Easy-to-Hard Learning for Information Extraction[*]

Chang Gao[1,2], **Wenxuan Zhang**[2†], **Wai Lam**[1], **Lidong Bing**[2]
[1]The Chinese University of Hong Kong
[2]DAMO Academy, Alibaba Group
{gaochang,wlam}@se.cuhk.edu.hk
{saike.zwx,l.bing}@alibaba-inc.com

Abstract

Information extraction (IE) systems aim to automatically extract structured information, such as named entities, relations between entities, and events, from unstructured texts. While most existing work addresses a particular IE task, universally modeling various IE tasks with one model has achieved great success recently. Despite their success, they employ a one-stage learning strategy, i.e., directly learning to extract the target structure given the input text, which contradicts the human learning process. In this paper, we propose a unified easy-to-hard learning framework consisting of three stages, i.e., the easy stage, the hard stage, and the main stage, for IE by mimicking the human learning process. By breaking down the learning process into multiple stages, our framework facilitates the model to acquire general IE task knowledge and improve its generalization ability. Extensive experiments across four IE tasks demonstrate the effectiveness of our framework. We achieve new state-of-the-art results on 13 out of 17 datasets. Our code is available at `https://github.com/DAMO-NLP-SG/IE-E2H`.

1 Introduction

Information extraction (IE) is a crucial task in natural language processing (NLP) that involves extracting structured knowledge from unstructured text data (Bing et al., 2013, 2015), enabling various applications such as information retrieval (Ruambo and Nicholaus, 2019), knowledge graph construction (Oramas et al., 2016; Wang et al., 2019), and question answering (Khot et al., 2017). Depending on what kind of information is to be extracted, IE consists of a wide range of tasks, including named entity recognition (NER) (Li et al., 2022a), joint entity and relation extraction (RE) (Taillé et al., 2020; Chia et al., 2022), event extraction (EE) (Li et al., 2022b), and aspect-based sentiment analysis (ABSA) (Zhang et al., 2022b).

Traditionally, IE has been approached with specialized models that are designed to handle specific IE tasks. For example, NER is often formulated as a sequence labeling (Ma and Hovy, 2016; Xu et al., 2021b) or span-based classification (Wang et al., 2020) problem. The more complex RE or EE task is usually solved with pipeline approaches that split the original task into several sequential subtasks and design specific models for each subtask (Subburathinam et al., 2019; Yang et al., 2019; Peng et al., 2020). These models often require extensive task-specific knowledge to design dedicated model architectures and thus suffer from poor generalization. Recently, motivated by pre-trained generative models such as T5 (Raffel et al., 2020) that handle multiple tasks with the unified text-to-text format, there has been a shift towards the use of unified models for IE as well, which can tackle all IE tasks with a single model structure. For example, TANL (Paolini et al., 2021) tackles various IE tasks with a text-to-text generative model by framing them as translation between augmented natural languages. UIE (Lu et al., 2022) models heterogeneous IE structures into a uniform representation via a structural extraction language.

Despite the success of existing unified models on various IE tasks, they typically adopt a one-stage learning paradigm, i.e., directly learning to predict the target structure given the input text. In contrast, humans often learn to tackle a task in an easy-to-hard manner. They learn basic concepts or skills before solving more complex problems and often tackle harder examples to gain a better understanding of the problem. Taking the RE task as an example, it aims to extract relational triplets, where each

[*] This work was supported by Alibaba Group through Alibaba Research Intern Program. It was also partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719). This work was done when Chang Gao was an intern at Alibaba DAMO Academy.

[†]Wenxuan Zhang is the corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 11913–11930
July 9-14, 2023 ©2023 Association for Computational Linguistics

triplet consists of a head entity, a relation, and a tail entity. To tackle it, humans first learn some basic skills, such as identifying entities, recognizing relations, and associating entities and relations, before extracting complex relational triplets. This process facilitates humans to learn meaningful substructures and the dependencies among them. Moreover, in practical scenarios, humans usually encounter harder cases, i.e., long input context of multiple sentences containing more entities and relations. By solving hard cases, humans improve their understanding of the task and problem-solving skills. By comparison, models are only trained with the provided training data. The gap between the model and human learning strategies hinders IE models from further development.

To bridge the gap, we propose an **easy-to-hard (E2H)** learning framework for IE tasks in this paper. E2H mimics the human learning procedure to learn each IE task in stages, i.e., the easy stage, the hard stage, and the main stage. The easy stage aims to help the model acquire basic skills of the task, and the hard stage aims to assist the model in handling broad-range variations of the task via training the model with diverse and harder data. Finally, the main stage focuses on the main task at hand for training. Thus an immediate question is how to prepare the data with different levels of difficulty for the easy and hard stages. It is labor-intensive and challenging to construct such data manually. In this work, we attempt only to leverage the existing data of the main task for constructing the data.

Specifically, for the easy stage, we observe that the target IE structure often has meaningful substructures. Therefore, we identify several basic skills for each task according to the substructures of its target structure. Returning to the RE example, the skills can be recognizing the entities, relations, and dependencies between them. We can automatically construct training data for learning these skills by modifying the input prompt and decomposing the target structure of the main task. For the hard stage, we combine two training instances of the main task to build a harder training instance by concatenating their input texts to form the new text and their targets to build the new target. The new instance contains more entities, relations, and complicated contexts, making it harder than the original instances. Through these two novel construction strategies, we can reduce much human effort to obtain the data for different stages.

To summarize, our contributions are three-fold: (1) We propose a unified easy-to-hard (E2H) learning framework for IE tasks by imitating the human learning process; (2) We develop two novel strategies to build the easy and hard stages of our framework without using any additional resources; (3) We conduct comprehensive evaluations on 17 datasets across four IE tasks and achieve state-of-the-art results on 13 datasets. Notably, our E2H method consistently outperforms the one-stage learning counterpart by introducing two extra learning stages with an average increase of 0.38, 2.96, 1.33, and 1.39 absolute points on the NER, RE, EE, and ABSA tasks, respectively.

2 Task Definition

This paper investigates four common IE tasks, i.e., NER, RE, EE, and ABSA. In this section, we provide formal definitions of these tasks. Detailed examples of these tasks are in Appendix A.3.

Named Entity Recognition (NER) Given an input text T, the task is to identify and classify entities in T into predefined categories, i.e., extract $\{(e_i, c_i)\}$, where e_i is the i-th entity, which is a continuous text span in T, $c_i \in \mathcal{C}$ is its category, and $\mathcal{C}$ is the entity category set.

