DEEPSTRUCT: Pretraining of Language Models for Structure Prediction

Chenguang Wang[*†], **Xiao Liu**[¶†], **Zui Chen**[¶†], **Haoyun Hong**[¶], **Jie Tang**[¶], **Dawn Song**[*]
[*]UC Berkeley, [¶]Tsinghua University
{chenguangwang,dawnsong}@berkeley.edu, jietang@tsinghua.edu.cn
{liuxiao21,chenzui19,honghy17}@mails.tsinghua.edu.cn

Abstract

We introduce a method for improving the structural understanding abilities of language models. Unlike previous approaches that finetune the models with task-specific augmentation, we pretrain language models on a collection of task-agnostic corpora to generate structures from text. Our structure pretraining enables zero-shot transfer of the learned knowledge that models have about the structure tasks. We study the performance of this approach on 28 datasets, spanning 10 structure prediction tasks including open information extraction, joint entity and relation extraction, named entity recognition, relation classification, semantic role labeling, event extraction, coreference resolution, factual probe, intent detection, and dialogue state tracking. We further enhance the pretraining with the task-specific training sets. We show that a 10B parameter language model transfers non-trivially to most tasks and obtains state-of-the-art performance on 21 of 28 datasets that we evaluate.[1]

1 Introduction

Pretrained language models (LMs) have revolutionized NLP over the last few years (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019b), increasingly adept in performing the flexible and task-agnostic downstream transfer. Their transfer performance is less studied in structure prediction tasks, however. Well-studied tasks mainly focus on understanding one particular aspect of the text, such as predicting the next word that comes after as in language modeling. Unlike those downstream tasks, structure prediction requires the structural understanding of the text for further integrating multiple relevant aspects into a structure. For instance, a typical structure prediction task, called open information extraction, seeks the entire structural information in a sentence (Figure 2). Different from traditional NLP tasks, structure prediction takes one step further and serves as a natural testbed for the structural understanding competence of LMs.

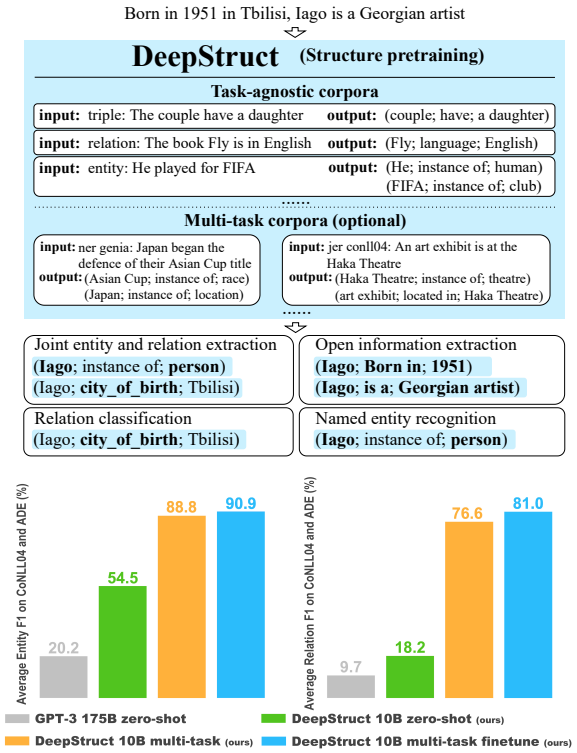

Figure 1: Summary of our approach and results. Upper: an overview of DEEPSTRUCT and the proposed structure pretraining. Lower: performance of our 10B DEEPSTRUCT zero-shot and multi-task, compared with 175B GPT-3 zero-shot.

It is non-trivial to transfer LMs to downstream structure prediction tasks. While the structure prediction requires structural understanding, the LMs are pretrained to understand an independent aspect. For example, GPT-3 (Brown et al., 2020) is trained to predict the next word, and BERT (Devlin et al., 2019) is trained to recover the masked tokens. Recent work has made efforts in bridging the gap in transferring pretrained models to structure prediction tasks with a focus on two directions. As shown in Figure 3, first, task-specific architectures are proposed to model the structures for different

[†] Equal contribution.
[1] The code and datasets are available at https://github.com/cgraywang/deepstruct.

Findings of the Association for Computational Linguistics: ACL 2022, pages 803 - 823
May 22-27, 2022 ©2022 Association for Computational Linguistics

structure prediction tasks (Stanovsky et al., 2018; Soares et al., 2019). Second, task-specific data augmentation (Paolini et al., 2021; Wang et al., 2021; Wei et al., 2021) is introduced, aiming to enrich text format with structure information. These approaches involve custom-designed task augmentations, impeding their usability in general structure prediction tasks.

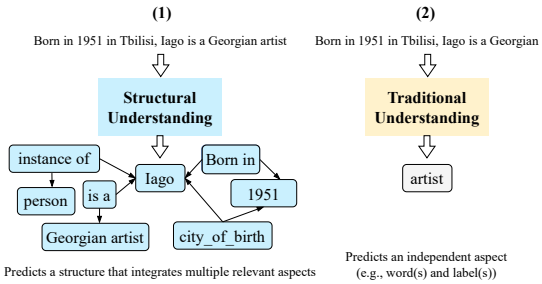

Figure 2: Comparison between structural understanding and traditional understanding of text.

In this paper, we improve the structural understanding capabilities of LMs. In contrast to previous approaches relying on task augmentations, we introduce structure pretraining, which systematically teaches LMs to better understand structures of text beyond independent aspects in a pretraining phase (Figure 1). This enables the zero-shot transfer of knowledge that LMs learned about structures during our pretraining to downstream structure prediction tasks. For example, our zero-shot 10B parameter LM significantly outperforms the zero-shot GPT-3 (175B) on a structure prediction benchmark dataset (Figure 1). We accomplish this by reformulating structure prediction as a series of unit tasks–triple prediction tasks. We then train LMs on a collection of task-agnostic structural corpora to generate triples from text. The design of triple representation is important: it unifies a wide set of standard structure prediction tasks into the same task format. We apply our pretrained model DEEPSTRUCT to 27 datasets spanning 10 structure prediction tasks, including open information extraction, joint entity and relation extraction, named entity recognition, relation classification, semantic role labeling, event extraction, coreference resolution, factual probe, intent detection, and dialogue state tracking. We further enhance the pretraining with multiple downstream structure prediction training sets and obtain state-of-the-art performance on 20 of 27 datasets. Our contributions are as follows:

- We improve structural understanding abilities of pretrained LMs. Compared to traditional NLP tasks that only consider the understanding of an independent aspect of the text, structural understanding takes a step further that requires the ability to integrate multiple relevant aspects into a structure. We argue that it is important for LMs to go beyond traditional understanding towards structural understanding, as it requires a higher level of intelligent competence and is more challenging. It can also benefit a wide spectrum of NLP tasks that require structure-level understanding capability.
- We propose structure pretraining, which further pretrains the LMs to understand structures in the text. The basic intuition is that the standard pretraining helps LMs to understand individual aspects of the information in the text, our method learns to integrate those individual aspects into structures. Compared to existing approaches, this method enables the zero-shot transfer of LMs to structure prediction tasks. For instance, our 10B LM produces superior zero-shot performance compared to 175B GPT-3 on a representative structure prediction task.
- We further equip our pretraining with multi-task learning and apply our method to 27 structure prediction datasets across 10 tasks. We achieve state-of-the-art performance on 20 of 27 datasets that we evaluate. We hope this can help facilitate the structural understanding research in the NLP community.

2 Structure Pretraining

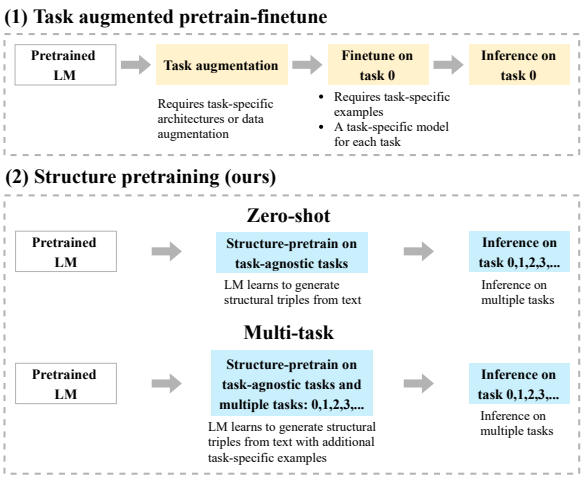

Figure 3: Comparing structure pretraining with standard pretrain-finetune paradigm.

The goal of our method is to improve the structural understanding capabilities of language models (LMs), i.e., understanding the structures of text. As shown in Figure 3, instead of using the standard pretrain-finetune paradigm for each task, we introduce structure pretraining that aims to teach LMs to correspond to structures in a wide spectrum of tasks at the same time. We evaluate their structural understanding ability on multiple structure prediction tasks.

2.1 Generative Pretraining

While the LM is pretrained to understand a single aspect of the text, structural understanding aims to recover the entire structure in the text (Figure 2). Structure pretraining is designed to bridge the gap via guiding LMs to produce structures from the text. It is ideal to generate arbitrary structures as needed. However, this is infeasible due to the highly complex nature of such structures.

As an alternative, we reformulate the structure prediction as a combination of triple generation tasks. We refer to a triple as *(head entity; relation; tail entity)* describing relations between entities. We design three pretraining tasks with a focus on predicting the entities, relations, and triples respectively. As shown in Figure 1, (i) Entity prediction aims to output triples regarding the entities and their types in an input sentence. We implement this via prepending "`entity:`" as a prefix in the input. (ii) Relation prediction aims to recover the relations and corresponding types in the input as a triple. Similarly, we add "`relation`" followed by a task separator "`:`" to each input. (iii) Triple prediction outputs the entire triple structure from the input. We attach "`triple:`" to indicate this task. These pretraining tasks are task-agnostic to downstream tasks, enabling the zero-shot downstream transfer (Sec. 2.3).

