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 machine reading comprehension, label-only classification, and label-span mixed information extraction tasks. We manually construct a corpus containing 57 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

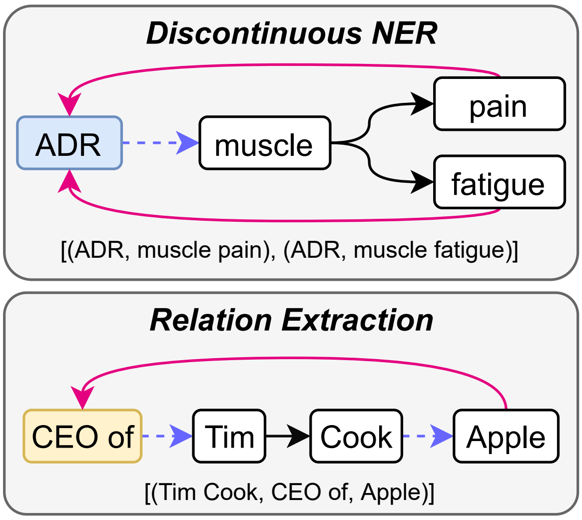

Figure 1: Multi-span cyclic graph for discontinuous NER and RE tasks (best viewed in color). The spans are connected by three types of edges, including ***consecutive connections***, dotted ***jump connections*** and ***tail-to-head connections***. *ADR* in discontinuous NER denotes the entity label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

(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

Model	TANL	UIE	DeepStruct	InstructUIE	USM	Mirror
PLM #Params	T5-base 220M	T5-large 770M	GLM 10B	FlanT5 11B	RoBERTa large 372M	DeBERTa-v3 large 434M
Decoding	AR	AR	AR	AR	NAR	NAR
Indexing	✗	✗	✗	✗	✓	✓
Triplet	✓	✓	✓	✓	✓	✓
Single-span NER	✓	✓	✓	✓	✓	✓
Multi-span	✗	○	○	○	✗	✓
N-ary	✗	✗	✗	✗	✗	✓
Cls.	✗	✗	✗	○	✗	✓
MRC	✗	✗	✗	✗	○	✓

Table 1: Comparisons with other systems. **Circle** ○ indicates the model supports the task theoretically, but the implementation is not available. **AR** denotes the auto-regressive decoding while **NAR** is the non-autoregressive decoding strategy. **Indexing** means whether the model could provide exact information positions. **Triplet** stands for "(head, relation, tail)" triplet extraction. **Single-span NER** denotes flat, and nested NER tasks with consecutive spans. **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, e.g. quadruple extraction. **Cls.** represents the classfication and multi-choice Machine Reading Comprehension (MRC) support. **MRC** stands for extractive Question Answering (QA) and extractive MRC 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 57 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

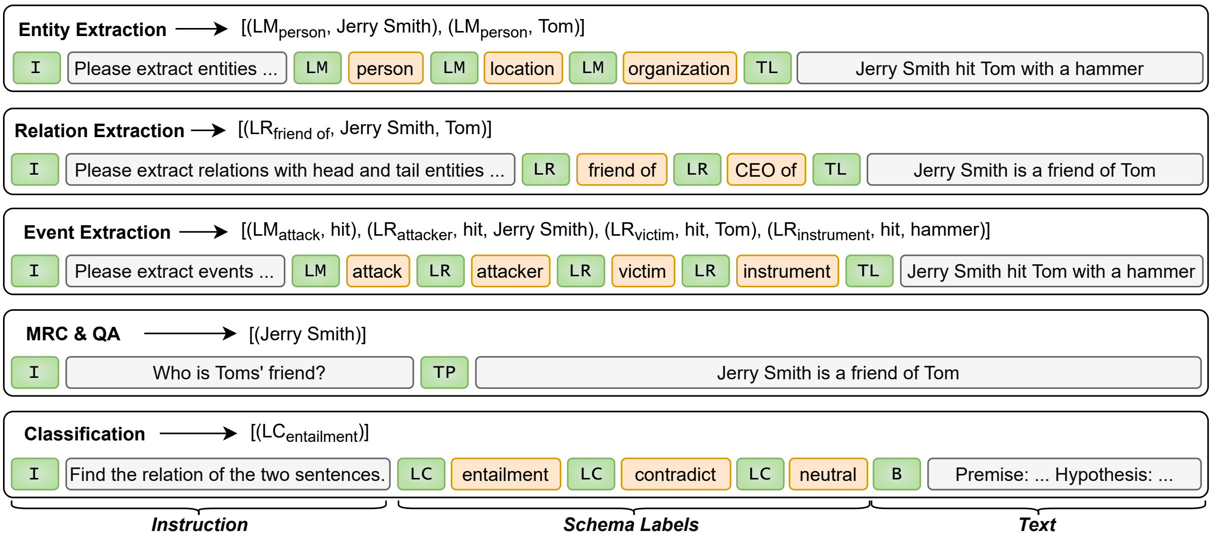

Figure 2: Unified data interface.