Relation Extraction (RE) Given an input text T, RE is to identify a set of (head entity, relation, tail entity) triplets, i.e., extract $\{((e_i^h, c_i^h), r_i, (e_i^t, c_i^t))\}$, where the superscripts h and t denote the head and tail entities, $r_i \in \mathcal{R}$ is the i-th relation, and $\mathcal{R}$ is the relation set.

Event Extraction (EE) Given an input text T, the task is to identify a set of events where each event consists of an event trigger and a set of corresponding arguments, i.e., extract $\{((e_i^{tri}, c_i^{tri}), (e_i^{arg_1}, c_i^{arg_1}), \cdots, (e_i^{arg_m}, c_i^{arg_m}))\}$, where e_i^{tri} is the i-th trigger, which is a continuous text span in T, $c_i^{tri} \in \mathcal{C}_{event}$ is its category, $e_i^{arg_j}$ is the j-th argument of the i-th event, which is also a continuous text span in T, $c_i^{arg_j} \in \mathcal{C}_{event}$ is its category, and $\mathcal{C}_{event}$ consists of all event and argument categories.

Aspect-based Sentiment Analysis (ABSA) There are four essential elements in ABSA, namely aspect category c, aspect term a, opinion term o, and sentiment polarity p. We focus on the aspect sentiment triplet extraction (ASTE) task (Peng et al., 2020) and the aspect sentiment quad

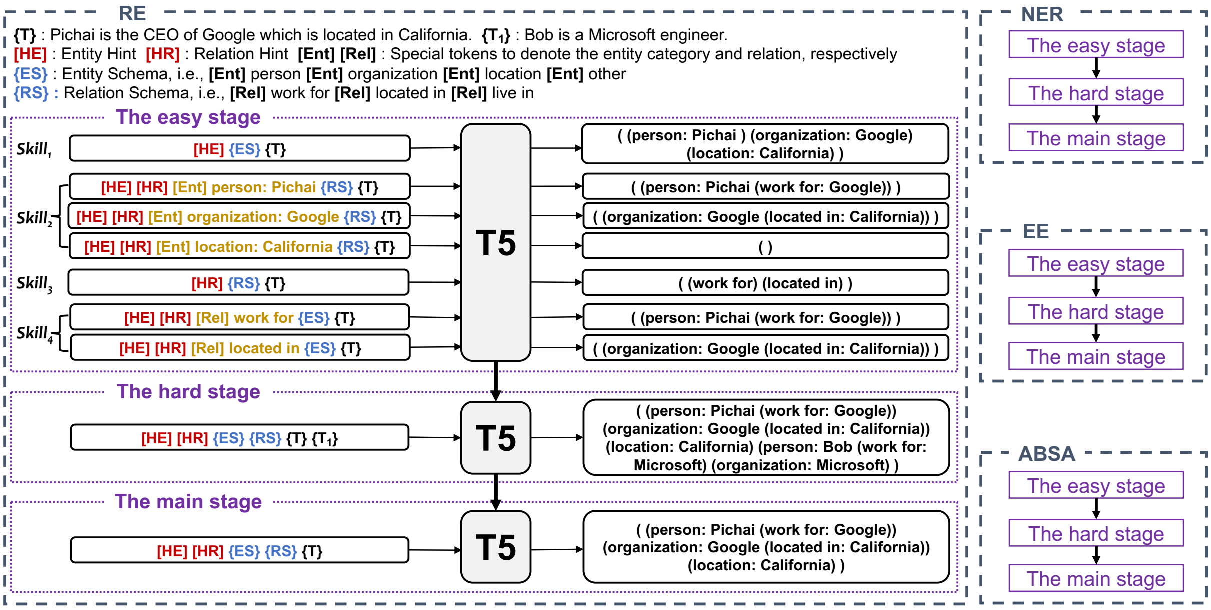

RE
{T} : Pichai is the CEO of Google which is located in California. {T1} : Bob is a Microsoft engineer.
[HE] : Entity Hint [HR] : Relation Hint [Ent] [Rel] : Special tokens to denote the entity category and relation, respectively
{ES} : Entity Schema, i.e., [Ent] person [Ent] organization [Ent] location [Ent] other
{RS} : Relation Schema, i.e., [Rel] work for [Rel] located in [Rel] live in
The easy stage
Skill1
[HE] {ES} {T}
((person: Pichai) (organization: Google) (location: California))
Skill2
[HE] [HR] [Ent] person: Pichai {RS} {T}
[HE] [HR] [Ent] organization: Google {RS} {T}
[HE] [HR] [Ent] location: California {RS} {T}
((person: Pichai (work for: Google)))
((organization: Google (located in: California)))
( )
Skill3
[HR] {RS} {T}
((work for) (located in))
Skill4
[HE] [HR] [Rel] work for {ES} {T}
[HE] [HR] [Rel] located in {ES} {T}
((person: Pichai (work for: Google)))
((organization: Google (located in: California)))
T5
The hard stage
[HE] [HR] {ES} {RS} {T} {T1}
T5
((person: Pichai (work for: Google)) (organization: Google (located in: California)) (location: California) (person: Bob (work for: Microsoft) (organization: Microsoft))
The main stage
[HE] [HR] {ES} {RS} {T}
T5
((person: Pichai (work for: Google)) (organization: Google (located in: California)) (location: California))
NER
The easy stage
The hard stage
The main stage
EE
The easy stage
The hard stage
The main stage
ABSA
The easy stage
The hard stage
The main stage

Figure 1: Overview of E2H consisting of three stages, i.e., the easy stage, the hard stage, and the main stage. We highlight `Hint` in red, `Constraint` in brown, and `Schema` in blue.

prediction (ASQP) task (Zhang et al., 2021a) given their popularity. Given an input text T, the ASTE task is to identify a set of $\{(a_i, o_i, p_i)\}$ triplets, and the ASQP task is to identify a set of $\{(c_i, a_i, o_i, p_i)\}$ quadruplets, where $c_i \in \mathcal{C}_{absa}$ is i-th aspect category, a_i is i-th aspect term, o_i is i-th opinion term, both a_i and o_i are continuous spans in T, $p_i \in$ {positive, negative, neutral} is i-th sentiment polarity, and $\mathcal{C}_{absa}$ is the aspect category set.

3 Our E2H Framework

Our proposed easy-to-hard (E2H) framework consists of three sequential stages: the easy stage, the hard stage, and the main stage. In this section, we first introduce our text-to-structure formulation for facilitating three-stage learning in a unified framework. Next, we will describe how to realize the easy and hard stages. Finally, we will discuss the main stage as well as the detailed training and inference process of our framework.

