Structured Language Generation Model: Loss Calibration and Formatted Decoding for Robust Structure Prediction

Minho Lee[1] **Junghyun Min**[2] **Yerang Kim**[3] **Woochul Lee** **Yeonsoo Lee**[5]

[1]KT Gen AI Lab [2]Georgetown University [3]Korea University [4]NC AI

[1]minolee@kt.com

Abstract

Modern generative pre-trained language models excel at open-ended text generation, yet continue to underperform on structure-related tasks such as NER, relation extraction, and semantic role labeling, especially when compared to encoder-only models of similar sizes. While this gap has been attributed to limited structure knowledge, we hypothesize this is also due to the missing connection between the model's internal representations of linguistic structure and the output space used during supervised fine-tuning. We propose the Structured Language Generation Model (SLGM), a model- and task-agnostic framework that reformulates structured prediction as a classification problem through three components: (1) reinforced input formatting with structural cues, (2) loss design, and (3) format-aware decoding that constrains generation to task-valid outputs. Across 5 tasks and 13 datasets, SLGM substantially improves structure prediction without relying on dataset-specific engineering or additional model parameters. It outperforms baseline fine-tuning on models of the same size, achieves comparable performance to much larger models when used with $<$1B parameter models, and acts as a zero-weight adapter that reproduces the benefits of dataset-specific fine-tuning in low-resource settings.

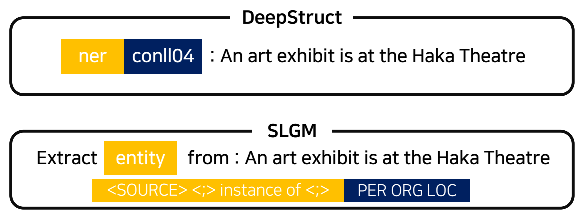

Figure 1: Sample NER example in DEEPSTRUCT (Wang et al. 2023) and SLGM. The yellow and navy highlight task specific and dataset information, respectively. SLGM breaks down the implicit information into explicit output format and tagset information.

1 Introduction

Recent advances in pre-trained language models (PLMs) allow computational approaches to more language tasks and problems than ever (Devlin et al. 2019; Raffel et al. 2023). Generative PLMs (GLMs) perform strongly on natural language generation (Radford et al. 2019; Lewis et al. 2020), and engineering methods like scaling and post-training methods like instruction tuning and reinforcement learning from human feedback further improve generation quality (Brown et al. 2020; Chung et al. 2022; Bai et al. 2023; OpenAI 2024).

However, such success has not been mirrored in many natural language understanding (NLU) tasks that require knowledge of syntactic and semantic structure where the scale and post-training associated with GLMs offer little improvement over much smaller, unaligned LMs (Zhong et al. 2023; Hu et al. 2025).

Beyond benchmark scores, GLMs still struggle to represent linguistic structure compared to their strength in next-token prediction. GLMs' ability to retrieve previously seen entities depend on the order of the input (Berglund et al. 2024; Kitouni et al. 2024). Their ability to predict syntactic structure remains poor even with in-context learning (Bai et al. 2025). Wang et al. (2023); Min et al. (2025) show that additional structure pre-training improves performance on structure-related downstream tasks. Table 1 illustrates a similar behavior, where GPT-4 (OpenAI 2024) falls short against smaller models in CoNLL04 joint entity and relation extraction (Roth and Yih 2004), even when equipped with detailed instructions and examples.

However, GLMs' lower-than-expected performance on various structure-related tasks, including named entity recognition (NER), relation extraction (RE), semantic role labeling (SRL), intent detection (ID), and dialogue state tracking (DST) may not be entirely attributed to the lack of linguistic structure knowledge in them, as they are capable of syntactically well-formed and semantically coherent text (Herbold et al. 2023; Olmedilla et al. 2024; Acciai et al. 2025; Lauriola, Campese, and Moschitti 2025). We hypothesize that GLMs' comparatively weak performance in structure-related tasks is due to the missing connection between the internal representations of linguistic structure and the output tokens, and test how model- and dataset-agnostic designs like input formatting, loss design, and restrictions in decoding can help retrieve structure information from GLMs more effectively.

In this paper, we present our model- and task-agnostic framework, the Structured Language Generation Model

Model	Meta-info	Ent. F1	Rel. F1
DeepStruct	+task +dset	88.4	72.8
	+task +tag	81.7	40.7
	+task	79.6	48.2
TANL	+dset	90.3	70.0
	-	24.0	2.8
GPT-4*	+task +tag	57.7	34.1
SLGM	+task +tag	71.7	27.9
	+task +tag +ft	85.5	55.2

Table 1: DeepStruct (Wang et al. 2023) and TANL (Paolini et al. 2021) performance on CoNLL-04 joint entity recognition and relation classification (JER) are better with task information (+task), dataset information (+dset), tagset information (+tag), and degrade without them. GPT-4 (OpenAI 2024) performance on a test subset and SLGM performance with and without finetuning (+ft) as reference; Appendix A describes GPT-4 prompt.

(SLGM), which comprises reinforced input formatting, loss functions complementary to the cross-entropy loss, and format-aware decoding method. An example of its input formatting is illustrated in Figure 1. We show that the SLGM framework is able to bridge the gap between GLMs' ability to capture hierarchical structure in language and their subpar structure prediction performance obtained from vanilla finetuning. SLGM offers significant performance improvements over baseline models of the same size, and is competitive against other much larger $>$10B models even when used with $<$1B models, achieving the state-of-the-art performance in NER with CoNLL-03 (Tjong Kim Sang and De Meulder 2003) and intent detection with ATIS (Hemphill, Godfrey, and Doddington 1990). We also present SLGM as an effective zero-weight adapter that simulates dataset-specific fine-tuning for low-resource environments.

2 Structure Prediction

Structure prediction tasks build on the sequence generation framing of NLP tasks, first adopted by generative pre-trained models (Lewis et al. 2020; Chung et al. 2022; Du et al. 2022; Raffel et al. 2023). TANL (Paolini et al. 2021), DeepEx (Wang et al. 2021a), and UIE (Lu et al. 2022) propose intermediate representations of entity and argument structure within text, which may be comparable to more general representations of meaning (e.g. Banarescu et al. 2013), incorporated into downstream tasks with mixed results (Wein and Opitz 2024; Jin et al. 2024; Min, Yang, and Wein 2025).