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. InstructUIE (Wang et al., 2023) formulates schemas into instructions and uses FlanT5-11B (Chung et al., 2022) to performing multi-task instruction tuning.

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

In this section, we introduce the Mirror framework. We first address the unified data input format to the model, then introduce the unified task formulation and the model structure.

3.1 Unified Data Interface

To make the model able to handle different IE tasks, we propose a unified data interface for the model input. As shown in Figure 2, there are three parts: 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.

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), `[LR]` denotes the label of relations (or argument roles), and `[LC]` denotes the label of classes. `[LC]`

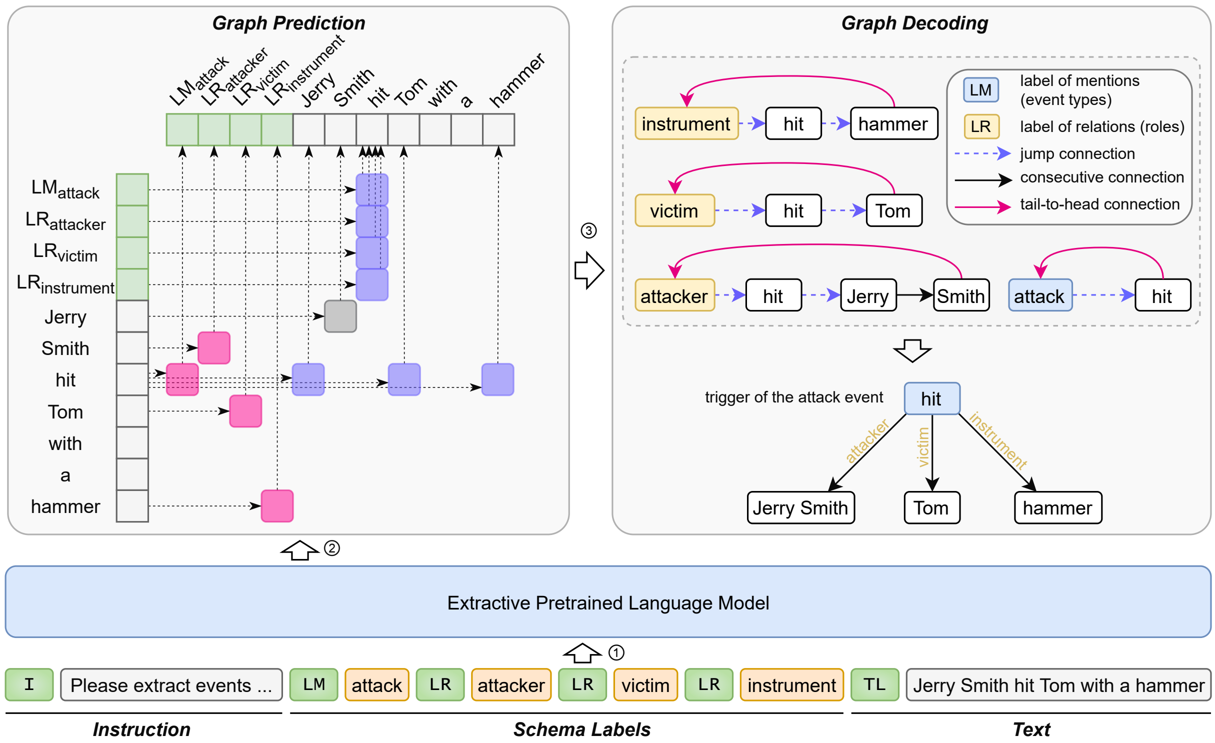

Figure 3: Model framework (best viewed in color).