3.1 Unified Text-to-Structure Formulation

Similar to UIE (Lu et al., 2022), we formulate NER, RE, EE, and ABSA as text-to-structure generation problems, which allows us to use a single model to tackle multiple tasks. Given a text T and its corresponding prompt P, we aim to generate the target IE structure S with an encoder-decoder model $M : (P, T) \rightarrow S$. To facilitate the learning of different stages, we design the prompt P containing three types of information: `Hint`, `Constraint`, and `Schema`. `Hint` guides the model on what elements should be extracted, `Constraint` indicates specific constraints for the task, and `Schema` provides necessary information such as the possible relation set for the extraction. With these three types of information, the prompt is able to connect the learning process in different stages.

Taking the RE task as an example, as depicted in Figure 1, `Hint` consists of one or both of an entity hint and a relation hint. The entity hint, represented by the special token `[HE]`, guides the model to extract entities, and the relation hint, represented by the special token `[HR]`, guides the model to extract relations. The use of both hints guides the model to extract both entity and relation information, in the form of (head entity, relation, tail entity) triplets. `Constraint` is a specific entity or relation, which limits the target structure to be related to that entity or relation. Lastly, `Schema` contains pre-defined entity categories or relations or both of them, depending on the information that needs to be extracted. It provides essential information for identifying entities and relations in a text.

3.2 The Easy Stage

The goal of the easy stage is to enable the model to learn basic skills that will aid in tackling the main task. To achieve this, we identify several skills for each task and automatically construct the training

Task	Basic Skills
NER	*Skill*$_1$: $T \rightarrow$ a set of entity categories $\{c_i\}$ *Skill*$_2$: T and an entity category constraint $c \rightarrow$ a set of entities of c $\{(e_i, c)\}$
RE	*Skill*$_1$: $T \rightarrow$ a set of entities $\{(e_i, c_i)\}$ *Skill*$_2$: T and a head entity constraint $(e^h, c^h) \rightarrow$ a set of relational triplets $\{((e^h, c^h), r_i, e^t_i)\}$ *Skill*$_3$: $T \rightarrow$ a set of relations $\{r_i\}$ *Skill*$_4$: T and a relation constraint $r \rightarrow$ a set of relational triplets $\{((e^h_i, c^h_i), r, e^t_i)\}$
EE	*Skill*$_1$: $T \rightarrow$ a set of event triggers $\{(e^{tri}_i, c^{tri}_i)\}$ *Skill*$_2$: T and a trigger constraint $(e^{tri}, c^{tri}) \rightarrow$ the event $((e^{tri}, c^{tri}), (e^{arg_1}, c^{arg_1}), \cdots, (e^{arg_m}, c^{arg_m}))$
ASTE	*Skill*$_1$: $T \rightarrow$ a set of aspect terms $\{a_i\}$ and a set of opinion terms $\{o_i\}$ *Skill*$_2$: T and an aspect term constraint $a \rightarrow$ a set of triplets $\{(a, o_i, p_i)\}$ *Skill*$_3$: $T \rightarrow$ a set of sentiment polarities $\{p_i\}$ *Skill*$_4$: T and a sentiment polarity constraint $p \rightarrow$ a set of triplets $\{(a_i, o_i, p)\}$
ASQP	*Skill*$_1$: $T \rightarrow$ a set of aspect categories $\{c_i\}$ *Skill*$_2$: $T \rightarrow$ a set of (aspect category, aspect term) tuples $\{(c_i, a_i)\}$ *Skill*$_3$: $T \rightarrow$ a set of (aspect category, opinion term) tuples $\{(c_i, o_i)\}$ *Skill*$_4$: $T \rightarrow$ a set of (aspect category, sentiment polarity) tuples $\{(c_i, p_i)\}$

Table 1: Basic skills for NER, RE, EE, ASTE, and ASQP. We omit `Hint` and `Schema` for simplicity. Detailed examples are in Appendix A.3.

data for them based on the data of the main task. Table 1 presents the basic skills of NER, RE, EE, ASTE, and ASQP. We design each skill to be a subtask of the main task according to its target structure. These skills are more fundamental and well-defined. Combining these skills gives the model a whole picture of how to tackle the main task. For example, the RE task has four skills. Skill$_1$ and Skill$_3$ help the model recognize substructures of the relational triplet, i.e., the entity and relation, respectively, and Skill$_2$ and Skill$_4$ help the model learn the dependencies between these substructures.

To construct the training data for each skill, we modify the input and target of the main task's training data. Specifically, the input text is the same for the skills and the main task, but the prompt is different. As shown in Figure 1, for the RE task, there is only `[HE]` in the hint of Skill$_1$ as it only extracts entities and only `[HR]` in the hint of Skill$_3$ as it only extracts relations. Both `[HE]` and `[HR]` are in the hints of Skill$_2$, Skill$_4$, and the main task because they extract (head entity, relation, tail entity) triplets. For Skill$_2$ and Skill$_4$, there is also a `Constraint`, i.e., a head entity or relation, which requires their targets to be triplets related to a specific head entity or relation. The schema of the RE task consists of both entity categories and relations. For a specific skill of RE, the schema only contains entity categories or relations. The target of each skill is a part of the target of the RE task. For Skill$_1$ and Skill$_3$, which extract a substructure of the relational triplet, we use the substructure as the target. For Skill$_2$ and Skill$_4$, we use the corresponding subset of triplets of the RE task as the target.

3.3 The Hard Stage

The hard stage aims to construct training examples that are harder than the original training examples of the main task to train the model. Intuitively, the training instance is harder if the input text contains more structural elements and more complicated contexts. To this end, we combine two training instances of the original task to construct a harder instance. Formally, given two training instances (P, T_1, S_1) and (P, T_2, S_2), we can construct a harder training instance $(P, T_1 \circ T_2, S_1 \circ S_2)$, where P is the prompt, T_i is the i-th text, S_i is the i-th target structure, and $\circ$ denotes concatenation. An example is shown in the hard stage part of the RE task in Figure 1. The model has to process and understand the combined information from both instances, making it more challenging for the model to correctly extract the target structure.

Let N denote the number of training examples of the original task. For each training example, we randomly sample M training examples whose target structures are not empty to construct M hard instances. This results in a total of $N * M$ hard instances. This approach allows us to easily construct a large amount of diverse hard training data.