Recent work explores how explicit structure modeling can enhance generation or extraction performance. Lu et al. (2022); Wang et al. (2023); Min et al. (2025) introduce additional structure-related pre-training, with or without supervision, while Wang et al. (2021b) concatenate existing pre-trained representations to improve downstream task performance. In a parallel line of research, Sun et al. (2020) and Wang et al. (2022) propose the Open Information Expression (OIX) and Open Information Annotation (OIA) frameworks, which represent predicate–argument relations in a unified, lossless format. These frameworks decouple structural annotation from task-specific modeling, allowing OIE strategies to be reused and adapted efficiently across diverse tasks. The OIA-based system of Wang et al. (2022) further demonstrates that explicitly modeling predicate–function–argument structures can yield strong adaptability and competitive performance even with limited training data, underscoring the value of explicit structure representations in general-purpose information extraction.

By definition, structured prediction tasks benefit from decoding methods that restrict the set of candidate tokens to choose from. Such characteristic resembles the copying mechanism (Gu et al. 2016), which allows subsequences of source text to be "copied" when generating target text. This approach has been adopted for summarization (Zeng et al. 2016; See, Liu, and Manning 2017; Gehrmann, Deng, and Rush 2018; Choi et al. 2021), dialogue systems (Park et al. 2023), machine translation (Ghazvininejad et al. 2019), data-to-text generation (Choi et al. 2021), and data augmentation for aspect-based sentiment analysis (Li et al. 2020).

Another challenge in structured prediction tasks lies in producing outputs with precise formatting. Earlier work often relied on rule-based systems (e.g. Hovy 1993), which guaranteed stability but lacked the ability to generalize. In contrast, pretrained language models offer broad generalization power, yet controlling their outputs remains difficult (Zhang et al. 2023). Despite their broad generalization abilities, pretrained language models often fail to reliably adhere to structural or formatting constraints, motivating approaches for constrained decoding and controllable generation (Keskar et al. 2019; Holtzman et al. 2020; Lu et al. 2021). Work in code generation have explored ways to enforce structure during generation, such as constraining decoding with formal grammars or pushdown automata (Dong et al. 2023), or adjusting the granularity of token generation (Ling et al. 2016). At the same time, large commercial models demonstrate strong capabilities in reliably producing well-structured outputs like code (Bai et al. 2023; OpenAI 2024; Team et al. 2025), highlighting the potential of combining powerful language modeling with explicit control mechanisms.

3 SLGM Framework

SLGM operates around task and data-specific information. For each task and dataset, we establish a predefined tagset and output format, which are then enforced via loss and forced decoding outlined in this section.

3.1 Task-specific Output Format

Each task or dataset, when framed as a sequence-to-sequence task, has its own output format. Such format consists of separator tokens and the segments between them, which we call *slots*. For example, a single output string may contain slot separator tokens (<;>) and an object separator token (</>) to indicate the end of the object, which can be a named entity, a verb with semantic roles, or a head-predicate-tail triple in information extraction. Each slot can contain a list of strings or one of two special slot tokens - `<ANY>` and `<SOURCE>`. An `<ANY>` token indicates the model can generate any token, while the `<SOURCE>` token indicates the model should source from the input. If a list is given in the slot, the model is to choose from the list of pre-defined strings, which contains tagset information or other placeholder labels. Before inference, SLGM parses the format string into a format mask tensor, a boolean tensor that has true values on legal token ids for each slot. This tensor is then used in calculating format losses or disabling illegal tokens in the decoding process, described in the following Sections 3.2 and 3.3.

3.2 Format Loss

Traditionally popular cross-entropy loss is calculated over every possible vocabulary defined in the model. We find this to target an overly broad token space. Thus, we introduce two complementary losses: a structure loss, and a slot loss.

Structure Loss. Structure loss L_{st} aims to improve separator generation. It penalizes incorrect separator predictions by:

$$L_{st} = \sum_{t \in S} l_t \cdot missed \cdot w_{miss}$$

where l_t is log probability of token t, S separator token locations, w_{miss} miss weight, a hyperparameter, and $missed$ the number of instances where a separator token did not have the highest token probability at its true position. Structure loss is thus exponential here–for example, if there are 3 missed separators, the loss is $L = \sum_{n=1}^{3} l_t * 3 * w_{miss}$, for a total multiplier of 9.

Slot Loss. Slot loss is similar to the traditional cross-entropy loss, but reflects the design of SLGM, which provides a set of candidate tokens for generation, whether with a `<SOURCE>` token or a pre-defined list. For example, NER predictions should comprise a set of tokens from the source sentence, and one label selected from a pre-defined list of possible labels. The loss's key distinction from CE is in the denominator's vocab size, effectively converting a sequence generation task into a classification task. We implement the final training loss as a weighted sum of cross-entropy, structure, and slot losses, where each coefficient-weight is a hyperparameter.

3.3 Formatted Decoding during Inference

During inference, SLGM maintains state information that tracks the current stage of structured text generation within each sentence. This state reflects which part of the slot is being generated (e.g., head, predicate, or tail) and is implemented as a simple finite state machine. Each time the model generates a slot separator, the state counter advances, and when a triple separator is generated, the counter resets to zero. Before producing the next token, SLGM retrieves a format-specific mask based on the current generation state. This mask restricts the model to generate only legal tokens at that stage. For example, the loss prevents the model from generating a slot delimiter after a tail or an entity delimiter within an entity, by adding a large negative penalty to the logits of invalid tokens.

4 Experiments

Our experiments follow the DEEPSTRUCT (Wang et al. 2023) experiment protocol, involving two stages: structural pre-training and multi-task training. We outline our detailed experimental setup in Appendix C.

4.1 Tasks and Datasets

Structural Pre-training. For our structural pre-training, we adapt the TEKGEN training and KELM corpora (Agarwal et al. 2021), which include NER and RE examples. However, the corpora do not include information on named entity type, which we address via augmentation by mapping entity types from WikiData (Vrandečić and Krötzsch 2014) to a selected suite of frequent types. To supplement the missing type information in TEKGEN and KELM, we map WikiData entities and relations to six coarse-grained entity supertypes—Person, Location, Organization, Product, Terminology, and Event, and manually cluster frequent relation phrases into relation supertypes as described in Appendix B. The augmentation is discussed in more detail in Appendix B. In total, we compile 100k sentences from each task and corpus pair for a total of 400k sentences.

Multi-task training. After structural pre-training, we perform multi-task training using datasets outlined in Table 2. We present dataset statistics in Appendix B. Across 5 tasks: named entity recognition (NER), relation extraction (RE), semantic role labeling (SRL), intent detection (ID), and dialogue state tracking (DST), we employ a total of 13 datasets. Two datasets are joint entity and relation extraction (JER) datasets, performing NER and RE jointly. Our dataset formats follow those of TANL (Paolini et al. 2021) and DEEPSTRUCT (Wang et al. 2023). This stage lasts for 2 epochs on each dataset.