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 (and extractive QA), 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. For the robust model training, we manually collect 57 datasets across 5 tasks to make a corpus for model pretraining. To balance the number of examples in each task, we randomly sample instances for each dataset. If the number of instances in a dataset is less than the sampling value, we keep the original dataset unchanged and do not perform over sampling. For NER, RE and EE tasks, we manually design a set of instructions, and randomly pick one of them for each sample. The number of instructions for each IE task is listed in Table 2. For more detailed statistics on each dataset, please refer to Appendix A.

Task	#Dataset	#Samples/Dataset	#Instruction	#Instance
NER	15	20,000	42	171,609
Cls♣	27	5,000	54,070	134,758
RE	9	20,000	9	123,876
MRC♡	5	30,000	75,200	85,658
EE	1	All	40	2,898
Total	57	-	-	518,799

Table 2: Pretraining dataset statistics. ♣ Classification tasks contain multi-choice MRC datasets. ♡ MRC stands for both extractive QA and extractive MRC datasets.

3.2 Multi-slot Tuple and Multi-span Cyclic Graph

We formulate IE tasks as a unified multi-slot tuple extraction problem. As exemplified in Figure 2, in the RE task, the model is expected to extract a 3-slot tuple like (`relation, head entity, tail entity`). Here, the tuple is (LR$_{\texttt{friend of}}$, `Jerry Smith, Tom`). The length of tuple slots could vary across tasks, so Mirror is capable of n-ary extraction problems.

As shown in Figure 1 and the top right of Figure 3, we formulate multi-slot tuples 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 adopted to **spans in the same entity**. For an entity that has multiple tokens, the consecutive connection connects from the first token to the last token. As shown in Figure 3, "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.

In practice, we convert the answer of each slot into span positions. For schema labels, we use the position of leading tags instead of label texts. For text spans like entities, the position is a one-digit number if there is only one character, otherwise the start and end positions are listed. For example, the 3-slot relation tuple (`LR`$_{\texttt{friend of}}$, `Jerry Smith`, `Tom`) will be converted into ($9 \vdots 16 \rightarrow 17 \vdots 22$), where $\vdots$ denotes the jump connection, $\rightarrow$ stands for the consecutive connection, 9 is the position of `LR`$_{\texttt{friend of}}$, 16 and 17 express *Jerry Smith*, and 22 is the position of *Tom*. There is also a tail-to-head connection from 22 to 9. The corresponding graph decoding algorithm is shown in Algorithm 1. During inference, we first find the forward chain (9,16,17,22), and then verify the chain with tail-to-head connection (22→9). After that, the multi-slot tuple is revealed with jump connections($9 \vdots 16$) and ($17 \vdots 22$).

Algorithm 1 MULTI-SPAN CYCLIC GRAPH DECODING

Input: Adjacency matrix $\mathcal{A}$
Output: A set of multi-slot tuples $\mathcal{T}$

```
T ← {}
Ã ← A^c | A^j                 ▷ merge consecutive and jump connections
Find forward chains C from Ã
for c ∈ C do                  ▷ find legal paths with tail-to-head connections
    if c meets the need in A^t then
        split c into a tuple t via A^j
        T ← T ∪ t
    end if
end for
return T
```

3.3 Model Structure

With the unified data interface and the multi-span cyclic graph, we propose a unified model structure for IE tasks. For each token x_i from the inputs, Mirror transforms it into a vector $h_i \in \mathbb{R}^{d_h}$ via a BERT-style extractive pretrained language model (PLM). Similar to Yu et al. (2020), we use biaffine attention to obtain the adjacency matrix $\mathcal{A}$ of the multi-span cyclic graph. Mirror calculates the linking probability $p_{ij}^k, k \in \{\text{consecutive}, \text{jump}, \text{tail-to-head}\}$ between x_i and x_j. The final $\mathcal{A}$ is obtained via thresholding ($\mathcal{A}_{ij}^k = 1$ if $p_{ij}^k > 0.5$ else 0).

$$\begin{aligned} \tilde{h}_i = \text{FFNN}_s\left(h_i\right), \quad \tilde{h}_j = \text{FFNN}_e\left(h_j\right) \\ p_{ij}^k = \text{sigmoid}\left(\tilde{h}_i{}^{\top} U \tilde{h}_j / \sqrt{d_h}\right) \end{aligned} \tag{1}$$

where $\tilde{h}_i, \tilde{h}_j \in \mathbb{R}^{d_b}$. $U \in \mathbb{R}^{d_b \times 3 \times d_b}$ is trainable parameter, and 3 denotes consecutive, jump and tail-to-head connections. FFNN is the feed forward neural network with rotary positional embedding as introduced in Su et al. (2021). The FFNN is composed of linear transformation, GELU activation function (Hendrycks and Gimpel, 2023) and dropout (Srivastava et al., 2014).