3.4 The Main Stage

After training the model in the easy and hard stages, we train the model with the main task in this stage.

Training We adopt the pre-trained sequence-to-sequence model T5 (Raffel et al., 2020) as the backbone of E2H. The model is trained with a maximum likelihood objective. Given the training example (P, T, S), the loss function L_θ is defined as

$$L_\theta = -\sum_{i=1}^{n} \log P_\theta (S_i \mid S_{<i}, P, T) \qquad (1)$$

where θ is the model parameters, P is the prompt, T is the text, S is the target structure, and n is the length of S. We train the model in the easy, hard, and main stages sequentially. For the easy stage, we adopt the weights of pre-trained T5 to initialize the model. For the hard and main stages, we initialize the model with the weights of the model trained in the previous stage.

Inference Once the training process is complete, we use the model trained in the main stage to generate the target structure S for any given tuple of the prompt and text (P, T). Although our training process has three stages, the inference is a one-stage process. The computational load is the same as that of the one-stage learning counterpart.

4 Experiments

4.1 Experimental Setup

Datasets We conduct experiments on 17 datasets across four IE tasks, i.e., NER, RE, EE, and ABSA. We evaluate the flat NER task with CoNLL03 (Tjong Kim Sang and De Meulder, 2003), and the nested NER task with ACE04-Ent (Mitchell et al., 2005) and ACE05-Ent (Walker et al., 2006). For RE, we experiment on CoNLL04 (Roth and Yih, 2004), ACE05-Rel (Walker et al., 2006), and SciERC (Luan et al., 2018). Regarding to EE, we use ACE05E, ACE05E+ (Walker et al., 2006), and CASIE (Satyapanich et al., 2020). As for ABSA, we consider the ASTE and ASQP tasks. For ASTE, we adopt four popular datasets, including Rest14, Laptop14, Rest15, and Rest16 provided by Xu et al. (2020). For ASQP, we use R-ACOS and L-ACOS provided by Cai et al. (2021), and Rest15 and Rest16 provided by Zhang et al. (2021a). These ABSA datasets are derived from the datasets provided by the SemEval ABSA challenges (Pontiki et al., 2014, 2015, 2016), except L-ACOS which is collected from the Amazon Laptop domain. Statistics of these datasets are provided in Appendix A.1.

Evaluation We use Micro-F1 as the primary evaluation metric. For each experimental result, we report the average performance on three random seeds. For NER, RE, EE, and ASTE, we follow Lu et al. (2022) to use Entity F1, Relation Strict F1, Event Trigger F1 and Argument F1, and Sentiment Triplet F1 as the evaluation metrics and map the generated string-level extraction results to offset-level for evaluation. For ASQP, we follow Zhang et al. (2021a) to use Sentiment Quad F1 to evaluate the model. A sentiment quad is correct if and only if the four elements are exactly the same as those in the gold sentiment quad.

Baselines We divide our baselines into two categories: specialized models and unified models. Specialized models are designed for a particular IE task, while unified models are designed for general IE. For specialized models, we use state-of-the-art methods such as BARTNER (Yan et al., 2021) and DeBias (Zhang et al., 2022a) for NER, UniRE (Wang et al., 2021) and PURE (Zhong and Chen, 2021) for RE, Text2Event (Lu et al., 2021) and DEGREE (Hsu et al., 2022) for EE, and PARAPHRASE (Zhang et al., 2021a) and Seq2Path (Mao et al., 2022) for ABSA. For unified models, we use TANL (Paolini et al., 2021), UIE (Lu et al., 2022), and LasUIE (Fei et al., 2022) as baselines. To make a fair comparison with one-stage learning methods, we also build T5-base and T5-large baselines. We set their inputs and outputs the same as those of E2H and only train them in the main stage.

Implementation Details E2H has two model sizes: E2H-base and E2H-large, which are initialized with pre-trained T5-base and T5-large models (Raffel et al., 2020), respectively. Other details are reported in Appendix A.2.

4.2 Main Results

We compare E2H with state-of-the-art specialized and unified models. Tables 2-4 report the experimental results on 17 datasets across four IE tasks. We have the following observations: (1) E2H is an effective framework for various IE tasks. E2H-large achieves new state-of-the-art results on 13 out of 17 datasets. (2) The proposed easy-to-hard three-stage learning method consistently outperforms the one-stage learning counterpart. E2H performs better than T5 on all the datasets for two model sizes,

Models	NER				RE			
	CoNLL03	ACE04-Ent	ACE05-Ent	Avg	CoNLL04	ACE05-Rel	SciERC	Avg
Specialized Models								
BARTNER (Yan et al., 2021)	**93.24**	86.84	84.74	88.27	-	-	-	-
DeBias (Zhang et al., 2022a)	93.12	85.28	84.93	87.78	-	-	-	-
UniRE (Wang et al., 2021)	-	-	-	-	-	64.30	36.90	-
PURE (Zhong and Chen, 2021)	-	-	-	-	-	64.80	36.80	-
Unified Models								
TANL (Paolini et al., 2021)	91.70	-	84.90	-	71.40	63.70	-	-
UIE* (Lu et al., 2022)	92.99	86.89	85.78	88.55	75.00	66.06	36.53	59.20
LasUIE* (Fei et al., 2022)	93.20	86.80	86.00	**88.67**	75.30	**66.40**	-	-
T5-base (Raffel et al., 2020)	91.72	85.60	84.16	87.16	69.58	62.91	33.13	55.20
T5-large (Raffel et al., 2020)	92.05	86.78	85.76	88.20	71.72	64.49	35.44	57.21
E2H-base	91.92	86.24	84.83	87.66	72.23	65.44	35.06	57.58
E2H-large	92.43	**87.06**	**86.25**	88.58	**75.31**	66.21	**39.00**	**60.17**

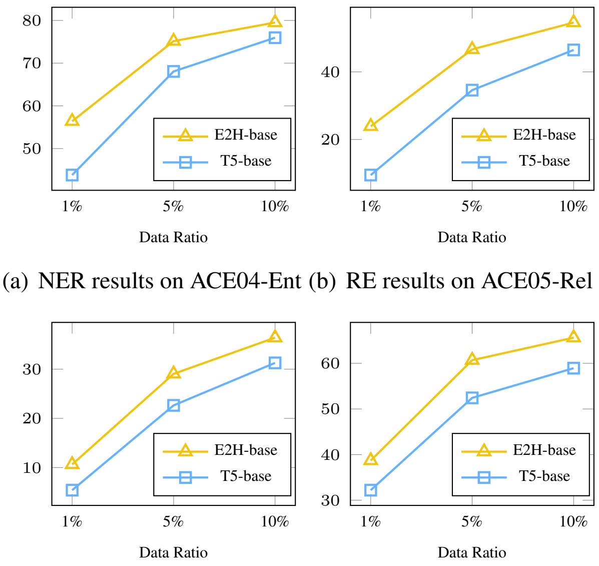
Table 2: Experimental results on the NER and RE tasks. The best results are in bold and the second-best results are underlined. Models marked with $*$ conduct large-scale continued pre-training with external resources. Except for T5-base and T5-large, the results of baselines are taken from their original papers.