4.2 Baseline Models

We establish 3 baseline models across our experiments. The first, **CE**, is a Flan-T5-based (Chung et al. 2022) cross-entropy loss model, using SLGM-like input and output format. The second, **CE+task**, is the Flan-T5-based cross-entropy loss model, with TANL-style (Paolini et al. 2021) prompt with task information. The third, **CE+data**, is another Flan-T5-based cross-entropy loss model, but with DEEPSTRUCT-style (Wang et al. 2023) prompt that includes task and dataset information. The baseline models differ from **SLGM**, which includes a more transparent task and dataset information as explicit description of output format and tagset, and uses additional format loss and formatted decoding for stable structure generation. We outline a tabular comparison of SLGM and the three baselines in Table 3.

4.3 Evaluation Method

We report the performance and the number of format errors on each dataset. Performance on all tasks but DST is measured with micro F1 score. For joint entity and relation extraction, we measure F1 score for entity and relation extraction respectively. For dialogue state tracking task, we use joint accuracy score to measure performance.

Task	Dataset	Format
NER	CoNLL-03 (Tjong Kim Sang and De Meulder 2003) Ontonotes-v5 (Weischedel et al. 2013) GENIA (Kim et al. 2003) CoNLL-04 (Roth and Yih 2004) NYT (Riedel, Yao, and McCallum 2010)	<SOURCE> <;> instance of <;> *tagset* </>
RE	TACRED (Zhang et al. 2017) TACREV (Alt, Gabryszak, and Hennig 2020) CoNLL-04 (Roth and Yih 2004) NYT (Riedel, Yao, and McCallum 2010)	<SOURCE> <;> *tagset* <;> <SOURCE> </>
SRL	CoNLL-12 (Pradhan et al. 2012)	<SOURCE> <;> instance of <;> *tagset* </>
ID	ATIS (Hemphill, Godfrey, and Doddington 1990) SNIPS (Coucke et al. 2018)	intent <;> is <;> *tagset* </>
DST	MultiWOZ (Budzianowski et al. 2018)	[User] <;> <SOURCE> <;> <ANY> </>

Table 2: Task, dataset and task-specific formats. Joint entity and relation extraction corpus like CoNLL 2004 and NYT dataset split one sentence into two sentences, with different prompt and format.

	SLGM	CE	CE+task	CE+data
Task name			✓	✓
Dataset name				✓
Task instructions	✓	✓		
Dataset format	✓	✓		
Format loss	✓			
Form. decoding	✓			
Fine-tuning	None; ablated in Section 5.4.			

Table 3: An overview of what features and meta-information SLGM and the 3 baselines use.

We categorize format errors into three types: length mismatch, source mismatch, and tagset mismatch. Length mismatch means length of generated tuple does not match with given format. Source mismatch means the element of output violated `<SOURCE>` slot token restriction. When the model generates tokens that don't exist in source sentence, source mismatch count increases. Finally, a tagset mismatch occurs when the model generates an output that is not defined within the tags specified by the dataset.

5 Results and Discussion

In Section 5.1, we compare our **SLGM** framework to other baselines. In Sections 5.2 and 5.3, we discuss ablation on formatted decoding (FD) and the distribution of format errors with and without FD. In Sections 5.4 through 5.6, we perform ablation studies on fine-tuning, model size, and low-data scenarios. Sections 5.1 through 5.2 report an average over 5 runs across seed; Sections 5.4 and after report results from single runs. Full tables with SLGM and baseline performance on every dataset in every setting are shown in Appendix D.

5.1 Main results

We report our main results in Table 4, comparing our **SLGM** framework to three baselines: **CE** without any meta-information, **CE+task** with task information, and **CE+data** with task and dataset information. Training and predicting without dataset information (**CE**), performance drops by 11 points compared to when dataset information is included (**CE+data**), and the frequency of format errors increases 10-fold. **SLGM** performance excels against baseline models without dataset information, **CE** and **CE+task** and is comparable to those of dataset-aware **CE+data** on most datasets. Even without explicit dataset information, **SLGM** is also more reliable than our baseline models, generating fewer format errors than the dataset-aware **CE+data**. This suggests **SLGM** is likely to be robust in out-of-distribution prediction with varying data format or tagset, offering competent performance without even specific dataset information.

At the same time, we acknowledge that **CE** and **CE+task** models have no access to any tagset information, unlike **SLGM**, whose task instructions provide tagset information. Since there are multiple datasets for NER and JER, each with its unique tagset, format errors are more likely. On the other hand, RE and ID datasets share identical or similar tagsets, resulting in a low number of format errors. The same is observed in SRL and DST, with only one dataset each. In the following subsection, we further analyze the distribution of format errors and experiment to determine whether the use of formatted decoding as a means of providing indirect tagset information is effective for baseline models as well.

5.2 Ablation: Formatted Decoding at Inference

To investigate the impact of formatted decoding on generating the correct format, we ablate formatted decoding in our baseline and **SLGM** models. Table 5 shows the average score and average format errors across all datasets, with and without formatted decoding during inference. The numbers suggest that formatted decoding is helpful for extracting the

Task	Dataset	SLGM		CE		CE+task		CE+data		DEEPSTRUCT
		Score	FE	Score	FE	Score	FE	Score	FE	Score
NER	CoNLL-03	80.28	43	69.32	4700	12.06	10273	87.11	5	93.1
	OntoNotes	75.87	142	75.44	766	24.79	5143	81.12	348	87.6
	GENIA	69.88	5	67.05	147	65.26	70	66.48	30	80.2
RE	TACREV	71.39	2	66.95	6	66.52	21	67.04	13	-
	TACRED	63.11	2	58.41	6	59.51	21	59.07	13	74.9
JER	CoNLL-04	71.74	0	0.00	873	0.00	827	74.88	14	88.4
		27.87	2	13.31	413	23.08	491	48.53	46	72.8
	NYT	88.80	149	88.68	322	74.60	412	88.45	23	95.4
		59.36	20	65.80	17	45.07	464	67.47	7	93.7
SRL	CoNLL-12	83.45	0	82.47	161	82.37	158	82.35	159	60.6
ID	ATIS	93.96	0	94.09	11	94.30	15	94.21	9	97.3
	SNIPS	96.86	0	96.58	0	95.79	1	96.07	1	97.4
DST	MultiWOZ	38.87	0	38.16	667	36.68	0	37.18	0	53.5
Average		70.88	28.08	62.79	622.23	52.31	1376.62	73.07	51.38	82.9

Table 4: Main result of our experiment on multi-task setting. FE stands for ”Format Error”. Higher scores are better, while fewer format errors are better. Average scores and format errors are dataset-wise macro average. DEEPSTRUCT does not report the number of format errors.

correct tagsets. In the **CE** baseline model, despite having been trained without dataset information, formatted decoding led to an increase of 6 points in F1 score and a decrease of 94% in format errors.