Although the triple formulation is straightforward, we find that it is very flexible and able to model all structure prediction tasks we consider. A structure prediction task can be generally decomposed into generating the entities, relations, or triples. For example, named entity recognition predicts the entities and their types. It can be naturally represented as an entity prediction problem. Besides, traditional structure prediction tasks focusing on relations (e.g., relation classification) or triples (e.g., open information extraction) can be formulated as relation or triple prediction task respectively. A summary of all downstream tasks is described in Sec. 2.2.

Dataset	#Sent.	#Ent.	#Rel. (#Tri.)	Task
T-REx (ElSahar et al., 2018)	6.2M	8.8M	11M	entity, relation
TEKGEN (Agarwal et al., 2021)	18M	23.5M	45M	entity, relation
KELM (Agarwal et al., 2021)	15.7M	54.5M	35.7M	entity, relation
WebNLG (Gardent et al., 2017)	88K	348K	261K	relation
ConceptNet (Speer and Havasi, 2012)	610K	3.1M	610k	relation
OPIEC (Gashteovski et al., 2019)	10.7M	43.0M	21.5M	triple

Table 1: Pretraining dataset statistics.

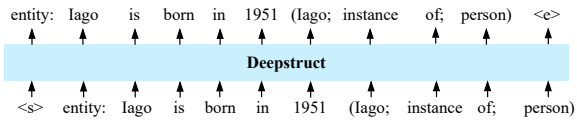

Figure 4: Summary of training procedure.

We frame the pretraining as a conditional generation task where the input corresponds to text x, and the output y is a sequence of triples. Our pretraining can be expressed as estimating a conditional distribution $p(y|x)$ in a probabilistic framework. We use an autoregressive LM to model $p(y|x)$.

Pretraining Data We train the model on a collection of task-agnostic corpora including prebuilt large-scale alignments between text and triples. In particular, we use T-REx (ElSahar et al., 2018), TEKGEN and KELM (Agarwal et al., 2021), WebNLG (Gardent et al., 2017), ConceptNet (Speer and Havasi, 2012). These corpora align text to triples consisting of high-quality entities and relations in knowledge graphs (e.g., Wikidata), which are used for entity and relation prediction tasks. In addition, for triple prediction tasks, we use OPIEC (Gashteovski et al., 2019) that provides open schema triples. The pretraining data statistics and the corresponding pretraining tasks are shown in Table 1.

Figure 4 shows an example of the training procedure for the entity prediction task based on the input/output sample below.

> **Input** `entity`: Iago is born in 1951
> **Output** (**Iago**; instance of; **person**)

where the input text and output triple are aligned, provided by our pretraining data. Tokens are predicted autoregressively starting with `<s>` token and ending with `<e>` token. The head entity (i.e., Iago) and the tail entity (i.e., person) of the output triple then serve as the predictions of named entity recognition.

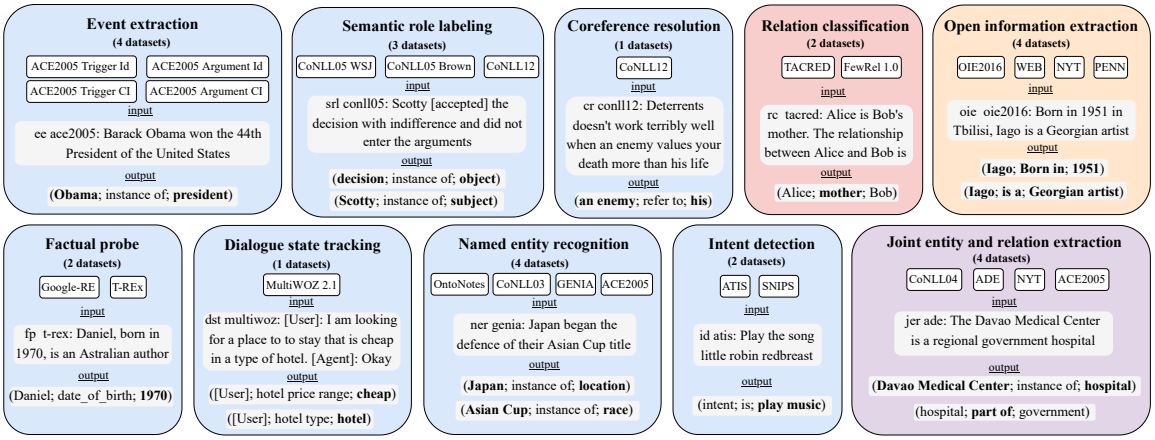

Figure 5: Summary of tasks and datasets. Blue: entity prediction task; Red: relation prediction task; Purple: entity and relation prediction task; Yellow: triple prediction task.

2.2 Tasks

It is resource-intensive to create large-scale structural understanding datasets from scratch. Therefore, we collect existing datasets in the field of structure prediction for the evaluation. We consider 27 datasets spanning across 10 structure prediction tasks as shown in Figure 5. Detailed descriptions and sizes are shown in Appendix A.

2.3 Zero-Shot

The zero-shot DEEPSTRUCT refers to the setting where the pretrained model is used without any task-specific training at inference time. This differs from prior fully supervised methods. This setting is challenging as it might be difficult for humans to understand the tasks without prior examples. For example, if we are asked about "semantic role labeling" that aims to recover the predicate-argument structure, it is hard to understand what this really means. Nevertheless, for at least some settings, zero-shot is closest to how humans perform tasks. For example, for named entity recognition, a human would likely know what to do.

We enable the zero-shot transfer to structure prediction tasks via converting the downstream tasks to one or a combination of the pretraining tasks. As shown in Figure 5, at inference time, seven structure prediction tasks are formulated as entity prediction with the prefix "`entity:`" attached to the input example (in blue), while one task is cast as relation prediction with the prefix "`relation:`" (in red). In addition, open information extraction is a triple prediction task with the prefix "`triple:`" (in yellow), and joint entity and relation extraction (JER) is a combination of entity and relation prediction (in purple). For the entity and relation prediction, we use the prefix "`entity:`" and "`relation:`" respectivley. Besides, for each dataset, we build a schema alignment between pretraining and downstream dataset (details are described in Sec. 5). The output triples are then decoded as corresponding structure predictions based on the pre-built schema alignment.

2.4 Multi-Task

However, the distribution of the pretraining data is not perfectly aligned with the distribution of downstream datasets. This results in a shift in the output distribution of the model. The zero-shot setting cannot perform at its best on several out-of-distribution tasks including dialogue state tracking. The reason is that its desired output is a dialogue state, which is lacking in our task-agnostic pretraining corpora. To mitigate this, we integrate multiple structure prediction datasets into the pretraining corpora, and train our method on the mixture of the datasets. We list an example input and output format for each task in Figure 5. For all datasets of a particular task, we adopt the same input and output format. We also add task name and dataset name followed by the separator "`:`" as a prefix to each input example. For example, we add "`jer ade:`" to indicate one of the JER datasets, ADE. More examples of each task and dataset are shown in Table 16. In contrast to fully pretrain-finetuned models that store a copy of parameters for each task, this setting is a lightweight alternative and produces a single model for all tasks, improving parameter sharing.

After multi-task training, for each task, we further finetune our method on the task-specific dataset. The intuition is that finetuning is the de facto way to leverage pretrained LMs to perform

downstream tasks. We aim to test an upper bound of the transfer performance of our structure pretraining via the additional finetuning phase. For both multi-task settings, we use the same data format with the training at test time. Basically, we add the task name and dataset name followed by the separator to the input example.

3 Experiments

In this section, we show that DEEPSTRUCT successfully transfers to the structure prediction tasks considered and obtain state-of-the-art results on 21 of 28 datasets we evaluate. All results are obtained via structure pretraining a pretrained 10B parameter LM, GLM (Du et al., 2021). The details of the experimental setup, datasets, and comparison methods are described in Appendix A.

3.1 Main Results

We have two settings as described in Sec. 2: zero-shot and multi-task. We also finetune the multi-task version on each downstream dataset. In total, we have three versions of DEEPSTRUCT (Cf. Table 2). For comparison: we report the performance of TANL (Paolini et al., 2021) when available. We also show the best performance among the task-specific models that are described in Appendix A.

With the zero-shot setting, a single model is used to perform on multiple tasks without the need of any task-specific training. This is in contrast to previous approaches that rely on task-specific models and datasets for each task. In Table 3, we also report the zero-shot performance of GPT-3 175B (Brown et al., 2020) on CoNLL04 and ADE (JER) via formulating the task as a question answering problem via prompting (details of the formulation are described in Sec. 5). JER requires the model to extract a set of entities and a set of relations between pairs of entities from the input text. Each predicted entity or relation has to be also assigned to an entity or a relation type. Zero-shot DEEPSTRUCT 10B outperforms zero-shot GPT-3 175B by a large margin without any prompt engineering. As shown in Table 2, overall, DEEPSTRUCT's zero-shot performance is still far from that of task-specific supervised models on most tasks. The only exception is that the zero-shot setting obtains the new state-of-the-art performance on the factual probe with averaging 20% P@1 improvement. This is because the task-specific method is also zero-shot. Note that we have removed the overlapped sentences with the T-REx test sets (factual probe) from our pretraining data. The result indicates that the structure pretraining is able to adapt the LM knowledge to the tasks by making LM aware of symbolic knowledge in the pretraining corpora. Besides, the zero-shot approach generalizes well to all task-agnostic pretraining tasks including entity prediction (e.g., named entity recognition), relation prediction (e.g., relation classification), and triple prediction (e.g., open information extraction).