During training, we adopt the imbalance-class multi-label categorical cross entropy (Su et al., 2022) as the loss function.

$$\mathcal{L}(i,j) = \log\left(1 + \sum_{\Omega_{\text{neg}}} e^{p_{ij}^k}\right) + \log\left(1 + \sum_{\Omega_{\text{pos}}} e^{-p_{ij}^k}\right) \tag{2}$$

where Ω_{neg} stands for negative samples ($\mathcal{A}_{ij}^k = 0$), and Ω_{pos} denotes positive samples ($\mathcal{A}_{ij}^k = 1$).

Task	Datasets	TANL	DeepStruct	UIE	InstructUIE	USM	Mirror w/ PT w/ Inst.	Mirror w/ PT w/o Inst.	Mirror w/o PT w/ Inst.	Mirror w/o PT w/o Inst.
NER	ACE04	-	-	86.89	-	87.62	87.16	86.39	87.66	87.26
	ACE05	84.90	86.90	85.78	86.66	87.14	85.34	85.70	86.72	86.45
	CoNLL03	91.70	93.00	92.99	92.94	93.16	92.73	91.93	92.11	92.97
RE	ACE05	63.70	66.80	66.06	-	67.88	67.86	67.86	64.88	69.02
	CoNLL04	71.40	78.30	75.00	78.48	78.84	75.22	72.96	71.19	73.58
	NYT	-	93.30	93.54	90.47	94.07	93.85		93.95	93.31
	SciERC	-	-	36.53	45.15	37.36	36.89	37.12	36.66	40.50
EE	ACE05-Tgg	68.40	69.80	73.36	77.13	72.41	74.44	73.05	72.66	73.38
	ACE05-Arg	47.60	56.20	54.79	72.94	55.83	55.88	54.73	56.51	57.87
	CASIE-Tgg	-	-	69.33	67.80	71.73	71.81		73.09	71.40
	CASIE-Arg	-	-	61.30	63.53	63.26	61.27		60.44	58.87
ABSA	14-res	-	-	74.52	-	77.26	75.06	74.24	76.05	75.89
	14-lap	-	-	63.88	-	65.51	64.08	62.48	59.56	60.42
	15-res	-	-	67.15	-	69.86	66.40	63.61	60.26	67.41
	16-res	-	-	75.07	-	78.25	74.24	75.40	73.13	77.46

Table 3: Results on single IE tasks.

4 Experiments

4.1 Experiment Setup

We utilize DeBERTa-v3-large (He et al., 2021) as the PLM with a max sequence length to 512. The biaffine size d_b is 512 with a dropout rate of 0.3. The epoch number of pretraining is 3 with a learning rate of 2e-5. For more detailed hyper-parameter settings, please refer to Appendix B. Datasets are processed following Lu et al. (2022) (13 IE datasets in Table 3 and 4 datasets in 5), Li et al. (2022) (CADEC), Chia et al. (2022) (HyperRED), Lou et al. (2023) (zero-shot NER datasets in Table 6), Rajpurkar et al. (2018) (SQuAD v2.0) and Wang et al. (2019) (GLUE datasets).

4.2 Main Results

Main results on single IE tasks are presented in Table 3.

4.3 Few-shot Results

4.4 Zero-shot Results

5 Results on MRC and classification

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. We believe there are no ethical issues in this paper.

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

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

References

Yew Ken Chia, Lidong Bing, Sharifah Mahani Aljunied, Luo Si, and Soujanya Poria. 2022. A dataset for hyper-relational extraction and a cube-filling approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10114–10133, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam

Task	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
	Mirror				
RE CoNLL04	UIE	34.88	51.64	58.98	48.50
	USM	36.17	53.20	60.99	50.12
	Mirror				
Event Trigger ACE05	UIE	42.37	53.07	54.35	49.93
	USM	40.86	55.61	58.79	51.75
	Mirror				
Event Arg ACE05	UIE	14.56	31.20	35.19	26.98
	USM	19.01	36.69	42.48	32.73
	Mirror				
ABSA 16res	UIE	23.04	42.67	53.28	39.66
	USM	30.81	52.06	58.29	47.05
	Mirror				

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

Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

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].

