
                    visited.add((c, n))
                    stack.append((path + (c,), n))
            # else:
            if path:
                paths.append(path + (c,))

    start_nodes = set(adj_map.keys()) - set(rev_adj_map.keys())
    for c in start_nodes:
        ns = adj_map[c]
        for n in ns:
            track((c,), n)

    return paths
```