Similar to the zero-shot setup, we only train a single model to conduct all the downstream tasks under the multi-task setting. This is different from the supervised models that use task-specific models and parameters. We achieve state-of-the-art performance on three datasets. For the other datasets, we obtain a competitive performance within a few points of the best-compared methods. Notably, most structure prediction tasks show a large gain from zero-shot to multi-task. This suggests that most tasks are out-of-distribution of our structure pretrained model. Nevertheless, our method appears to be able to adapt to the downstream distributions, recovering strong performance in the multi-task setting. Another explanation is that multi-task examples help the model better understand the downstream tasks, such as the output format of each task. We also observe strong multi-task performance on FewRel, which is a low-resource structure prediction benchmark. This suggests that the multi-task setting is beneficial in low-resource regimes via transferring knowledge from similar tasks. For our multi-task training, we leave out the ACE2005 named entity recognition dataset due to the overlap between train and test splits for different tasks. After finetuning, we obtain state-of-the-art performance on 21 datasets. For instance, we obtain a +8.0 absolute improvement and a +2.9 absolute improvement on CoNLL05 Brown (semantic role labeling) and TACRED (relation classification) compared to the state-of-the-art methods.

All above results are based on a pretrained 10B parameter LM, GLM. GLM is an autoregressive LM. In addition, the context x is encoded by a bidirectional encoder. In principle, generative LMs, such as T5 (Raffel et al., 2019), BART (Lewis et al., 2020) and GPT-3 (Brown et al., 2020), can also be used with the proposed structure pretraining for the structure prediction tasks as well. We leave this as one of the future investigations.

Task	Dataset		Metric	Task-specific model (w/o extra data)	TANL	DEEPSTRUCT		
						zero-shot	multi-task w/ finetune	
Open information extraction	OIE2016		F1	67.0 (Stanovsky et al., 2018)	-	28.1	71.2	**71.3**
	WEB			**58.9** (Stanovsky et al., 2016)	-	43.8	50.8	49.1
	NYT			38.3 (Saha and Mausam, 2018)	-	28.9	43.6	**45.0**
	PENN			42.6 (OpenIE4 [3])	-	51.0	**54.5**	45.1
Relation classification	TACRED		F1	73.9 (Sainz et al., 2021)	71.9	36.1	74.9	**76.8**
	FewRel 1.0	5-way 1-shot		90.1 (Soares et al., 2019)	93.6±5.4	72.4±6.9	93.6±6.0	**98.4±2.8**
		5-way 5-shot		89.5 (Gao et al., 2019)	97.6±3.2	70.8±8.0	96.4±4.2	**100.0±0.0**
		10-way 1-shot		83.4 (Soares et al., 2019)	82.2±5.1	67.6±4.5	92.2±6.4	**97.8±2.0**
		10-way 5-shot		81.8 (Gao et al., 2019)	89.8±3.6	66.4±6.3	94.6±3.6	**99.8±0.6**
Joint entity and relation extraction	CoNLL04		F1 ($\frac{\text{Ent.}}{\text{Rel.}}$)	88.9 / 71.9 (Zhao et al., 2020)	90.3 / 71.4	48.3 / 25.8	87.4 / 69.6	**90.7** / **78.3**
	ADE			89.3 / 78.8 (Eberts and Ulges, 2020)	**91.2** / **83.8**	60.7 / 10.6	90.2 / 83.7	91.1 / **83.8**
	NYT			- / 84.6 (Yuan et al., 2020)	94.9 / 90.8	60.5 / 28.6	95.4 / **93.9**	**95.9** / 93.3
	ACE2005			88.4 / 63.2 (Luan et al., 2019)	88.9 / 63.7	31.8 / 5.3	87.8 / 54.0	**90.0** / **66.8**
Event extraction	ACE2005	Trigger Id	F1	72.5 (Nguyen and Nguyen, 2019)	72.9	-	71.7	**73.5**
		Trigger Cl		**69.8** (Nguyen and Nguyen, 2019)	68.5	-	67.9	**69.8**
		Argument Id		**59.9** (Nguyen and Nguyen, 2019)	50.1	-	54.9	59.4
		Argument Cl		52.5 (Wadden et al., 2019)	48.5	-	52.7	**56.2**
Coreference resolution	CoNLL12		MUC	**86.3**	81.0	-	63.9	74.9
			B^3	**77.6** (Wu et al., 2020)	69.0	-	57.7	71.3
			$CEAF_{\phi 4}$	**75.8**	68.4	-	60.2	73.1
			Ave. F1	**79.9**	72.8	-	60.6	73.1
Intent detection	ATIS		F1	**97.8** / 97.4 (E et al., 2019)	97.6	-	66.6	**97.8**
	SNIPS				**98.7**	-	78.4	97.3
Semantic role labeling	CoNLL05 WSJ		F1	88.8	89.3	-	**95.6**	95.2
	CoNLL05 Brown			82.0 (Shi and Lin, 2019)	84.1	-	92.0	**92.1**
	CoNLL12			86.5	87.7	-	**97.6**	96.0
Named entity recognition	CoNLL03		F1	**93.5** (Yu et al., 2020b)	91.7	44.4	93.1	93.0
	OntoNotes			**90.4** (Yan et al., 2021)	89.9	2.5	87.6	87.8
	GENIA			80.5 (Yu et al., 2020b)	76.4	47.2	80.2	**80.8**
	ACE2005			**86.9** (Li et al., 2020a)	84.9	28.1	-	**86.9**
Dialogue state tracking	MultiWOZ 2.1		Joint Acc.	**55.7** (Hosseini-Asl et al., 2020)	51.4	-	53.5	54.2
Factual probe	Google-RE		P@1	78.0 (Petroni et al., 2020)	-	**97.9**	90.3	-
	T-REx			62.6 (Petroni et al., 2020)	-	**85.0**	71.0	-

Table 2: Results on all tasks. All evaluation scores are higher the better. TANL is introduced in (Paolini et al., 2021). The **bold** denotes the best, and the underline indicates the second best. Detailed results are included in Appendix A.

Model	CoNLL04		ADE	
	Ent.	Rel.	Ent.	Rel.
GPT-3 175B zero-shot	34.7	18.1	5.8	1.3
DEEPSTRUCT zero-shot	48.3	25.8	60.7	10.6
DEEPSTRUCT multi-task	87.4	69.6	90.2	83.7
DEEPSTRUCT w/ finetune	90.7	78.3	91.1	83.8

Table 3: Compare DEEPSTRUCT to GPT-3 (Brown et al., 2020) 175B zero-shot on CoNLL04 and ADE datasets (joint entity and relation extraction). Ent. and Rel. denote entity F1 and relation F1 respectively.

3.2 Ablation Studies

Pretraining Strategies As the key question of our work is to investigate how structure pretraining improves the structural understanding ability of LMs, we examine how different pretraining strategies impact the downstream performance. We evaluate the below settings on the CoNLL04 (JER). The first two settings examine the relative importance of the pretraining data: (i) With example-proportional mixing: We follow (Raffel et al., 2019) with a mixing rate maximum of 10K to balance the different sizes of datasets. All other components are kept the same with DEEPSTRUCT multi-task with finetuning. (ii) With entity and relation augmentation: We add special tokens "[ ]" to indicate the positions of the entities and relations in a sentence. Additional details are shown in Appendix A.5. (iii) No pretrain, finetune: We remove structure pretraining, and only finetune the LM on CoNLL04. (iv) Zero-shot: We only use the task-agnostic datasets and exclude the multi-task datasets in the pretraining. (v) Multi-task: We use the multi-task model without finetuning. (iv) and (v) are the same with the zero-shot and multi-task settings in Sec. 2. (vi) Finetune: The multiple downstream datasets are excluded in the structure pretraining, but the model is finetuned on CoNLL04.

Method	Ent.	Rel.
DEEPSTRUCT 220M multi-task finetune	90.7	75.7
With example-proportional mixing	88.0	73.1
With entity and relation augmentation	88.6	74.9
No pretrain 220M, finetune	84.7	63.5
DEEPSTRUCT 220M zero-shot	51.5	22.9
DEEPSTRUCT 220M multi-task	76.9	55.2
DEEPSTRUCT 220M finetune	87.4	70.4

Table 4: Ablation over different facets of structure pretraining on CoNLL04 test set (joint entity and relation extraction). Ent. and Rel. indicate entity F1 and relation F1 respectively.

Table 4 shows the results. First, the distribution of pretraining data does not significantly shift from that of most tasks. This limits the impact of the balanced strategy ((i)). The data augmentation ((ii)) does not bring additional benefits to the downstream performance. This confirms that the key to the success of structure prediction is our formulation that narrows down a complex structure to a set of triple prediction tasks. This allows the pretraining to capture the entities and relations that are important for tasks. Second, removing the structure pretraining ((iii)) provides the most direct ablation of how much structure pretraining helps. Structure pretraining significantly improves the LM in structure prediction. This is due to the gap between LM pretraining and downstream structural understanding. For example, the distribution of structure prediction datasets is different from or is considered as out-of-distribution for the pretraining data. Structure pretraining improves the adaptation to those datasets. Next, similar to the findings in Table 2, we find that both task-agnostic training sets ((iv)) and multi-tasks datasets ((v)) contribute to the strength of structure pretraining. In particular, finetuning is still very important to improve the downstream performance (IV et al., 2021). However, it produces a task-specific model for each dataset instead of a unified model for all tasks as in our zero-shot or multi-task setup. Compared to only finetuning the model on a downstream dataset ((vi)), the multi-task setting obtains sizable improvements. This is because if the downstream dataset sizes are small such as of CoNLL04, multi-task learning can be extremely helpful in the low-resource regimes (Paolini et al., 2021). We conduct the above ablation studies using a base version of DEEPSTRUCT with 220M parameters.

Scaling Laws As it is often the case that larger models substantially improve the transferring capabilities of LMs (Brown et al., 2020; Wei et al., 2021), we explore how model scaling benefits the structure pretraining. We evaluate the effect on models with 110M, 220M, 2B, 10B parameters on JER datasets with multi-task and multi-task finetuned DEEPSTRUCT (Cf. Figure 6).