However, when trained with dataset information, the model suffers from the absence of dataset information even with format information, as seen in **CE+task** performance with and without formatted decoding. This suggests that the model relies on dataset-specific distributions during training, and removing that signal at inference prevents it from generalizing properly. In this sense, formatted decoding acts as a structural inductive bias, guiding the model to produce correct formats, but it cannot fully compensate for missing dataset cues. Formatted decoding does increase F1 score and decrease the number of format errors, but the number of format errors is still high, compared to other baseline and **SLGM** models that use formatted decoding. From this, we find that when designing models with real-world applications in mind, including dataset names as a structure cue during training can lead to undesirable effects, hindering performance and generalization.

Without format-aware decoding, models trained with the format loss still achieved the highest scores and produced the fewest format errors. This indicates that, even when implicit, format loss provides a useful signal that helps the model infer the task, dataset type, and output structure. The slot losses contribute additional tagset-specific supervision. Because they are weighted more heavily than cross-entropy, which distributes gradients across all tokens, they exert stronger pressure on the model to learn which spans to extract and how to label them. Together, format loss not only guides the models toward a more explicit understanding of the sentence's structure, but also helps models produce well-formed structured outputs. We outline the full result on each dataset in Appendix D.1.

5.3 Format Error Analysis

We analyze the frequency of format errors produced by each model, both with and without formatted decoding. Figure 2 summarizes the total error counts and the distribution of error types. As shown in the figure, the absence of dataset information in **CE+task** most prominently leads to tagset mismatches: without dataset-specific cues, the model struggles to generate well-formed output strings. Conversely, when dataset information is provided, the model may overfit to the supplied tagsets and therefore exhibits more source prediction errors, as seen in **CE+data**. Formatted decoding substantially reduces tagset mismatches, though it does not eliminate them entirely because it constrains only individual tokens rather than full output sequences. Most remaining mismatches involve producing a shorter tag with equivalent meaning, like generating `LOC` instead of `LOCATION`. In **CE+data**, trained with explicit dataset information, source mismatches are more common. Because this model can reliably identify legal tagsets, its errors primarily reflect difficulties in predicting the correct source tokens, especially in NER.

5.4 Ablation Study: Fine-tuning

We next examine the effects of fine-tuning to SLGM. Since fine-tuning involves optimizing on a single dataset with clean tagset, we expect it to improve performance even for models without dataset information, reducing the comparative merit of **SLGM** to baseline models without dataset information **CE** and **CE+task**. Starting from the models trained in the

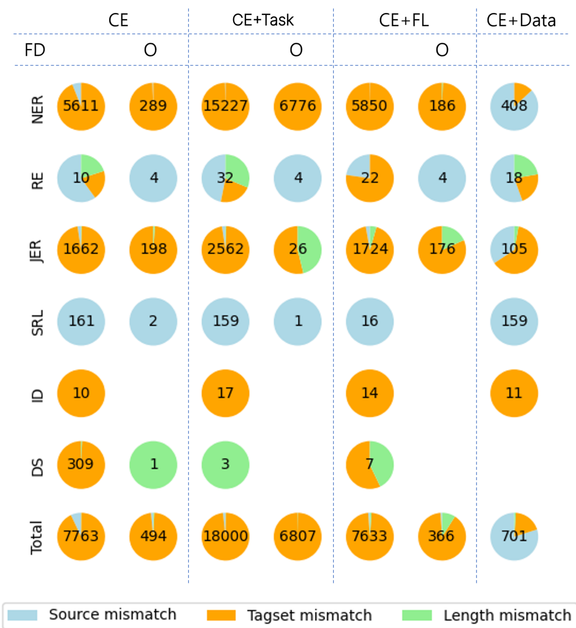

Figure 2: Format error frequency and ratio for each task. Numbers indicate the total number of FEs across every dataset inside given task.

Loss	Avg. F1	Avg. FE
CE	63.03	625
+ FD	69.87	45
+ FL	63.51	582
+ FL + FD (=SLGM)	**70.86**	**29**
CE+task	52.73	1405
+ FD	59.32	569

Table 5: Average score with respect to loss and inference method. FD is expressed as an indented row. SLGM corresponds to CE + Format Loss + Formatted Decoding.

multi-task setting, we fine-tune the models on each dataset for 5 epochs and evaluate their performance. The results are shown in Table 6. Compared to results with formatted decoding shown in Table 5, fine-tuning generally yields higher F1 scores. As expected, for **CE** with format loss and **SLGM**, formatted decoding contributes only marginal additional gains, suggesting a conceptual connection between formatted decoding and fine-tuning: while fine-tuning explicitly adapts the model to dataset-specific distributions, formatted decoding provides a lightweight mechanism that captures some of the same benefits when task-specific adaptation is not feasible. This supports our interpretation of formatted decoding as a partial surrogate for fine-tuning—similar in spirit to parameter-efficient adapters (Houlsby et al. 2019), which leave the base model unchanged while adding small task-specific components.

Loss	Avg. F1	Avg. FE
CE	53.10	742
+ FT	77.70	56
+ FL	66.74	504
+ FL + FT	78.93	16
CE+task	53.15	1404
+ FT	63.53	606
SLGM	73.37	19
+ FT	78.85	**1**
DEEPSTRUCT	82.9	-
+ FT	**84.9**	-

Table 6: Average score and format error with and without fine-tuning (+FT). SLGM corresponds to CE + FL + FD. DEEPSTRUCT (Wang et al. 2023) does not report the number of format errors.