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian Error Linear Units (GELUs). ArXiv:1606.08415 [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: 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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual

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								
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.

Model	SQuAD 2.0 (EM/F1)	CoLA (Mcc)	QQP (Acc)	MNLI (Acc)	SST-2 (Acc)	QNLI (Acc)	RTE (Acc)	MRPC (Acc)
BERT-large	79.0/81.8	60.6	91.3	-	93.2	92.3	70.4	84.1
RoBERTa-large	86.5/89/4	68.0	92.2	90.2	96.4	93.9	86.6	88.8
DeBERTa v3-large	89.0/91.5	75.3	93.0	91.9	96.9	96.0	92.7	92.2
Mirror$_{\text{direct}}$	40.4/67.4	63.9	84.8	85.9	93.6	91.6	85.9	89.2

Table 7: Results on MRC and classification tasks.

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global Pointer: Novel Efficient Span-based Approach for Named Entity Recognition. ArXiv:2208.03054 [cs].

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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. ArXiv:2304.08085 [cs].

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

This section contains detailed statistics for pretraining datasets and fine-tuning datasets. Pretraining data statistics are listed in Table 8, 9, 10, 11 and 12.

B Hyper-parameter Settings

Table 13 shows the hyper-parameters in our experiments. For few-shot experiments, we follow Lu et al. (2022) and generate 1-, 5-, 10-shot data with 5 seeds.

Name	#Instruction	#Instance
ag_news	5	5,000
ANLI♣	29	15,000
ARC	3,361	3,370
CoLA	43	5,000
CosmosQA	4,483	5,000
cos_e	5,000	5,000
dbpedia	6	5,000
DREAM	3,842	5,000
hellaswag	20	5,000
IMDB	26	5,000
MedQA	5,000	5,000
MNLI	29	5,000
MRPC	40	3,668
MultiRC	4,999	5,000
OpenBookQA	4,835	4,957
QASC	4,832	5,000
QNLI	31	5,000
QQP	40	5,000
RACE	4,482	5,000
RACE-C	4,782	5,000
ReClor	3,368	4,638
RTE	29	2,490
SciQ	4,989	5,000
SNLI	29	5,000
SST-2	26	5,000
Winogrande	20	5,000
WNLI	31	635
Total	54,070	134,758

Table 8: Pretraining data statistics on classification. ♣: ANLI contains 3 subsets, so the total number is greater than 5,000.

Name	#Instruction	#Instance
AnatEM	42	5,861
bc2gm	42	12,500
bc4chemd	42	20,000
bc5cdr	42	4,560
Broad_Tweet_Corpus	42	5,334
FabNER	42	9,435
FindVehicle	42	20,000
GENIA	42	15,023
HarveyNER	42	3,967
MultiNERD	42	20,000
NCBIdiease	42	5,432
ontoNotes5	42	20,000
TweetNER7	42	7,103
WikiANN_en	42	20,000
WNUT-16	42	2,394
Total	42	171,609

Table 9: Pretraining data statistics on NER.

Name	#Instruction	#Instance
ADE_corpus	9	3417
FewRel	9	20000
GIDS	9	8526
kbp37	9	15807
New-York-Times-RE	9	20000
NYT11HRL	9	20000
semeval	9	8000
WebNLG	9	5019
Wiki-ZSL♣	9	23107
Total	9	123,876

Table 10: Pretraining data statistics on RE.

Name	#Instruction	#Instance
BiPaR	11,524	11,668
ms_marco_v2.1	20,000	20,000
newsqa	19,659	20,000
squad_v2	19,998	20,000
SubjQA	4,060	13,990
Total	75,220	85,658

Table 11: Pretraining data statistics on MRC.

Name	#Instruction	#Instance
PHEE	40	2,898
Total	40	2,898

Table 12: Pretraining data statistics on EE.

Item	Setting
warmup proportion	0.1
pretraining epochs	3
fine-tuning epochs	20
fine-tuning epoch patience	3
few-shot epochs	200
few-shot epoch patience	10
batch size	8
PLM learning rate	2e-5
PLM weight decay	0.1
others learning rate	1e-4
max gradient norm	1.0
dropout	0.3

Table 13: Hyper-parameter settings.