As expected, average performance across the datasets improves as models grow larger. We find that when the models reach the order of 10B parameters, structure pretraining obtains the best performance. The 10B parameter model significantly improves the results compared to the 110M parameter model. One reason is that for small-scale models, learning across 28 structure prediction datasets during the structure pretraining may exceed the model capacity. For larger models, structure pretraining fully utilizes the model capacity and also teaches the models to generate triples according to the downstream tasks, allowing them to generalize well to most tasks with the rest capacity. It is also interesting that the performance does not seem to significantly saturate, indicating that the performance may further improve with larger-scale models. Under both setups, we observe similar trends. We also see that the model size matters more to the multi-task setting than to the finetuned version, suggesting finetuning is able to specifically adapt to a task given a limited model size. The main pitfall is its generalization to more tasks.

4 Related Work

Pretrained LMs (Devlin et al., 2019; Radford et al., 2019b; Yang et al., 2019) are the key ingredients in contemporary NLP. Sequence-to-sequence (seq2seq) LMs target conditional generation, such as T5 (Raffel et al., 2019), BART (Lewis et al., 2020) and GLM (Du et al., 2021). These models have benefited a wide range of nature language generation tasks such as summarization (Zhang et al., 2020) and text infilling (Zhu et al., 2019; Shen et al., 2020). Recent attempts of generative prediction (Paolini et al., 2021; Schick and Schütze, 2021; Lester et al., 2021) have found that seq2seq models are able to provide a unified solution for modeling a wide set of NLP tasks. While existing approaches focus on text-to-text generation, DEEPSTRUCT aims to perform text-to-triple generation.

Multi-task learning (Caruana, 1997) aims to train a model for multiple tasks simultaneously. For deep learning, it is usually categorized into hard weight sharing and soft weight constraint (Ruder, 2017). In the context of NLP, weight sharing has

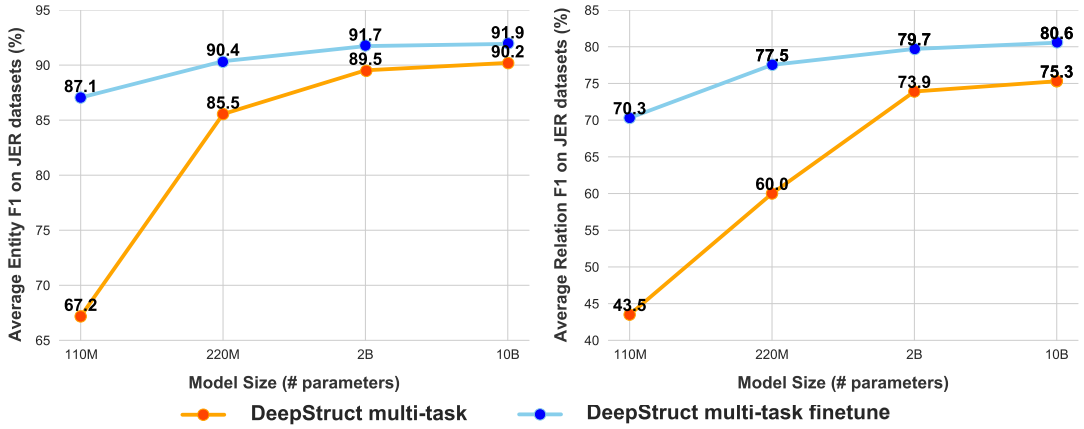

Figure 6: Model scaling results on joint entity and relation extraction (JER) datasets. Left: entity F1; Right: relation F1.

been adopted in (Collobert and Weston, 2008; Yang et al., 2016; Liu et al., 2020). Since the emerging of large pretrained LMs (Radford et al., 2019a; Devlin et al., 2019; Yang et al., 2019), multi-task training has been shown effective to enhance LMs' transferability to downstream tasks (Raffel et al., 2019). Recent studies (Wei et al., 2021) also show that pretrained models finetuned with abundant downstream tasks can conduct effective zero-shot learning. The main difference is that DEEPSTRUCT trains across multiple structure prediction datasets in structure pretraining with task-agnostic corpora, where we cast all datasets into triple formats.

Structure prediction is a long-standing challenge that relates to many NLP applications such as open information extraction (Gashteovski et al., 2019), named entity recognition (Sang and Meulder, 2003; Weischedel et al., 2013), and relation classification (Zhang et al., 2017; Han et al., 2018; Gao et al., 2019). To handle different structure prediction problems, prior work present a variety of task-specific models in the form of sequence tagging (Stanovsky et al., 2018; Li et al., 2019), machine reading comprehension (Zhao et al., 2020) and text classification (Soares et al., 2019), which hinders the knowledge transfer across different tasks. TANL (Paolini et al., 2021) proposes a translation-based approach to unify different structure prediction tasks with task-specific data augmentation. By contrast, our DEEPSTRUCT unifies more structure prediction tasks via a single model and an uniform data format.

5 Discussion

Related Models Recent studies have provided unified solutions for structural prediction tasks. We focus on the comparison between our DEEPSTRUCT to the state-of-the-art TANL (Paolini et al., 2021) and DeepEx (Wang et al., 2021). TANL (Paolini et al., 2021) proposes task-specific data augmentation (i.e., augmented natural language) that annotates task information and predictions in the input and output respectively for each structure prediction task. The main difference is that DEEPSTRUCT decomposes the structure prediction tasks into a collection of triple generation tasks. The triple format serves as the unified representation for all considered structure prediction tasks without the need of introducing new data augmentation as in TANL. While TANL mainly works in the multi-task setting, we additionally enable the zero-shot transfer via the task-agnostic structure pretraining. DeepEx (Wang et al., 2021) explores the attention matrices of pretrained LMs via beam search to generate triples for information extraction tasks. Following the search, DeepEx introduces an extra ranking stage to improve the quality of the triples. Differently, DEEPSTRUCT aims to generate the triples for a wide set structure prediction tasks in an end-to-end fashion thanks to the proposed structure pretraining.

Besides, both TANL and DeepEx explore relatively small-scale pretrained LMs. Instead, DEEPSTRUCT scales up to 10 billion parameters. Figure 6 shows that the performance improvements follow the scaling law (Raffel et al., 2019; Lester et al., 2021; Wei et al., 2021; Sanh et al., 2021; Liu et al., 2021). Based on our results, DEEPSTRUCT generalizes better to more structure prediction tasks compared to TANL and DeepEx.

Zero-Shot Setup For our zero-shot setup, we follow the zero-shot usage in recent pretrained LM studies (Brown et al., 2020; Wei et al., 2021; Sanh et al., 2021). It refers to the setting where a pre-

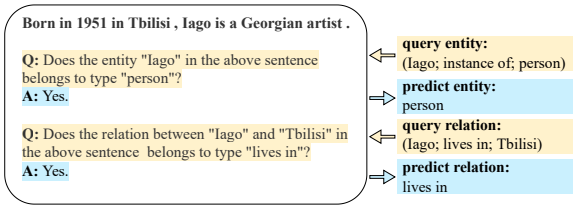

Figure 7: An example of GPT-3 zero-shot setting. To predict entities, we convert the gold entity triple (Iago; instance of; person) to an entity based true-or-false question. Similarly, to predict relations, the gold relation triple (Iago; lives in; Tbilisi) is turned into a relation based true-or-false question. The task predictions are correct if the answers are "yes".

trained model is used to directly perform downstream tasks without including downstream training sets in its own pretraining data. For DEEPSTRUCT our pretraining data is task-agnostic. For each task, we build an offline alignment between the schema of the pretraining data and the task dataset based on co-occurrence information in the pretraining data and downstream data (Angeli et al., 2015). We then manually curate the alignment. The resulting schema alignment is part of our release [1]. At test time, we convert each task to one or a combination of the pretraining tasks based on Figure 5: entity, relation, and triple prediction. After producing the triples, we use the pre-built schema alignment to obtain the task predictions.

For GPT-3 zero-shot setting, we follow the prompting method in the GPT-3 paper (Brown et al., 2020). In more detail, we aim to test the upper bound performance of GPT-3 for structure prediction, in particular the JER task. Therefore, instead of using standard prompts in the form of question answering, we design the prompts for "true-or-false" questions based on the ground truth. In this case, GPT-3 only answers with "yes" or "no" to produce a task prediction (Cf. Figure 7).

6 Conclusion

We improve structural understanding capabilities of language models. We evaluate it on a wide set of structure prediction tasks including 10 tasks and 28 datasets, and successfully transfer pretrained language models to them through the proposed structure pretraining, which teaches language models to output triples from text. We enable both zero-shot and multi-task transfer learning. DEEPSTRUCT obtains state-of-the-art results on 21 of 28 datasets. The result shows that pretrained language models can master higher-level understanding (e.g., structural understanding), which may benefit more NLP tasks. We hope it will foster future research along the language structural understanding direction.

7 Ethical Considerations

We hereby acknowledge that all of the co-authors of this work are aware of the provided *ACM Code of Ethics* and honor the code of conduct. This work is mainly about the pretraining and multi-task learning of LMs for structural prediction. The followings give the aspects of both our ethical considerations and our potential impacts to the community. This work uses LMs, for which the risks and potential harms are discussed in (Brown et al., 2020). There are potential undesirable biases existed in task-agnostic data (e.g., from Wikipedia) and multi-task downstream datasets (mostly created from news articles). We do not anticipate production of harmful outputs, especially towards vulnerable populations, after using our model or training NLP models on our datasets.