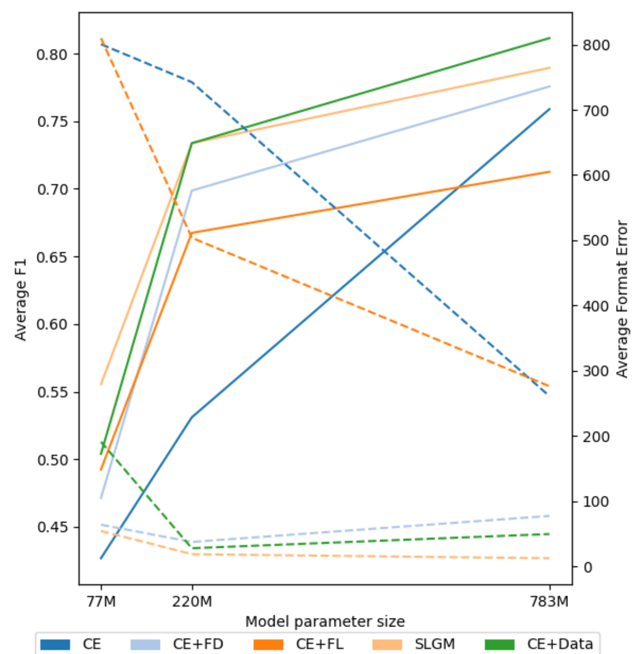

Figure 3: Average score and format error according to model size: `small`, `base`, and `large`. Solid line indicates Average F1 score (left y-axis). Dotted line indicates format errors (right y-axis).

5.5 Ablation Study: Model Size

We additionally evaluate the SLGM framework with `Flan-T5-small` (77M params) and `Flan-T5-large` with (0.8B), with the same hyperparameters as described in Section 4. We outline a summary of results in Figure 3.

On the `small` model, our method **SLGM** even outperforms **CE+data**, with access to dataset-specific information. The `small` model with a more limited capacity, appear to benefit from the stronger supervision provided by the format loss and formatted decoding, which offer more direct cues about the correct labels than plain cross-entropy. For the `large` model, a standard cross-entropy objective already yields substantially better results than `small`: with sufficient capacity, the model is able to infer dataset characteristics and retrieve the correct structure information without explicit cues. Interestingly, we observe that using the format loss alone with the `large` model can hurt performance. Using format loss and formatted decoding together, as done in **SLGM**, is still effective.

Using our best configuration, we achieve state-of-the-art performance on several benchmarks (Table 13). In particular, we obtain 94.8 micro-F1 on CoNLL-03 (Tjong Kim Sang and De Meulder 2003) NER and 98.3 micro-F1 on ATIS intent detection (Hemphill, Godfrey, and Doddington 1990). These findings indicate that our method can be further strengthened through fine-tuning, and that it can deliver strong performance even with smaller <1B-parameter models.

5.6 Ablation Study: Low-Resource Settings

Our analysis indicates that with SLGM, even smaller models are able to effectively encode structure in data and reliably retrieve entities, relations, and other structural information. To test the robustness of the framework, we also investigate its behavior in low-resource settings in the data side. We operationalize low-resource settings in two ways: 1) no structure pre-training and 2) training on a subset (1%, 5%, 10%, and 20%) of available data while preserving label distributions. We note that our implementation without structure pre-training is equivalent to the traditional NLP pipeline of fine-tuning after pre-training (Devlin et al. 2019; Linzen 2020). For each multi-task dataset, we label-proportionally sample training subsets, ensuring at least one instance of each label. In this ablation, we exclude MultiWOZ (Budzianowski et al. 2018) due to ambiguous label boundaries.

Figure 4 presents the results. Across all settings, structure pre-training is conducive to performance gains, echoing findings from prior work (Wang et al. 2023; Min et al. 2025). The finding indicates it contributes to the model's robustness and helps guide sentence interpretation when data is limited.

SLGM consistently outperforms other baseline models, with and without structural pre-training. SLGM reports higher performance and fewer format errors than the dataset-aware **CE+data** even under severe data constraints. We also find that, in low-resource scenarios, formatted decoding is generally more reliable than the format loss: without structural pre-training, larger training data does not improve **CE+FL**, **SLGM**, which rely on format loss. While performance remains stable across most datasets, we observe a sharp increase in malformed output and degeneration for the NYT relation extraction task (Riedel, Yao, and McCallum 2010), which we attribute to sparse supervision (Li et al. 2023).

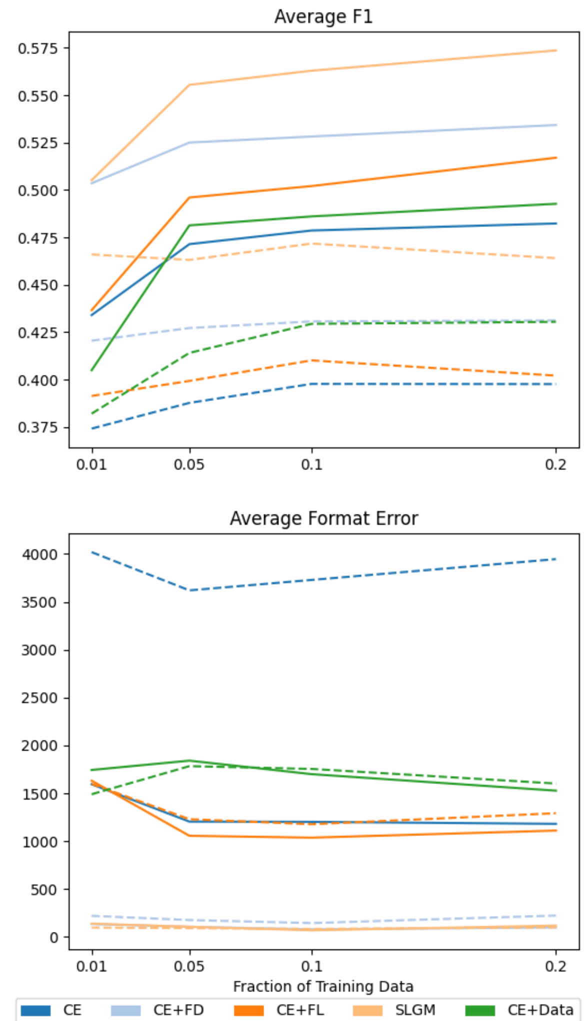

Figure 4: Average score and format errors according to dataset fraction. Solid line represents model with structural pre-training, and dotted line represents model without structural pre-training.

6 Conclusion

In this work, we introduce the Structured Language Generation Model, a novel framework for improving formatted structure prediction with generative pre-trained language models via structure-aware input formatting, loss design, and format-aware decoding. Our experiments on 13 datasets across named entity recognition, relation extraction, semantic role labeling, intent detection, and dialogue state tracking show that SLGM consistently enhances the structure prediction abilities of sub-billion-parameter models. SLGM achieves these gains without additional model parameters or dataset-specific engineering, and can act as a zero-weight adapter that approximates task-specific fine-tuning in low-resource settings. These results highlight the value of aligning GLMs' internal structural knowledge with their output space, and suggest a promising direction for building more robust, general-purpose structured prediction systems using generative models.