Models	ACE05-E		ACE05-E+		CASIE		Avg	
	Trig F1	Argu F1	Trig F1	Argu F1	Trig F1	Argu F1	Trig F1	Argu F1
Specialized Models								
Text2Event (Lu et al., 2021)	71.90	53.80	71.80	54.40	-	-	-	-
DEGREE (Hsu et al., 2022)	**73.30**	**55.80**	70.90	**56.30**	-	-	-	-
Unified Models								
TANL (Paolini et al., 2021)	68.40	47.60	-	-	-	-	-	-
UIE* (Lu et al., 2022)	-	-	73.36	54.79	69.33	61.30	-	-
T5-base (Raffel et al., 2020)	68.19	49.68	69.68	50.65	68.40	60.19	68.76	53.51
T5-large (Raffel et al., 2020)	70.40	52.42	71.45	54.08	69.29	60.98	70.38	55.83
E2H-base	70.12	50.98	69.99	52.85	68.45	60.40	69.52	54.74
E2H-large	72.19	53.85	**73.50**	55.67	**69.58**	**61.96**	**71.76**	**57.16**

Table 3: Experimental results on the EE task. The best results are in bold and the second-best results are underlined. Models marked with $*$ conduct large-scale continued pre-training with external resources. Except for T5-base and T5-large, the results of baselines are taken from their original papers.

Models	ASTE					ASQP				
	Rest14	Laptop14	Rest15	Rest16	Avg	R-ACOS	L-ACOS	Rest15	Rest16	Avg
Specialized Models										
PARAPHRASE (Zhang et al., 2021a)	72.03	61.13	62.56	71.70	66.86	-	-	46.93	57.93	-
Seq2Path (Mao et al., 2022)	75.52	64.82	65.88	72.87	69.77	58.41	42.97	-	-	-
Unified Models										
UIE* (Lu et al., 2022)	74.52	63.88	67.15	75.07	70.16	-	-	-	-	-
T5-base (Raffel et al., 2020)	72.11	63.06	66.27	72.24	68.42	59.26	43.12	48.24	58.92	52.39
T5-large (Raffel et al., 2020)	73.48	63.62	67.08	74.85	69.76	61.24	44.37	51.76	60.93	54.58
E2H-base	75.40	65.78	68.58	73.83	70.90	60.66	43.51	49.45	59.55	53.29
E2H-large	**75.92**	**65.98**	**68.80**	**75.46**	**71.54**	**63.50**	**44.51**	**52.39**	**61.86**	**55.57**

Table 4: Experimental results on two ABSA tasks, including the ASTE task and the ASQP task. The best results are in bold and the second-best results are underlined. Models marked with $*$ conduct large-scale continued pre-training with external resources. Except for T5-base and T5-large, the results of baselines are taken from their original papers.



(a) NER results on ACE04-Ent (b) RE results on ACE05-Rel

(c) EE results on ACE05-E (d) ABSA results on Rest14

Figure 2: Results of E2H-base and T5-base in low-resource scenarios.

and E2H-large obtains an average improvement of 0.38, 2.96, 1.33, and 1.39 absolute points over T5-large on the NER, RE, EE, and ABSA tasks, respectively. This demonstrates the strong generalization ability of our framework. (3) Without using any external resources, our method exhibits comparable or stronger performance than models with large-scale continued pre-training. Compared with UIE (Lu et al., 2022), which is pre-trained with large-scale structured, unstructured, and parallel data, E2H-large achieves better performance on the RE, EE, and ASTE tasks and obtains comparable results on the NER task. (4) Easy-to-hard learning brings more benefits to complex tasks than simple tasks. Specifically, compared with the improvement on the NER task, which only extracts entities, the improvements of E2H over T5 are more significant on the other three tasks, which extract tuples with multiple elements. This shows that our method can help the model effectively capture the structural dependency of complex structures.

4.3 Low-Resource Results

Our experiments in low-resource scenarios show that E2H is particularly effective in situations where there is limited training data. As shown in Figure 2, by training on a fraction (1%, 5%, and 10%) of the original data[1], we observe that E2H-base significantly outperforms T5-base on all datasets. For example, when there is only 5% of the training data, E2H-base obtains an average of 7.1, 12.0, 6.4, and 8.2 absolute points of improvement over T5-base on ACE04-Ent, ACE05-Rel, ACE05-E, and Rest14 respectively. This highlights the effectiveness of our easy-to-hard learning framework when data is scarce. On one hand, the easy stage facilitates the model to identify the substructures of the target structure and capture the dependencies among them, which are difficult when there is limited data. On the other hand, the hard stage provides diverse and harder data to help the model tackle broad-range variations of the task, which is especially important in low-source scenarios.

Models	NER ACE04-Ent	RE ACE05-Rel	EE ACE05-E	ABSA Rest14
E2H-base	**86.24**	**65.44**	**50.98**	**75.40**
w/o Skill$_1$	85.91	64.28	50.85	74.33
w/o Skill$_2$	86.13	64.05	49.89	74.98
w/o Skill$_3$	-	63.74	-	75.14
w/o Skill$_4$	-	64.00	-	74.88

Table 5: Ablation results of E2H-base regarding different skills in the easy stage.

5 More Analysis

Analysis on different learning strategies In the main result table, we report the results of E2H trained with the easy→hard→main strategy, i.e., training the model in the easy, hard, and main stages sequentially. In this section, we investigate alternative learning strategies. Table 6 reports the results of T5-base models trained with different learning strategies on four datasets across four tasks. We have the following observations: (1) The easy→hard→main strategy is the best among the seven concerned strategies. It performs better than other strategies on all datasets. (2) Easy-to-hard multi-stage learning outperforms multi-task learning (i.e., easy+main+hard). When the easy, main, and hard parts of the training data are used, the easy→hard→main and easy→main→hard strategies show superiority over the easy+main+hard strategy on all datasets. This indicates that easy-to-hard multi-stage learning is essential to the model's performance. (3) Each stage is critical to our E2H framework. Removing any of the stages will reduce the performance of E2H. (4) In general, three-stage learning is better than two-stage learning, and they are better than one-stage learning.