8 Environmental Considerations

We use the same pretrained LMs as in (Du et al., 2021). The energy cost and carbon footprint for the pretrained models were 80.6 MWh and 4.6 tCO2e, respectively. The additional structure pretraining gradient-steps is less than 1.5% of the number of pretraining steps of LMs, and so the estimated additional energy cost is comparatively smaller. In addition, training and tuning pretrained LMs on a wide range of tasks and datasets consume plenitude of energy and increase emissions of carbon dioxide. To alleviate the problem, in this work we make efforts to study the multi-task training, which only involves training on a combination of all datasets once. Our results (e.g., Figure 6) show that, despite the gap between multi-task and multi-task finetune on smaller models, the performance gap becomes minor when the model size scales up to 10 billion parameters. This indicates that we can reduce energy consumption when training large pretrained models via employing the multi-task training.

Acknowledgement

We would like to thank the anonymous reviewers for their suggestions and comments. This material is in part based upon work supported by Berkeley DeepDrive and Berkeley Artificial Intelligence Research. Xiao Liu, Zui Chen, Haoyun Hong, and Jie Tang are supported by the NSFC for Distinguished Young Scholar (61825602) and NSFC (61836013).

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *NAACL-HLT*, pages 3554–3565.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*, pages 344–354.

Ben Athiwaratkun, Cícero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *EMNLP*, pages 375–385.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. In *EMNLP-IJCNLP*, pages 5359–5368.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, pages 5016–5026.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *ACL/IJCNLP*, pages 1860–1874.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *CoNLL*, pages 152–164.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, pages 41–75.

Jun Chen, Robert Hoehndorf, Mohamed Elhoseiny, and Xiangliang Zhang. 2020. Efficient long-distance relation extraction with dg-spanbert. *CoRR*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *WWW*, pages 355–366.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*.

Filipe de Sá Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *EMNLP-ACL*, pages 447–457.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. All NLP tasks are generation tasks: A general pretraining framework. *CoRR*.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*, pages 5467–5471.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI-PAIS*, pages 2006–2013.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *LREC-ELRA*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *LREC*, pages 422–428.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. In *EMNLP-IJCNLP*, pages 6249–6254.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from RDF data. In *INLG*, pages 124–133.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. OPIEC: an open information extraction corpus. *CoRR*.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *COLING*, pages 2537–2547.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Informatics*, pages 885–892.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *EMNLP*, pages 4803–4809.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*, pages 643–653.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *NeurIPS*.

Robert L. Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *CoRR*.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S. Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *EMNLP-IJCNLP*, pages 5802–5807.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*, pages 687–692.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059.

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 *ACL*, pages 7871–7880.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020a. A unified MRC framework for named entity recognition. In *ACL*, pages 5849–5859.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020b. Dice loss for data-imbalanced NLP tasks. In *ACL*, pages 465–476.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. In *AAAI*, pages 6730–6737.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*.

Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, and Jianfeng Gao. 2020. The microsoft toolkit of multi-task deep neural networks for natural language understanding. In *ACL*, pages 118–126.

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 *NAACL-HLT*, pages 3036–3046.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *AAAI*, pages 6851–6858.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *HLT*, page 82–86.

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 *ICLR*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237.

Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *CoRR*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*, pages 143–152.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2019a. Improving language understanding by generative pre-training. *OpenAI blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners. *OpenAI blog*, page 9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*.

Osman Ramadan, Pawel Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *ACL*, pages 432–437.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML-PKDD*, pages 148–163.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *HLT-NAACL*, pages 1–8.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *COLING*, pages 2288–2299.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *EMNLP*, pages 1199–1212.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *NAACL-HLT*, pages 2339–2352.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi S. Jaakkola. 2020. Blank language models. In *EMNLP*, pages 5186–5198.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL*, pages 2895–2905.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *EMNLP*, pages 2300–2305.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *CoRR*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *NAACL-HLT*, pages 885–895.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *EMNLP-IJCNLP*, pages 5783–5788.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. Linguistic Data Consortium.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *EMNLP*, pages 1225–1238.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *CoRR*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL*, pages 808–819.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *ACL*, pages 6953–6963.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *HLT-NAACL*, pages 868–877.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *ACL/IJCNLP*, pages 5808–5822.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *CoRR*.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020a. Joint extraction of entities and relations based on a novel decomposition strategy. In *ECAI-PAIS*, pages 2282–2289.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020b. Named entity recognition as dependency parsing. In *ACL*, pages 6470–6476.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *IJCAI*, pages 4054–4060.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. *CoRR*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *ICML*, pages 11328–11339.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*, pages 35–45.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *IJCAI*, pages 3948–3954.

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *CoRR*.

A Experimental Setup

A.1 Implementation Details

Model Architecture We leverage the Generalized Language Model (GLM) as our base language model, which is pre-trained on autoregressive blank infilling objectives. It improves the pre-train – fine-tune consistency via cloze-style fine-tuning, and naturally handles variable-length blank infilling which is crucial for many downstream tasks. To some extend, GLM can be viewed as an adaptive encoder-decoder architecture.

GLM has the same vocabulary as GPT2 series models', covering 50257 tokens. In this work, we leverage its models in four different scales: 110M, 220M, 2B, and 10B, in which 110M is pre-trained over English Wikipedia and Book-corpus and the others are pre-trained over the Pile corpora (Gao et al., 2021) (approximately the same corpora for training GPT-3). GLM has been reported to outperform T5 over text summarization challenges[2], which is a task that accords with structural prediction. Compared to GPT-3, GLM is a bidirectional model but can conduct autoregressive generation.

Structure Pretraining Procedure

- Pretraining for zero-shot: we conduct the pretraining on 8 NVIDIA DGX-A100 machines using an Adam optimizer with 5e-6 learning rate and 0.1 learning rate decay. We train the model with batch size 4 per GPU for 3 epochs. We select the last iteration checkpoint.

- Downstream multi-task training: we conduct the multi-task training on 8 NVIDIA DGX-A100 machines using an Adam optimizer with 5e-6 learning rate and 0.1 learning rate decay. We train the model with batch size 4 per GPU for 6 epochs and select the best checkpoint for each task based on their development set performance.

- Inference: In the inference, length penalty and minimum target length are the most important hyper-parameters. Length penalty is a float between 0 and 1 to control the GLM's generation length (the larger the longer). For entity-based tasks (e.g., NER, SRL, Event Extraction), a larger length penalty is preferred (e.g., 0.8-1.0); for triple-based tasks (e.g., JER, OIE, DST), a smaller one is preferred (e.g., 0.3-0.5); for other tasks that require a specific number of predicted triples (e.g., Relation Classification, Intent Detection, Factual Probe), we will trim the generation result in the postprocessing.

Pretraining data We apply the task-agnostic pretraining data presented in Table 1 as described in Section 2.1. An exception was for T-REx (ElSahar et al., 2018), where there is an overlap between itself and the Factual Probe task dataset T-Rex used in (Petroni et al., 2020). To avoid the leak, we only sample a portion of the T-REx as our pretraining data to exclude samples appeared in Factual Probe.

[2]https://github.com/THUDM/GLM

Task	Dataset	#Sents		
		Train	Dev	Test
Open information extraction	OIE2016	2,278	571	589
	WEB	-	-	920
	NYT	-	300	149
	PENN	-	-	51
Relation classification	TACRED	68,124	22,631	15,509
	FewRel 1.0	56,000	1,120	–
Joint entity and relation extraction	CoNLL04	922	231	288
	ADE	3,845	–	427
	NYT	56,195	5,000	5,000
	ACE2005	7,477	1,789	1,517
Event extraction	ACE2005 Trigger	11,178	649	642
	ACE2005 Argument	4,450	531	612
Coreference resolution	CoNLL12	3,991	2,359	2,421
Intent detection	ATIS	4,478	500	893
	SNIPS	13,084	700	700
Semantic role labeling	CoNLL05	39,832	3,206	–
	CoNLL05 WSJ	39,832	3,206	5,221
	CoNLL05 Brown	39,832	3,206	779
	CoNLL12	89,549	32,397	21,499
Named entity recognition	CoNLL03	14,041	3,250	3,453
	OntoNotes	59,924	8,528	8,262
	GENIA	14,824	1,855	1,854
	ACE2005	7,299	971	1,060
Dialogue state tracking	MultiWOZ 2.1	62,367	7,371	7,368
Factual probe	Google-RE	-	-	552
	T-REx	-	-	3,403

Table 5: Statistics of downstream datasets.

In the following sections, we introduce the dataset formats, comparison methods and training details for all 10 structural prediction tasks.

A.2 Open Information Extraction

For OIE, we are given a sentence and asked to extract triples.

> **Input** Born in 1951 in Tbilisi, Iago is a Georgian artist.
> **Output** (Iago; Born in; 1951) (Iago; is a; Georgian artist)

Datasets We evaluate the performance of the open information extraction (OIE) systems on OIE benchmark datasets consisting of **OIE2016** (Stanovsky and Dagan, 2016), a dataset from Newswire and Wikipedia automatically converted from QA-SRL (He et al., 2015); three news datasets **NYT** (de Sá Mesquita et al., 2013), **WEB** (de Sá Mesquita et al., 2013), **PENN** (Xu et al., 2013). The statistics of the benchmark are shown in Table 5. The preprocessed datasets are obtained from Supervised OIE (Stanovsky et al., 2018).

Comparison Methods We compare our method DEEPSTRUCT to the following prominent OIE systems recently evaluated in (Stanovsky et al., 2018): ClausIE (Corro and Gemulla, 2013), Open IE4 [3], PropS (Stanovsky et al., 2016), RnnOIE (Stanovsky et al., 2018). We also compare to MAMA with BERT$_{\text{LARGE}}$ recently introduced in (Wang et al., 2020) that also leverages pre-trained LMs to extract open triples. See results in Table 6.

Training Details During multi-task fine-tuning, we train our model on OIE2016 training set for 10 epochs, with a per GPU batch size 4. During inference, for oie2016, we choose a length penalty of 0.8. For WEB, NYT, and PENN, they only contain the test sets, and during the inference, we use a length penalty of 0.5 and trim the prediction to reserve only one triple.