Limitations

Format loss and formatted decoding represent basic strategies designed to compel the model to produce outputs in accordance with a predetermined format. There may be many improvements on both scoring and real-world usage. There may be many problems in specific situations. Suppose we extract named entity with location but there is no location tag in format. The model expected a location and extracted some words, but the location tag is illegal. In plain decoding strategy, the model would generate any token similar to location. However, when using formatted decoding, it would generate unexpected string, which could be worse than extracting a wrong tag. We did not conduct quantitative experiments regarding this situation.

This scenario could potentially be addressed by employing a beam decoding strategy, which operates on triple units. This issue is related to the fact that the model does not know about format during actual attention calculation. Despite the explicit format being given, model cannot utilize format information at attention layers. We think additional cross attention layer over formats may show better results. We leave these possible improvements as future works.

Responsible Research Statement

In our research, we utilized a pretrained model and made use of publicly available datasets published on the web. We acknowledge that the data obtained from the web may contain potential biases. Our model can bear potential risks and harms discussed in Brown et al. (2020), and we clarify that our research did not specifically address or consider these risks. We ensured that all datasets employed in our study were accessed and used in a manner that respects their intended use and complies with any associated licenses or terms of service. We are also mindful of the potential biases present in these datasets and the pretrained model.

We used ChatGPT's GPT-4 (OpenAI 2024) as a debugging and text refining tool. We acknowledge and address the ethical considerations associated with the use of such AI technology, and have thoroughly reviewed the content to ensure that it does not include any unethical material.

Acknowledgements

This paper is based on our work performed at NC AI. Minho Lee designed and implemented the experiments. Minho Lee, Junghyun Min, and Woochul Lee collected datasets used in this work. Minho Lee, Junghyun Min, and Yerang Kim participated in writing. Woochul Lee and Yeonsoo Lee provided and high-level guidance in this work. We thank Chunghee Lee for his helpful comments.

References

Acciai, A.; Guerrisi, L.; Perconti, P.; Plebe, A.; Suriano, R.; and Velardi, A. 2025. Narrative coherence in neural language models. *Frontiers in Psychology*, 16: 1572076. Copyright © 2025 Acciai, Guerrisi, Perconti, Plebe, Suriano and Velardi.

Agarwal, O.; Ge, H.; Shakeri, S.; and Al-Rfou, R. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3554–3565. Online: Association for Computational Linguistics.

Alt, C.; Gabryszak, A.; and Hennig, L. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. arXiv:2004.14855.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. arXiv:2309.16609.

Bai, X.; Wu, J.; Chen, Y.; Wang, Z.; Chen, K.; Zhang, M.; and Zhang, Y. 2025. Constituency Parsing Using LLMs. *IEEE Transactions on Audio, Speech and Language Processing*, 33: 3762–3775.

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract Meaning Representation for Sembanking. In Pareja-Lora, A.; Liakata, M.; and Dipper, S., eds., *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. Sofia, Bulgaria: Association for Computational Linguistics.

Berglund, L.; Tong, M.; Kaufmann, M.; Balesni, M.; Stickland, A. C.; Korbak, T.; and Evans, O. 2024. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A". In *The Twelfth International Conference on Learning Representations*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5016–5026. Brussels, Belgium: Association for Computational Linguistics.

Choi, S.; in Hwang, J.; Noh, H.; and Lee, Y. 2021. May the Force Be with Your Copy Mechanism: Enhanced Supervised-Copy Method for Natural Language Generation. arXiv:2112.10360.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; Webson, A.; Gu, S. S.; Dai, Z.; Suzgun, M.; Chen, X.; Chowdhery, A.;

Castro-Ros, A.; Pellat, M.; Robinson, K.; Valter, D.; Narang, S.; Mishra, G.; Yu, A.; Zhao, V.; Huang, Y.; Dai, A.; Yu, H.; Petrov, S.; Chi, E. H.; Dean, J.; Devlin, J.; Roberts, A.; Zhou, D.; Le, Q. V.; and Wei, J. 2022. Scaling Instruction-Finetuned Language Models. arXiv:2210.11416.

Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; Primet, M.; and Dureau, J. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. arXiv:1805.10190.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, Y.; Jiang, X.; Liu, Y.; Li, G.; and Jin, Z. 2023. CodePAD: Sequence-based Code Generation with Pushdown Automaton. arXiv:2211.00818.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. arXiv:2103.10360.

Gehrmann, S.; Deng, Y.; and Rush, A. 2018. Bottom-Up Abstractive Summarization. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4098–4109. Brussels, Belgium: Association for Computational Linguistics.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. arXiv:1904.09324.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In Erk, K.; and Smith, N. A., eds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640. Berlin, Germany: Association for Computational Linguistics.

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

Herbold, S.; Hautli-Janisz, A.; Heuer, U.; Kikteva, Z.; and Trautsch, A. 2023. A large-scale comparison of human-written versus ChatGPT-generated essays. *Scientific Reports*, 13(1): 18617.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. arXiv:1902.00751.

Hovy, E. H. 1993. Automated discourse generation using discourse structure relations. *Artificial intelligence*, 63(1-2): 341–385.

Hu, B.; Somayajula, S. A.; Pan, X.; and Xie, P. 2025. Improving the Language Understanding Capabilities of Large Language Models Using Reinforcement Learning. arXiv:2410.11020.

Jin, Z.; Chen, Y.; Gonzalez Adauto, F.; Liu, J.; Zhang, J.; Michael, J.; Schölkopf, B.; and Diab, M. 2024. Analyzing the Role of Semantic Representations in the Era of Large Language Models. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 3781–3798. Mexico City, Mexico: Association for Computational Linguistics.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. arXiv:1909.05858.

Kim, J.-D.; Ohta, T.; Tateisi, Y.; and Tsujii, J. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1): i180–i182.

Kitouni, O.; Nolte, N.; Bouchacourt, D.; Williams, A.; Rabbat, M.; and Ibrahim, M. 2024. The Factorization Curse: Which Tokens You Predict Underlie the Reversal Curse and More. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 112329–112355. Curran Associates, Inc.

Lauriola, I.; Campese, S.; and Moschitti, A. 2025. Analyzing and Improving Coherence of Large Language Models in Question Answering. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 11740–11755. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Li, H.; Lan, T.; Fu, Z.; Cai, D.; Liu, L.; Collier, N.; Watanabe, T.; and Su, Y. 2023. Repetition In Repetition Out: Towards Understanding Neural Text Degeneration from the Data Perspective. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 72888–72903. Curran Associates, Inc.