[1] We repeat each experiment three times with different samples and report their averaged results.

Learning Strategy	**Type**	NER ACE04-Ent	RE ACE05-Rel	EE ACE05-E	ABSA Rest14	Avg
easy→hard→main	three-stage	**86.24**	**65.44**	**50.98**	**75.40**	**69.52**
easy→main→hard	three-stage	86.23	65.40	49.76	74.45	68.96
easy+main+hard	multi-task	86.10	64.46	49.16	73.94	68.42
easy→main	two-stage	85.93	63.85	50.31	74.52	68.65
hard→main	two-stage	85.99	64.41	49.26	74.67	68.58
easy→hard	two-stage	86.18	65.35	46.69	75.34	68.39
main	one-stage	85.60	62.91	49.68	72.11	67.58

Table 6: Experimental results of T5-base models trained with different learning strategies. The easy+main+hard strategy represents that the model is trained with the easy, main, and hard parts in a multi-task learning manner. The arrow → indicates the order between different stages.

Models	CoNLL03→ACE04-Ent	ACE04-Ent→CoNLL03
T5-base	19.54	17.45
E2H-base	**19.71**	**30.08**
Models	Rest16→Laptop14	Laptop14→Rest16
T5-base	42.37	60.50
E2H-base	**44.86**	**62.32**

Table 7: Cross-domain generalization performance of E2H-base and T5-base.

Is each skill necessary in the easy stage? To quantify the contribution of each skill, we examine the performance of E2H-base after removing a basic skill for training in the easy stage. Ablation results on four datasets across four tasks are shown in Table 5. Removing any skill degrades the performance of E2H on the main task, indicating that recognizing substructures and the dependency between them is crucial to the model's performance.

Does easy-to-hard learning improve the model's cross-domain generalization ability? To answer this question, we compare the performance of the E2H-base model and the T5-base model trained on a dataset on another dataset in a different domain of the same task. Table 7 reports the results of the cross-domain generalization performance of different models on two dataset pairs: CoNLL03↔ACE04-Ent of the NER task and Rest16↔Laptop14 of the ASTE task. E2H-base performs better than T5-base in all scenarios. This indicates that easy-to-hard learning can enhance the model's cross-domain generalization ability.

6 Related Work

IE is a long-standing research area in natural language processing. Over the years, the paradigm for IE has undergone several transitions. Early approaches to IE focus on sequence labeling techniques (McCallum and Li, 2003; Ma and Hovy, 2016; Zhang et al., 2018; Li et al., 2019; Zhang et al., 2021b), in which each word in a text is assigned a label indicating its role in the extraction task. Span-based approaches (Luan et al., 2019; Wang et al., 2020; Zhao et al., 2020; Xu et al., 2021a; Zhou et al., 2022, 2023), which involve identifying spans in the text that correspond to the desired information, are later introduced for IE. MRC-based methods (Du and Cardie, 2020; Li et al., 2020; Mao et al., 2021; Xu et al., 2023) that frame the extraction task as a reading comprehension problem and generation-based methods (Yan et al., 2021; Lu et al., 2021; Zhang et al., 2021c) that generate the extracted information directly from the text have gained popularity in recent years for IE. They have been shown to be more effective and flexible. Most of these methods target a specific IE task. There have been some efforts to develop unified IE methods (Paolini et al., 2021; Lu et al., 2022; Fei et al., 2022), which can unify various IE tasks with one framework. Our E2H framework, a unified IE framework, introduces a novel easy-to-hard learning paradigm for IE to reduce the gap between model and human learning.

From the perspective of improving the learning process, E2H shares similar spirits with transfer learning (Pan and Yang, 2010), which uses the knowledge gained from solving one task to help solve another related task. By comparison, E2H learns basic skills specifically designed to assist with the target task. E2H is also related to curriculum learning (Bengio et al., 2009; Wang et al., 2022) in its fundamental motivation of learning from easy to hard. Curriculum learning, inspired by the human learning process, presents examples starting from the easiest samples, then gradually introducing more complex ones. However, curriculum learning involves the intricate task of ordering instances based on their difficulty. This requires a reliable difficulty criterion or a ranking system, which can be challenging to define and often necessitates substantial human effort. In contrast, E2H emphasizes on mastering certain fundamental skills prior to tackling more intricate tasks, eliminating the requirement for a difficulty criterion. This approach can be particularly beneficial in scenarios where the target task requires a distinct set of skills, or when the learning setting does not naturally provide a straightforward measure of difficulty.

7 Conclusion

This paper proposes an easy-to-hard learning framework consisting of the easy stage, the hard stage, and the main stage for IE. Two novel strategies are proposed to build the easy and hard parts of the framework to enable the learning process. Experimental results in both full and low-resource scenarios demonstrate the effectiveness of our framework and its superiority over one-stage learning methods.

Limitations

While the results have shown the effectiveness of our framework in IE without using any additional resources, we did not explore the potential enhancement by utilizing existing resources in the easy-to-hard learning process. On one hand, we can build the easy stage with the help of existing data of simpler tasks. On the other hand, the data of harder tasks can be used for the hard stage. To enhance the E2H framework via effectively using existing resources is an interesting and promising direction. Another limitation is that we did not extensively explore the possible skill sets for each task. Exploring more approaches to obtain the skill sets is also open for future research. We plan to investigate these possibilities in our future work.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Lidong Bing, Sneha Chaudhari, Richard Wang, and William Cohen. 2015. Improving distant supervision for information extraction using label propagation through lists. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Lidong Bing, Wai Lam, and Tak-Lam Wong. 2013. Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, New York, NY, USA.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. 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 340–350, Online. Association for Computational Linguistics.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. 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 *Advances in Neural Information Processing Systems*.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

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

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022a. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, and Philip S. Yu. 2022b. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6714–6721.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

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.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2Path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225, Dublin, Ireland. Association for Computational Linguistics.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13543–13551.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191.

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. Ace 2004 multilingual training corpus.