A.3 Relation Classification

For this task, we are given an input sentence with gold head and tail entities aiming to classify the relation type in a pre-defined category.

> **Input** The 1976 Thomas Cup was the tenth edition of Thomas Cup, the world championship of men's international team badminton (its female counterpart is the Uber Cup). The relationship between Uber Cup and badminton is
> **Output** (Uber Cup; sport; badminton)

Datasets We evaluate on FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017).

- **FewRel** contains 100 relations with 7 instances for each relation. The standard evaluation for this benchmark uses few-shot N-way K-shot settings. The entire dataset is split into the train (64 relations), validation (16 relations), and test set (20 relations). We report the same results on the dev set for all the settings because of our zero-shot setting.

- **TACRED** is a large-scale relation classification benchmark that consists of 106,344 examples and 41 relation types including 68,164 for training, 22,671 for validation, and 15,509 for testing. We do not use train and validation sets and report the result on the test set.

We use F1 to evaluate the results. We parse every relation type and the corresponding head and tail entities from every original sample, and formulate every sample into the aforementioned input and output.

Comparison Methods We compare our method with the following supervised methods. (i) BERT-PAIR (Gao et al., 2019) is a sequence classification

[3] https://github.com/dair-iitd/OpenIE-standalone

model based on BERT, optimizing the score of two instances expressing the same relation. (ii) BERT_{EM} + Matching the Blanks (MTB) (Soares et al., 2019), which uses entity markers (BERT_{EM}) and additional pre-training of relations on a large-scale corpus (i.e., MTB). (iii) TANL (Paolini et al., 2021) is a sequence to sequence model based on T5 (Raffel et al., 2019) aiming to generate structured objects from an encoded natural language format. See results in Table 7.

Training Details During multi-task fine-tuning, we train our model on TACRED/FewREL-meta training set for 20 epochs, with a per GPU batch size of 4. During inference, for TACRED, we provide the decoder with the prefix "(head; " and ask the model to generate the relation and tail. FewRel dev set results are acquired similarly. We choose a length penalty of 0.5.

A.4 Factual Probe

Given an input sentence with gold head entity name and relation name, the task aims to fill in the tail entity.

> **Input** Daniel Bowen, born in 1970, is a Melbourne resident best known as the author of the blog, Diary of an Average Australian.
> **Output** (Daniel Bowen; date of birth; 1970)

Datasets We consider the Google-RE consisting of 3 relations and 5,527 facts, and T-REx with 41 relations and 34,039 facts of the LAMA benchmark (Cao et al., 2021). We evaluate the results using mean precision at one (P@1), where higher values are better. We parse every relation type and the corresponding head and tail entities from every original sample, and formulate every sample into the aforementioned input and output.

Comparison Methods We compare to pre-trained LM-based methods that leverage the output probabilities of the LM to make predictions given the sentence known to express the fact. Two methods are considered: (i) LAMA (Cao et al., 2021) leverages the input sentence without the tail entity to query the LMs, and (ii) LAMA-Oracle (Petroni et al., 2020) enriches the query with (at most) five gold sentences as additional context. See results in Table 15.

Training Details During multi-task fine-tuning, we train our model on TACRED/FewREL-meta training set for 20 epochs, with a per GPU batch size of 4. During inference, for TACRED, we provide the decoder with the prefix "(head; " and ask the model to generate the relation and tail. FewRel dev set results are acquired similarly. We choose a length penalty of 0.5.

A.5 Joint Entity and Relation Extraction

Given a sentence, this task aims to extract a set of entities (one or more consecutive tokens) and a set of relations between pairs of entities. Each predicted entity and relation has to be assigned to an entity or a relation type.

> **Input** Blackstone already holds a 50 percent stake in the two parks that make up Universal Orlando .
> **Entity Output** (Blackstone; instance of; organization) (parks; instance of; organization) (that; instance of; organization) (Universal Orlando; instance of; organization)
> **Relation Output** (Blackstone; employer; parks)

Datasets In the ablation study, we also present a model variant with entity and relation augmentation (e.g., we asked DEEPSTRUCT to generate "([Iago]; instance of; person) ([Iago]; city_of_birth; [Tbilisi])" for the case in Figure 1).We experiment on the following datasets: CoNLL04 (Roth and Yih, 2004), ADE (Gurulingappa et al., 2012), NYT (Riedel et al., 2010), and ACE2005 (Walker and Consortium, 2005).

- The **CoNLL04** dataset: CoNLL04 consists of annotated named entities and relations on sentences taken from WSJ, AP, etc. We are using the same split as what was proposed (Gupta et al., 2016). We train all models for 200 epochs.
- The **ADE** dataset: ADE contains annotated documents aiming at improving automatic extraction of drug-related adverse effects from medical case reports. We are using the same 10-fold cross-validation split as in (Paolini et al., 2021). We train all models for 200 epochs and report our test macro-F1 score across all 10 splits.
- The **NYT** dataset: NYT is processed from New York Times corpus automatically labeled using distant supervision. We are using the processed version of this dataset (Yu et al., 2020a). We train all models for 50 epochs.

- The **ACE2005** dataset: ACE2005 is processed from the ACE 2005 Multilingual Training Corpus held by Linguistic Data Consortium. We are using the processed version of this dataset by (Luan et al., 2019), preserving 7 entity types and 6 relation types as in TANL. We train all models for 100 epochs.

Comparison Methods We compare our method DEEPSTRUCT on the four datasets to the following JER baselines: SpERT (Eberts and Ulges, 2020), DyGIE (Luan et al., 2019), MRC4ERE (Zhao et al., 2020), RSAN (Yuan et al., 2020) and TANL (Paolini et al., 2021). See results in Table 12.

Training Details During multi-task fine-tuning, we train our model on JER training sets for 5-20 epochs, with a per GPU batch size of 4. Since we discover that relation extraction and named entity recognition need different length penalties, we split the training set corresponding to two tasks separately. During inference, we choose a length penalty of 0.8 for named entity recognition, and 0.3 for relation extraction.

A.6 Named Entity Recognition

This is an entity-only case of the joint entity and relation extraction task.

> **Input** What we need to do is to make sure that state boards, number one, have adequate funding.
> **Output** (we; instance of; human) (state; instance of; geographical entity) (state boards; instance of; organization)

Datasets We experiment on the following datasets: CoNLL03 (Sang and Meulder, 2003), Ontonotes (Pradhan et al., 2013), GENIA (Ohta et al., 2002), and ACE2005 (Walker and Consortium, 2005).

- The **CoNLL03** dataset: CoNLL03 (English) data was taken from the Reuters Corpus. We are using the processed version of this dataset (Li et al., 2020a). We train all models for 200 epochs.

- The **Ontonotes** dataset: Ontonotes is processed from the OntoNotes Release 5.0 Corpus held by Linguistic Data Consortium. We are using the preprocessing scripts provided by (Luan et al., 2019). We train all models for 50 epochs.

- The **GENIA** dataset: GENIA dataset consists of compiled and annotated biomedical literature. We are using the processed version of this dataset (Li et al., 2020a). We train all models for 100 epochs.

- The **ACE2005** dataset: ACE2005 is processed from the ACE 2005 Multilingual Training Corpus held by Linguistic Data Consortium. Notice that it is also processed from ACE2005 corpus but using data split differently from the ACE2005 joint entity and relation extraction dataset. We are using the processed version of this dataset by (Li et al., 2020a). We train all models for 50 epochs.

Comparison Methods We compare our method DEEPSTRUCT on the four datasets to the following NER baselines: BERT-MRC (Li et al., 2019), BERT-MRC+DSC (Li et al., 2020b), Cloze-CNN (Baevski et al., 2019), GSL (Athiwaratkun et al., 2020), BiaffineLSTM (Yu et al., 2020b), and TANL (Paolini et al., 2021). See results in Table 13.

Training Details During multi-task fine-tuning, we train our model on NER training sets for 15 epochs, with a per GPU batch size of 4. During inference, we choose a length penalty of 0.8 for named entity recognition. Since some datasets may require null prediction, we set the minimum target length to 0 to represent the null prediction.

Implementation Details As the conventional NER's evaluation is based on extractive span matching, to make a fair comparison on situations where there are multiple entities with the same surface, we adopt the following strategy: we match the generated entities' spans from left to right in the original sentence at place where they are first mentioned; if there are duplicated entities, the first generated one matches the first mention span, and the second matches the second mention span, etc.

A.7 Semantic Role Labeling

Here we are given an input sentence along with a predicate, and seek to predict a list of arguments and their types. Every argument corresponds to a span of tokens that correlates with the predicate in a specific manner (e.g. subject, location, or time). The predicate is marked in the input, whereas arguments are marked in the output and are assigned an argument type.

Input Scotty [accepted] the decision with indifference and did not enter the arguments .
Output (Scotty; instance of; first argument) (the decision; instance of; second argument)

Datasets We experiment on the following datasets: CoNLL05 WSJ/Brown (Carreras and Màrquez, 2005) and CoNLL12 (Pradhan et al., 2013). Sentences with multiple target predicates for semantic role labeling are duplicated during preprocessing so that each sentence will be related to one and only one target predicate, marked by symbols "[]". We adopted the same evaluation scripts as TANL (Paolini et al., 2021).

- The **CoNLL05 WSJ/Brown** dataset: CoNLL05 WSJ/Brown dataset shares the same train and validation split while differing on the test set. As their names suggest, the test dataset is taken from WSJ corpus and Brown corpus, separately. We train all models for 50 epochs.
- The **CoNLL12** dataset: CoNLL12 dataset is built upon Ontonotes dataset. We train all models for 50 epochs.