Li, K.; Chen, C.; Quan, X.; Ling, Q.; and Song, Y. 2020. Conditional Augmentation for Aspect Term Extraction via Masked Sequence-to-Sequence Generation. arXiv:2004.14769.

Ling, W.; Grefenstette, E.; Hermann, K. M.; Kočiský, T.; Senior, A.; Wang, F.; and Blunsom, P. 2016. Latent Predictor Networks for Code Generation. arXiv:1603.06744.

Linzen, T. 2020. How Can We Accelerate Progress Towards Human-like Linguistic Generalization? In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5210–5217. Online: Association for Computational Linguistics.

Lu, X.; West, P.; Zellers, R.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2021. NeuroLogic Decoding: (Un)supervised Neural Text Generation with Predicate Logic Constraints. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4288–4299. Online: Association for Computational Linguistics.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 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)*, 5755–5772. Dublin, Ireland: Association for Computational Linguistics.

Min, J.; Lee, M.; Lee, W.; and Lee, Y. 2025. Punctuation Restoration Improves Structure Understanding without Supervision. In Adlakha, V.; Chronopoulou, A.; Li, X. L.; Majumder, B. P.; Shi, F.; and Vernikos, G., eds., *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, 120–130. Albuquerque, NM: Association for Computational Linguistics. ISBN 979-8-89176-245-9.

Min, J.; Yang, X.; and Wein, S. 2025. When Does Meaning Backfire? Investigating the Role of AMRs in NLI. In Frermann, L.; and Stevenson, M., eds., *Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, 202–211. Suzhou, China: Association for Computational Linguistics. ISBN 979-8-89176-340-1.

Olmedilla, M.; Romero, J. C.; Martínez-Torres, R.; Galván, N. R.; and Toral, S. 2024. Evaluating coherence in AI-generated text. In *Proceedings of the 6th International Conference on Advanced Research Methods and Analytics (CARMA 2024)*. Polytechnic University of Valencia.

OpenAI. 2024. ChatGPT. Available online.

Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; dos Santos, C. N.; Xiang, B.; and Soatto, S. 2021. Structured Prediction as Translation between Augmented Natural Languages. arXiv:2101.05779.

Park, C.; Ha, E.; Jeong, Y.; Kim, C.-y.; Yu, H.; and Sung, J.-w. 2023. CopyT5: Copy Mechanism and Post-Trained T5 for Speech-Aware Dialogue State Tracking System. In Chen, Y.-N.; Crook, P.; Galley, M.; Ghazarian, S.; Gunasekara, C.; Gupta, R.; Hedayatnia, B.; Kottur, S.; Moon, S.; and Zhang, C., eds., *Proceedings of The Eleventh Dialog System Technology Challenge*, 89–94. Prague, Czech Republic: Association for Computational Linguistics.

Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; and Zhang, Y. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint conference on EMNLP and CoNLL-shared task*, 1–40.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*, 148–163. Springer.

Roth, D.; and Yih, W.-t. 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*, 1–8.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083. Vancouver, Canada: Association for Computational Linguistics.

Sun, M.; Hua, W.; Liu, Z.; Wang, X.; Zheng, K.; and Li, P. 2020. A Predicate-Function-Argument Annotation of Natural Language for Open-Domain Information eXpression. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2140–2150. Online: Association for Computational Linguistics.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; Rouillard, L.; Mesnard, T.; Cideron, G.; bastien Grill, J.; Ramos, S.; Yvinec, E.; Casbon, M.; Pot, E.; Penchev, I.; Liu, G.; Visin, F.; Kenealy, K.; Beyer, L.; Zhai, X.; Tsitsulin, A.; Busa-Fekete, R.; Feng, A.; Sachdeva, N.; Coleman, B.; Gao, Y.; Mustafa, B.; Barr, I.; Parisotto, E.; Tian, D.; Eyal, M.; Cherry, C.; Peter, J.-T.; Sinopalnikov, D.; Bhupatiraju, S.; Agarwal, R.; Kazemi, M.; Malkin, D.; Kumar, R.; Vilar, D.; Brusilovsky, I.; Luo, J.; Steiner, A.; Friesen, A.; Sharma, A.; Sharma, A.; Gilady, A. M.; Goedeckemeyer, A.; Saade, A.; Feng, A.; Kolesnikov, A.; Bendebury, A.; Abdagic, A.; Vadi, A.; György, A.; Pinto, A. S.; Das, A.; Bapna, A.; Miech, A.; Yang, A.; Paterson, A.; Shenoy, A.; Chakrabarti, A.; Piot, B.; Wu, B.; Shahriari, B.; Petrini, B.; Chen, C.; Lan, C. L.; Choquette-Choo, C. A.; Carey, C.; Brick, C.; Deutsch, D.; Eisenbud, D.; Cattle, D.; Cheng, D.; Paparas, D.; Sreepathihalli, D. S.; Reid, D.; Tran, D.; Zelle, D.; Noland, E.; Huizenga, E.; Kharitonov, E.; Liu, F.; Amirkhanyan, G.; Cameron, G.; Hashemi, H.; Klimczak-Plucińska, H.; Singh, H.; Mehta, H.; Lehri, H. T.; Hazimeh, H.; Ballantyne, I.; Szpektor, I.; Nardini, I.; Pouget-Abadie, J.; Chan, J.; Stanton, J.; Wieting, J.; Lai, J.; Orbay, J.; Fernandez, J.; Newlan, J.; yeong Ji, J.; Singh, J.; Black, K.; Yu, K.; Hui, K.; Vodrahalli, K.; Greff, K.; Qiu, L.; Valentine, M.; Coelho, M.; Ritter, M.; Hoffman, M.; Watson, M.; Chaturvedi, M.; Moynihan, M.; Ma, M.; Babar, N.; Noy, N.; Byrd, N.; Roy, N.; Momchev, N.; Chauhan, N.; Sachdeva, N.; Bunyan, O.; Botarda, P.; Caron, P.; Rubenstein, P. K.; Culliton, P.; Schmid, P.; Sessa, P. G.; Xu, P.; Stanczyk, P.; Tafti, P.; Shivanna, R.; Wu, R.; Pan, R.; Rokni, R.; Willoughby, R.; Vallu, R.; Mullins, R.; Jerome, S.; Smoot, S.; Girgin, S.; Iqbal, S.; Reddy, S.; Sheth, S.; Põder, S.; Bhatnagar, S.; Panyam, S. R.; Eiger, S.; Zhang, S.; Liu, T.; Yacovone, T.; Liechty, T.; Kalra, U.; Evci, U.; Misra, V.; Roseberry, V.; Feinberg, V.; Kolesnikov, V.; Han, W.; Kwon, W.; Chen, X.; Chow, Y.; Zhu, Y.; Wei, Z.; Egyed, Z.; Cotruta, V.; Giang, M.; Kirk, P.; Rao, A.; Black, K.; Babar, N.; Lo, J.; Moreira, E.; Martins, L. G.; Sanseviero, O.; Gonzalez, L.; Gleicher, Z.; Warkentin, T.; Mirrokni, V.; Senter, E.; Collins, E.; Barral, J.; Ghahramani, Z.; Hadsell, R.; Matias, Y.; Sculley, D.; Petrov, S.; Fiedel, N.; Shazeer, N.; Vinyals, O.; Dean, J.; Hassabis, D.; Kavukcuoglu, K.; Farabet, C.; Buchatskaya, E.; Alayrac, J.-B.; Anil, R.; Dmitry; Lepikhin; Borgeaud, S.; Bachem, O.; Joulin, A.; Andreev, A.; Hardin, C.; Dadashi, R.; and Hussenot, L. 2025. Gemma 3 Technical Report. arXiv:2503.19786.