Sergio Oramas, Luis Espinosa-Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. 2016. Information extraction for knowledge base construction in the music domain. *Data & Knowledge Engineering*, 106:70–83.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8600–8607. AAAI Press.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. 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. *Journal of Machine Learning Research*, 21(140):1–67.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Francis A. Ruambo and Mrindoko R. Nicholaus. 2019. Towards enhancing information retrieval systems: A brief survey of strategies and challenges. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–8.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8749–8757.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. 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 313–325, Hong Kong, China. Association for Computational Linguistics.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Online. Association for Computational Linguistics.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2022. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A unified label space for entity relation extraction. 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 220–231, Online. Association for Computational Linguistics.

Zihao Wang, Kwunping Lai, Piji Li, Lidong Bing, and Wai Lam. 2019. Tackling long-tailed relations and uncommon entities in knowledge graph completion. 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 250–260, Hong Kong, China. Association for Computational Linguistics.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021a. Learning span-level interactions for aspect sentiment triplet extraction. 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 4755–4766, Online. Association for Computational Linguistics.

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021b. Better feature integration for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3457–3469, Online. Association for Computational Linguistics.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Weiwen Xu, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. Peerda: Data augmentation via modeling peer relation for span identification tasks.

In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. 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.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022a. De-bias for generative extraction in unified NER task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 808–818, Dublin, Ireland. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Ruidan He, Haiyun Peng, Lidong Bing, and Wai Lam. 2021b. Cross-lingual aspect-based sentiment analysis with aspect term code-switching. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9220–9230, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021c. Towards generative 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 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022b. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20.

Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018. Learning tag dependencies for sequence tagging. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4581–4587. International Joint Conferences on Artificial Intelligence Organization.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3239–3248, Online. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Ran Zhou, Xin Li, Lidong Bing, Erik Cambria, and Chunyan Miao. 2023. Improving self-training for cross-lingual named entity recognition with contrastive and prototype learning. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.

Ran Zhou, Xin Li, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. ConNER: Consistency training for cross-lingual named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8438–8449, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

A.1 Statistics of Datasets

Statistics of datasets are reported in Table 8.

	#Train	#Val	#Test
CoNLL03	14,041	3,250	3,453
ACE04-Ent	6,202	745	812
ACE05-Ent	7,299	971	1,060
CoNLL04	922	231	288
ACE05-Rel	10,051	2,420	2,050
SciERC	1,861	275	551
ACE05-E	17,172	923	832
ACE05-E+	19,216	901	676
CASIE	11,189	1,778	3,208
Rest14	1,266	310	492
Laptop14	906	219	328
Rest15-ASTE	605	148	322
Rest16-ASTE	857	210	326
R-ACOS	1,530	171	583
L-ACOS	2,934	326	816
Rest15-ASQP	834	209	537
Rest16-ASQP	1,264	316	544

Table 8: Statistics of datasets.

A.2 Implementation Details

We set the maximum input length to 384 and the maximum target length to 256. Following the practices of Lu et al. (2022), we use a batch size of 64 for E2H-base and 32 for E2H-large. The learning rate is chosen from {1e-4, 3e-4} for E2H-base and {5e-5, 1e-4} for E2H-large, and we use the AdamW optimizer (Loshchilov and Hutter, 2019) with linear learning rate decay. The number of training epochs for the easy, hard, and main stages are set to [15, 30, 30] or [25, 50, 50], with the easy stage having fewer epochs as it typically has more data. For the hard stage, we choose M from {1, 2} for the datasets of the NER, RE, and EE tasks and from {1, 2, 3} for the datasets of the ABSA task. The parameters are chosen based on the model's performance on the development set. Generally, for large datasets such as ACE05-E, a smaller value of M like 1 is more appropriate, while for smaller datasets such as Laptop14, a larger value of M such as 3 is preferred. All experiments are conducted on NVIDIA Tesla A100.

A.3 Examples of IE tasks

Detailed examples of different IE tasks are shown in Tables 9-13. We use the structural extraction language proposed by Lu et al. (2022) to encode the target structure.

Task	Input	Target
NER	[HEC] [HES] [Ent] location [Ent] miscellaneous [Ent] organization [Ent] person [Text] Only France and Britain backed Fischler's proposal.	`((location: France) (location: Britain) (person: Fischler))`
Skill$_1$	[HEC] [Ent] location [Ent] miscellaneous [Ent] organization [Ent] person [Text] Only France and Britain backed Fischler's proposal.	`((location) (person))`
Skill$_2$	[HEC] [HES] [Ent] location [Text] Only France and Britain backed Fischler's proposal.	`((location: France) (location: Britain))`

Table 9: Detailed Examples for NER. We provide an instance for the main task and each skill. We highlight `Hint` in red, `Constraint` in brown, and `Schema` in blue. [HEC] and [HES] are the entity category hint and entity span hint, respectively. [Ent] is a special token to denote the entity category.

Task	Input	Target
RE	[HE] [HR] [Ent] generic [Ent] material [Ent] method [Ent] metric [Ent] other scientific term [Ent] task [Rel] compare [Rel] conjunction [Rel] evaluate for [Rel] feature of [Rel] hyponym of [Rel] part of [Rel] used for [Text] The demonstrator embodies an interesting combination of hand-built, symbolic resources and stochastic processes.	`((task: demonstrator) (material: hand-built, symbolic resources (part of: demonstrator)(conjunction: stochastic processes)) (method: stochastic processes (part of: demonstrator)))`
Skill$_1$	[HE] [Ent] generic [Ent] material [Ent] method [Ent] metric [Ent] other scientific term [Ent] task [Text] The demonstrator embodies an interesting combination of hand-built, symbolic resources and stochastic processes.	`((task: demonstrator) (material: hand-built, symbolic resources) (method: stochastic processes))`
Skill$_2$	[HE] [HR] [Ent] method: stochastic processes [Rel] compare [Rel] conjunction [Rel] evaluate for [Rel] feature of [Rel] hyponym of [Rel] part of [Rel] used for [Text] The demonstrator embodies an interesting combination of hand-built, symbolic resources and stochastic processes.	`((method: stochastic processes (part of: demonstrator)))`
Skill$_3$	[HR] [Rel] compare [Rel] conjunction [Rel] evaluate for [Rel] feature of [Rel] hyponym of [Rel] part of [Rel] used for [Text] The demonstrator embodies an interesting combination of hand-built, symbolic resources and stochastic processes.	`((part of) (conjunction))`
Skill$_4$	[HE] [HR] [Rel] conjunction [Ent] generic [Ent] material [Ent] method [Ent] metric [Ent] other scientific term [Ent] task [Text] The demonstrator embodies an interesting combination of hand-built, symbolic resources and stochastic processes.	`((material: hand-built, symbolic resources (conjunction: stochastic processes)))`

Table 10: Detailed Examples for RE. We provide an instance for the main task and each skill. We highlight `Hint` in red, `Constraint` in brown, and `Schema` in blue. [HE] and [HR] are the entity hint and relation hint, respectively. [Ent] and [Rel] are special tokens to denote the entity category and relation, respectively.