Comparison Methods We compare our method DEEPSTRUCT on the four datasets to the following SRL baselines: Dep and Span (Li et al., 2019), BERT SRL (Shi and Lin, 2019), and TANL (Paolini et al., 2021). See results in Table 11.

A.8 Event Extraction

This task requires extracting (1) event triggers, each indicating the occurrence of a real-world event and (2) trigger arguments indicating the attributes associated with each trigger.

Trigger input But the Saint Petersburg summit ended without any formal declaration on Iraq .
Trigger output (summit; instance of; meet)
Argument input But the Saint Petersburg [summit] ended without any formal declaration on Iraq .
Argument output (Saint Petersburg; instance of; place)

Datasets We experiment on the following dataset: ACE2005 (Walker and Consortium, 2005). For the trigger prediction task, the dataset is handled similar to named entity recognition fashion. For the argument prediction task, which is based on trigger predictions, we generated all trigger predictions using our 10B model during preprocessing.

- The **ACE2005** dataset: ACE2005 is processed from the ACE 2005 Multilingual Training Corpus held by Linguistic Data Consortium. The data of event extraction is different from that of named entity recognition or joint entity and relation extraction. We train all models for 50 epochs.

Comparison Methods We compare our method DEEPSTRUCT on the four datasets to the following EE baselines: J3EE (Nguyen and Nguyen, 2019), DyGIE++ (Wadden et al., 2019), and TANL (Paolini et al., 2021). See results in Table 8.

Training Details During multi-task fine-tuning, we train our model on ACE2005 event trigger/argument training sets for 20 epochs, with a per GPU batch size 4. During inference, we choose a length penalty of 0.8. Since the argument dataset requires assigning a trigger and then doing the prediction, we use a pair of square brackets to wrap up the trigger. If there is more than one trigger in a dataset, we will duplicate the sentence with different marked triggers.

A.9 Coreference Resolution

This is the task of grouping individual text spans (mentions) referring to the same real-world entity. For each mention that is not the first occurrence of a group, we reference with the first mention.

Input And deterrents don't work terribly well when an enemy values your death more than his life.
Output (an enemy; refer to; his)

Datasets We experiment on the following dataset: CoNLL12 (Pradhan et al., 2013). During preprocessing, the dataset is chopped into chunks of a fixed size 512. Only intra-chunk coreferences are preserved, following TANL (Paolini et al., 2021). Also, we used the same evaluation scripts as TANL.

- The **CoNLL12** dataset: CoNLL12 dataset is built upon Ontonotes dataset. We train all models for 50 epochs.

Comparison Methods We compare our method DEEPSTRUCT on the four datasets to the following COREF baselines: Higher-order c2f-coref (Lee et al., 2018), BERT+c2f-coref (Joshi et al., 2019), CorefQA+SpanBERT (Wu et al., 2020), and TANL (Paolini et al., 2021). See results in Table 9.

Training Details During multi-task fine-tuning, we train our model on CoNLL12 coreference resolution training sets for 40 epochs, with a per GPU batch size 4. During inference, we choose a length penalty of 0.8.

A.10 Dialogue State Tracking

Here we are given as input history of dialogue turns, typically between a user (trying to accomplish a goal) and an agent (trying to help the user). The desired output is the dialogue state, consisting of a value for each key (or slot name) from a predefined list.

> **Input** [User]: I would like a taxi from Saint Johns College to Pizza Hut Fen Ditton. [Agent]: What time do you want to leave and what time do you want to arrive by? [User]: I want to leave after 17:15.
> **Output** ([User]; taxi arrive by; not given) ([User]; taxi departure; Saint Johns College) ([User]; taxi destination; Pizza Hut Fen Ditton) ([User]; taxi leave at; 17:15)

Datasets We use the MultiWOZ 2.1 (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2020; Zang et al., 2020) task-oriented dialogue dataset in our experiments. It consists of 8,420 conversations for training, 1,000 for validation, and 999 for testing. We follow the pre-processing procedure put forward (Wu et al., 2019) for dialogue state tracking. In addition, we remove the "police" and "hospital" domains from the training set since they are not present in the test set. Removing these two domains reduces the training set size from 8,420 to 7,904. We fine-tune for 100 epochs, with a maximum sequence length set to 512 tokens. We train a single generative model that predicts the dialogue state for the entire dialogue history up to the current turn. Following prior work, we report the joint accuracy. We parse every slot name and the corresponding value from every original sample, and formulate every sample into the aforementioned input and output.

Comparison Methods We compare our performance on MultiWOZ 2.1 against SimpleTOD (Hosseini-Asl et al., 2020), the current state of the art for MultiWOZ dialogue state tracking. SimpleTOD uses a sequence to sequence approach based on the GPT-2 (Radford et al., 2019b) language model. Unlike our approach, SimpleTOD is trained to jointly generate actions and responses as well as dialogue states. See results in Table 14.

A.11 Intent Detection

Intent detection is the task of interpreting user commands or queries by extracting the intent and the relevant slots.

> **Input** Show flight and prices from Kansas City to Chicago next Wednesday arriving in Chicago by 7 pm.
> **Output** (intent; is; flight and airfare)

Datasets We use two datasets, the ATIS dataset (Hemphill et al., 1990) and the Snips dataset (Coucke et al., 2018). The ATIS dataset consists of 4,478 samples for training, 500 for validation, and 893 for testing. The Snips dataset consists of 13,084 samples for training, 700 for validation, and 700 for testing. We formulate the label of every sample to "(intent; is; [label])". We fine-tune for 20 epochs, with a maximum sequence length set to 512 tokens. Following prior work, we report accuracy. We parse every intent from every original sample, and formulate them into the aforementioned input and output.

Comparsion Methods We majorly compare our methods to SF-ID (E et al., 2019) and TANL (Paolini et al., 2021) in this task. See results in Table 10.

B Dataset Examples

The examples are shown in Table 16.

C Error Analysis

We analyzed recall errors of DEEPSTRUCT 10B MP on CoNLL04 relation extraction task in Table 17. We found that most relation extraction errors in our method are caused by slight deviation in entity prediction: either the predicated entity has almost the same span of the ground truth entity (e.g.: "U.S." and "the U.S.", "America" and "American"), or the predicated entity has a roughly similar meaning to the ground truth entity and plays roughly the same role in the relation (e.g.: "Fairbanks" and "Alaska"). Besides, we also observed some interesting errors in which our prediction has a different focus from the ground truth relation, and our prediction is also meaningful in terms of human understanding.

		OIE2016	WEB	NYT	PENN
ClausIE (Corro and Gemulla, 2013)		58.8	44.9	29.6	34.6
OpenIE 4		59.6	55.7	38.3	42.6
PropS (Stanovsky et al., 2016)		55.6	58.9	37.2	39.1
RnnOIE (Stanovsky et al., 2018)		67.0	58.1	28.3	34.5
MAMA (Wang et al., 2020)		36.6	54.3	32.9	33.0
DEEPSTRUCT	zero-shot	28.1	43.8	28.9	51.0
	multi-task	71.2	50.8	43.6	54.5
	w/ finetune	71.3	49.1	45.0	45.1

Table 6: Results on open information extraction.

		TACRED	FewRel 1.0			
			5-1	5-5	10-1	10-5
BERT_{EM} (Soares et al., 2019)		70.1	88.9	-	82.8	-
BERT_{EM}+MTB (Soares et al., 2019)		71.5	90.1	-	83.4	-
DG-SpanBERT (Chen et al., 2020)		71.5	-	-	-	-
BERT-PAIR (Gao et al., 2019)			85.7	89.5	76.8	81.8
NLI-DeBERTa (Sainz et al., 2021)		73.9				
TANL (Paolini et al., 2021)		71.9	93.6±5.4	97.6±3.2	82.2±5.1	89.8±3.6
TANL (multitask) (Paolini et al., 2021)		69.1	-	-	-	-
DEEPSTRUCT	zero-shot	36.1	72.4±6.9	70.8±8.0	67.6±4.5	66.4±6.3
	multi-task	74.9	93.6±6.0	96.4±4.2	92.2±6.4	94.6±3.6
	w/ fine-tune	76.8	98.4±2.8	100±0.0	97.8±2.0	99.8±0.6

Table 7: Results on relation classification

		Trigger Id	Trigger Cl	Argument Id	Argument Cl
J3EE (Nguyen and Nguyen, 2019)		72.5	69.8	59.9	52.1
DyGIE++ (Wadden et al., 2019)			69.7	55.4	52.5
TANL (Paolini et al., 2021)		72.9	68.4	50.1	47.6
TANL (multitask) (Paolini et al., 2021)		71.8	68.5	48.5	48.5
DEEPSTRUCT	multi-task	71.7	67.9	54.9	52.7
	w/ fine-tune	73.5	69.8	59.4	56.2

Table 8: Results on event extraction (ACE2005).

		CoNLL12			
		MUC	B^3	$\text{CEAF}_{\phi 4}$	Avg. F1
Higher-order c2f-coref (Lee et al., 2018)		80.4	70.8	67.6	73
BERT+c2f-coref (Joshi et al., 2019)		81.4	71.7	68.8	73.9
CorefQA+SpanBERT (Wu et al., 2020)		86.3	77.6	75.8	79.9
TANL (Paolini et al., 2021)		81.0	69.0	68.4	72.8
TANL (multitask) (Paolini et al., 2021)		78.7	65.7	63.8	69.4
DEEPSTRUCT	multi-task	63.9	57.7	60.2	60.6
	w/ fine-tune	74.9	71.3	73.1	73.1

Table 9: Results on coreference resolution.

		ATIS	SNIPS
SF-ID (E et al., 2019)		97.8	97.4
TANL (Paolini et al., 2021)		97.6	98.7
DEEPSTRUCT	multi-task	66.6	78.4
	w/ fine-tune	97.8	97.3

Table 10: Results on intent detection.