Task	Input	Target
EE	[HT] [HA] [Tri] acquit [Tri] appeal [Tri] arrest jail [Tri] attack [Tri] born [Tri] charge indict [Tri] convict [Tri] declare bankruptcy [Tri] demonstrate [Tri] die [Tri] divorce [Tri] elect [Tri] end organization [Tri] end position [Tri] execute [Tri] extradite [Tri] fine [Tri] injure [Tri] marry [Tri] meet [Tri] merge organization [Tri] nominate [Tri] pardon [Tri] phone write [Tri] release parole [Tri] sentence [Tri] start organization [Tri] start position [Tri] sue [Tri] transfer money [Tri] transfer ownership [Tri] transport [Tri] trial hearing [Arg] adjudicator [Arg] agent [Arg] artifact [Arg] attacker [Arg] beneficiary [Arg] buyer [Arg] defendant [Arg] destination [Arg] entity [Arg] giver [Arg] instrument [Arg] organization [Arg] origin [Arg] person [Arg] place [Arg] plaintiff [Arg] prosecutor [Arg] recipient [Arg] seller [Arg] target [Arg] vehicle [Arg] victim [Text] It was talking something about the war in Iraq. I guess it's a good thing about the elections that are going on.	`((attack: war (place: Iraq)) (elect: elections (place: Iraq)))`
Skill$_1$	[HT] [Tri] acquit [Tri] appeal [Tri] arrest jail [Tri] attack [Tri] born [Tri] charge indict [Tri] convict [Tri] declare bankruptcy [Tri] demonstrate [Tri] die [Tri] divorce [Tri] elect [Tri] end organization [Tri] end position [Tri] execute [Tri] extradite [Tri] fine [Tri] injure [Tri] marry [Tri] meet [Tri] merge organization [Tri] nominate [Tri] pardon [Tri] phone write [Tri] release parole [Tri] sentence [Tri] start organization [Tri] start position [Tri] sue [Tri] transfer money [Tri] transfer ownership [Tri] transport [Tri] trial hearing [Text] It was talking something about the war in Iraq. I guess it's a good thing about the elections that are going on.	`((attack: war) (elect: elections))`
Skill$_2$	[HT] [HA] [Tri] attack: war [Arg] adjudicator [Arg] agent [Arg] artifact [Arg] attacker [Arg] beneficiary [Arg] buyer [Arg] defendant [Arg] destination [Arg] entity [Arg] giver [Arg] instrument [Arg] organization [Arg] origin [Arg] person [Arg] place [Arg] plaintiff [Arg] prosecutor [Arg] recipient [Arg] seller [Arg] target [Arg] vehicle [Arg] victim [Text] It was talking something about the war in Iraq. I guess it's a good thing about the elections that are going on.	`((attack: war (place: Iraq)))`

Table 11: Detailed Examples for EE. We provide an instance for the main task and each skill. We highlight `Hint` in red, `Constraint` in brown, and `Schema` in blue. [HT] and [HA] are the event trigger hint and event argument hint, respectively. [Tri] and [Arg] are special tokens to denote the event category and argument category, respectively.

Task	Input	Target
ASTE	[HE] [HR] [Ent] aspect [Ent] opinion [Rel] negative [Rel] neutral [Rel] positive [Text] Great food but the service was dreadful!	((opinion: Great) (aspect: food (positive: Great)) (aspect: service (negative: dreadful)) (opinion: dreadful))
Skill$_1$	[HE] [Ent] aspect [Ent] opinion [Text] Great food but the service was dreadful!	((opinion: Great) (aspect: food) (aspect: service) (opinion: dreadful))
Skill$_2$	[HE] [HR] [Ent] aspect: sevice [Rel] negative [Rel] neutral [Rel] positive [Text] Great food but the service was dreadful!	((aspect: service (negative: dreadful)))
Skill$_3$	[HR] [Rel] negative [Rel] neutral [Rel] positive [Text] Great food but the service was dreadful!	((positive) (negative))
Skill$_4$	[HE] [HR] [Rel] positive [Ent] aspect [Ent] opinion [Text] Great food but the service was dreadful!	((aspect: food (positive: Great)))

Table 12: Detailed Examples for ASTE. We provide an instance for the main task and each skill. We highlight Hint in red, Constraint in brown, and Schema in blue. Following Lu et al. (2022), we formulate ASTE as the RE task, where aspect terms and opinion terms are entities, and sentiment polarities are relations. [HE] and [HR] are the entity hint and relation hint, respectively. [Ent] and [Rel] are special tokens to denote the entity category and relation, respectively.

Task	Input	Target
ASQP	[HC] [HA] [Cat] category [Arg] aspect [Arg] opinion [Arg] polarity [Text] The pizza is delicious.	((category: food quality (aspect: pizza) (opinion: delicious) (polarity: positive))
Skill$_1$	[HC] [Cat] category [Text] The pizza is delicious.	((category: food quality))
Skill$_2$	[HC] [HA] [Cat] category [Arg] aspect [Text] The pizza is delicious.	((category: food quality (aspect: pizza))
Skill$_3$	[HC] [HA] [Cat] category [Arg] opinion [Text] The pizza is delicious.	((category: food quality (opinion: delicious))
Skill$_4$	[HC] [HA] [Cat] category [Arg] polarity [Text] The pizza is delicious.	((category: food quality (polarity: positive))

Table 13: Detailed Examples for ASQP. We provide an instance for the main task and each skill. We highlight Hint in red, Constraint in brown, and Schema in blue. We treat the aspect term, opinion term, and sentiment polarity as the arguments of the aspect category. [HC] and [HA] are the aspect category hint and argument hint, respectively. [Cat] and [Arg] are special tokens to denote the aspect category and its arguments, respectively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The last section "Limitation"

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 0 "Abstract" and Section 1 "Introduction"

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4 "Experiments"

☑ B1. Did you cite the creators of artifacts you used?
Section 4 "Experiments"

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 "Experiments"

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A.1 "Statistics of Datasets"

C ☑ Did you run computational experiments?

Section 4 "Experiments"

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 "Experiments" and Appendix A.2 "Implementation Details"

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4 "Experiments" and Appendix A.2 "Implementation Details"

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 "Experiments"

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.