		CoNLL05 WSJ	CoNLL05 Brown	CoNLL12
Dep and Span (Li et al., 2019)		86.3	76.4	83.1
BERT SRL (Shi and Lin, 2019)		88.8	82.0	86.5
TANL (Paolini et al., 2021)		89.3	82.0	87.7
TANL (multitask) (Paolini et al., 2021)		89.1	84.1	87.7
DEEPSTRUCT	multi-task	95.6	92.0	97.6
	w/ fine-tune	95.2	92.1	96.0

Table 11: Results on semantic role labeling.

		CoNLL04		ADE		NYT		ACE2005	
		Ent	Rel	Ent	Rel	Ent	Rel	Ent	Rel
SpERT (Eberts and Ulges, 2020)		88.9	71.5	89.3	78.8				
DyGIE (Luan et al., 2019)								88.4	63.2
MRC4ERE (Zhao et al., 2020)		88.9	71.9					85.5	62.1
RSAN (Yuan et al., 2020)							84.6		
TANL (Paolini et al., 2021)		89.4	71.4	90.2	80.6	94.9	90.8	88.9	63.7
TANL (multitask) (Paolini et al., 2021)		90.3	70.0	91.2	83.8	94.7	90.7	-	-
DEEPSTRUCT	zero-shot	48.3	25.8	60.7	10.6	60.5	28.6	31.8	5.3
	multi-task	87.4	69.6	90.2	83.7	95.4	93.9	87.8	54.0
	w/ fine-tune	90.7	78.3	91.1	83.8	95.9	93.3	90.0	66.8

Table 12: Results on joint entity relation extraction.

		CoNLL03	OntoNotes	GENIA	ACE2005
BERT-MRC (Li et al., 2020a)		93.0	91.1	-	86.9
BERT-MRC+DSC (Li et al., 2020b)		93.3	92.1		
Cloze-CNN (Baevski et al., 2019)		93.5			
GSL (Athiwaratkun et al., 2020)		90.7	90.2		
BiaffineLSTM (Yu et al., 2020b)		93.5	91.3	80.5	85.4
TANL (Paolini et al., 2021)		91.7	89.8	76.4	84.9
TANL (multitask) (Paolini et al., 2021)		91.7	89.4	76.4	-
DEEPSTRUCT	zero-shot	44.4	42.5	47.2	28.1
	multi-task	93.1	87.6	80.2	-
	w/ fine-tune	93.0	87.8	80.8	86.9

Table 13: Results on named entity recognition.

		MultiWOZ 2.1
TRADE (Wu et al., 2019)		45.6
SimpleTOD (Hosseini-Asl et al., 2020)		55.7
TANL (Paolini et al., 2021)		50.5
TANL (multitask) (Paolini et al., 2021)		51.4
DEEPSTRUCT	multi-task	53.5
	w/ fine-tune	54.2

Table 14: Results on dialogue state tracking.

		Google-RE	T-Rex
LAMA-Oracle (Petroni et al., 2020)		74.3	66.0
DEEPSTRUCT	zero-shot	97.9	85.0
	multi-task	90.3	71.0

Table 15: Results on factual probe.

Task	Dataset	Input	Output
Open Information Extraction	OIE2016	oie oie2016: But for now, at least, Americans are far better at making PCs and the software that runs them.	(Americans; making; PCs and the software that runs them) (PCs; runs; software)
	WEB	oie web: Finally google bought youtube.	(google; bought; youtube)
	NYT	oie nyt: Now in its 58th final, the United States is pursuing a 30th cup title.	(United States; pursuing; cup)
	PENN	oie penn: Samsung already owns korea first advertising co., that country's largest agency.	(Samsung; owns; korea first advertising co.)
Relation Classification	TACRED	rc tacred: Donald Wildmon , the founder and head of the American Family Association , is asking its members to petition Congress to end all funding for PBS . The relationship between Donald Wildmon and American Family Association is	(Donald Wildmon; employee of; American Family Association)
	FewRel 1.0	rc fewrel: Boott was elected an Associate Fellow of the American Academy of Arts and Sciences in 1835 . The relationship between Boott and American Academy is	(Boott; member of; American Academy)
Factual Probe	Google-RE	fp google-re: Eldon Coombe (born c 1941) is a Canadian curler from Ottawa, Canada.	(Eldon Coombe; date of birth; 1941)
	T-REx	fp t-rex: Kurt Schwertsik (born 25 June 1935, Vienna) is an Austrian contemporary composer.	(Kurt Schwertsik; place of birth; Vienna)
Joint Entity and Relation Extraction	CoNLL04	jer conll04: An art exhibit at the Hakawati Theatre in Arab east Jerusalem was a series of portraits of Palestinians killed in the rebellion .	(Hakawati Theatre; instance of; organization) (Arab; instance of; other) (Jerusalem; instance of; location) (Palestinians; instance of; other) (Hakawati Theatre; organization based in; Jerusalem)
	ADE	jer ade: Lethal anuria complicating high dose ifosfamide chemotherapy in a breast cancer patient with an impaired renal function .	(Lethal anuria; instance of; disease) (ifosfamide; instance of; drug) (Lethal anuria; effect; ifosfamide)
	NYT	jer nyt: Mary L. Schapiro , who earlier this year became the new head of NASD , was more amenable to fashioning a deal to the New York Exchange 's liking than her predecessor , Robert R. Glauber .	(NASD; instance of; organization) (Robert R. Glauber; instance of; human) (Robert R. Glauber; company; NASD)
	ACE2005	jer ace2005: The Davao Medical Center , a regional government hospital , recorded 19 deaths with 50 wounded .	(Davao Medical Center; instance of; organization) (government; instance of; geographical entity) (hospital; instance of; organization) (50; instance of; human) (hospital; part of; government)
Named Entity Recognition	CoNLL03	ner conll03: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday .	(Japan; instance of; location) (Asian Cup; instance of; miscellaneous) (Syria; instance of; location)
	OntoNotes	ner ontonotes: Relevant departments from Beijing Municipality promptly activated emergency contingency plans .	(Beijing Municipality; instance of; country city state)
	GENIA	ner genia: Human T and B lymphocytes demonstrate an early and transient hyperpolarization after ligand binding .	(Human T and B lymphocytes; instance of; cell type)
	ACE2005	ner ace2005: BEGALA Dr . Palmisano , again , thanks for staying with us through the break .	(Dr; instance of; human) (Dr . Palmisano; instance of; human) (us; instance of; human)
Semantic Role Labeling	CoNLL05 WSJ	srl conll05: But while the New York Stock Exchange did n't [fall] apart Friday as the Dow Jones Industrial Average plunged 190.58 points – most of it in the final hour – it barely managed to stay this side of chaos .	(the New York Stock Exchange; instance of; second argument) (n't; instance of; negation)
	CoNLL05 Brown	srl conll05: His father [tried] to make the food a topic .	(His father; instance of; first argument) (to make the food a topic; instance of; second argument)
	CoNLL12	srl conll12: Dear viewers , the China News program will [end] here .	(the China News program; instance of; second argument) (will; instance of; modal) (here; instance of; location)
Event Extraction	ACE2005 Trigger	ee ace2005 trg: The European Union held a summit in Brussels.	(summit; instance of; meet)
	ACE2005 Argument	ee ace2005 arg: The European Union held a [summit] in Brussels.	(Brussels; instance of; place)
Coreference Resolution	CoNLL12	cr conll12: And deterrents does n't work terribly well when an enemy values your death more than his life .	(an enemy; refer to; his)
Dialogue State Tracking	MultiWOZ 2.1	dst multiwoz: [User]: I am looking for a place to to stay that has cheap price range it should be in a type of hotel. [Agent]: Okay , do you have a specific area you want to stay in? [User]: No , I just need to make sure it s cheap. Oh, and I need parking. [Agent]: I found 1 cheap hotel for you that include parking. Do you like me to book it? [User]: Yes, please. 6 people 3 nights starting on Tuesday.	([User]; hotel area; not given) ([User]; hotel book day; Tuesday) ([User]; hotel book people; 6) ([User]; hotel book stay; 3) ([User]; hotel internet; not given) ([User]; hotel name; not given) ([User]; hotel parking; yes) ([User]; hotel price range; cheap) ([User]; hotel stars; not given) ([User]; hotel type; hotel)
Intent Detection	ATIS	id atis: Please give me a list of all the flights between Dallas and Baltimore and their cost.	(intent; is; flight and airfare)
Intent Detection	SNIPS	id snips: Play the song little robin redbreast.	(intent; is; play music)

Table 16: Input/output examples for every datasets.

Error type	Percentage	Input	Ground Truth	Ours Prediction
Close Entity	65.3%	*Locations containing suitable federally owned land were listed as : Fort Wainwright annex , Fairbanks , Alaska ;*	(Fort Wainwright annex ; located in ; Fairbanks)	(Fort Wainwright annex ; located in ; Alaska)
Totally Missing	26.4%	*Judith C. Toth says she returned for a fourth term in Maryland 's House of Delegates because she couldn 't find a better job .*	(House of Delegates ; organization based in ; Maryland)	(Judith C. Toth ; lives in ; Maryland)
Wrong Relation	4.2%	*After buying the shawl for $1 , 600 , Darryl Breniser of Blue Ball , said the approximately 2-by-5 foot shawl was worth the money .*	(Darryl Breniser ; lives in ; Blue Ball)	(Darryl Breniser ; works for ; Blue Ball)
Different Focus	1.7%	*An architect of President Nixon 's unsuccessful executive-privilege Watergate defense is a top prospect for the post of U.S. solicitor in the new Bush administration .*	(Bush ; lives in ; U.S.)	(Nixon ; lives in ; U.S.)

Table 17: Analysis of frequently-occurring recall errors of DEEPSTRUCT on CoNLL04 relation extraction task. For each type we list the percentage of missing triples caused by this particular type of error, and an example of this type of error taken from the CoNLL04 corpus.