Tjong Kim Sang, E. F.; and De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

Wang, C.; Liu, X.; Chen, Z.; Hong, H.; Tang, J.; and Song, D. 2021a. Zero-Shot Information Extraction as a Unified Text-to-Triple Translation. arXiv:2109.11171.

Wang, C.; Liu, X.; Chen, Z.; Hong, H.; Tang, J.; and Song, D. 2023. DeepStruct: Pretraining of Language Models for Structure Prediction. arXiv:2205.10475.

Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; and Tu, K. 2021b. Automated Concatenation of Embeddings for Structured Prediction. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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)*, 2643–2660. Online: Association for Computational Linguistics.

Wang, X.; Peng, M.; Sun, M.; and Li, P. 2022. OIE@OIA: an Adaptable and Efficient Open Information Extraction Framework. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6213–6226. Dublin, Ireland: Association for Computational Linguistics.

Wein, S.; and Opitz, J. 2024. A Survey of AMR Applications. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6856–6875. Miami, Florida, USA: Association for Computational Linguistics.

Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; El-Bachouti, M.; Belvin, R.; and Houston, A. 2013. OntoNotes.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771.

Zeng, W.; Luo, W.; Fidler, S.; and Urtasun, R. 2016. Efficient Summarization with Read-Again and Copy Mechanism. arXiv:1611.03382.

Zhang, H.; Song, H.; Li, S.; Zhou, M.; and Song, D. 2023. A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models. *ACM Comput. Surv.*, 56(3).

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Zhong, Q.; Ding, L.; Liu, J.; Du, B.; and Tao, D. 2023. Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT. arXiv:2302.10198.

A GPT-4 Instructions

Among the results shown in Table 1 is GPT-4 (OpenAI 2024) performance. We use a custom prompt defining the task and label space, with two examples to show the expected output format. A full example prompt is shown below:

```
You are given multiple sentences. You have to extract relations between entities in given sentences.
For each sentence, extract every named entity that have relations first. Types of named entity must be one of "location", "organization", "other", "human". Then, extract relations between extracted named entities. Types of relation must be one of "kills", "lives in", "works for", "located in", and "organization based in".

You must FOLLOW given structure. When extracting named entities, you must extract in this form: [ENTITY] <;> instance of <;> [TYPE] . When there are multiple named entitis. separate with </> . When extracting relations, you must extract in this form: [HEAD] <;> [TYPE] <;> [TAIL] . Same with named entity extraction, if there are multiple relations, separate with </> .

For each sentence, first line should be result of named entity extraction, and second line should be result of relation extraction. When given multiple sentence(separated by empty line), pad empty line between extractions.

When given a file, each line contains single sentence.

Example

```

Type	Count
Location	10,594,011
Person	10,239,553
Organization	3,182,685
Product	2,233,553
Term	1,186,434
Event	514,083

Table 7: Manually compiled entity supertypes.

```
Input

John Wilkes Booth , who assassinated President
    Lincoln , was an actor .

The opera company performed at the Palace of Fine
     Arts , in San Francisco , on June 30 and
    July 1-2 , said Kevin O 'Brien , a spokesman
     for the theater.

<input end>

Output

John Wilkes Booth <;> instance of <;> human </>
    President Lincoln <;> instance of <;> human

John Wilkes Booth <;> kills <;> President Lincoln

Palace of Fine Arts <;> instance of <;> location
    </> San Francisco <;> instance of <;>
    location </> June 30 <;> instance of <;>
    other </> July 1-2 , <;> instance of <;>
    other </> Kevin O 'Brien <;> instance of <;>
     human

Palace of Fine Arts <;> located in <;> San
    Francisco

<output end>
```

With the given prompt, we measure GPT-4 performance on the first 40 examples of CoNLL-04 joint entity recognition and relation classification's test split (JER; Roth and Yih 2004).

B Details on Dataset Statistics and Mapping to Supertypes

Dataset statistics. Table 8 shows the statistics for each multi-task dataset. Note that TACRED (Zhang et al. 2017) and TACREV (Alt, Gabryszak, and Hennig 2020) share same number of sentences; their difference is on validation and test labels. When training with format loss, we remove sentences which have labels violating format errors, which accounts for less than 1% of the data. On evaluation, we used every sentences as-is, without manual validation.

Entity supertypes. In Section 4, we describe that entities and relation types have been mapped to supertypes. Among entity types, we filter out those that appear less than 20k times to remove the long tail. Then, we manually labeled them into 6 different entity types: Person, Location, Organization, Product, Terminology, and Event. Surface forms that we could not find from WikiData are not used. In Table 10, we illustrate examples of entity mapping for each entity supertype.

Relation supertypes. Similarly, we collect relations containing entities that have valid types mapped at Table 10. Among these relations, we filtered out relations that appear less than 10k times, again to remove the long tail and reduce noise. There were 168 relation phrases meeting these conditions. We additionally clustered 82 of 168 relation phrases into 35 similar clusters. Table 9 shows statistics of mapped relations.

Task	Dataset	Train	Dev	Test
NER	CoNLL-03	14,986	3,465	7,148
	OntoNotes-v5	75,187	9,603	9,479
	GENIA	15,023	1,669	1,854
JER	CoNLL-04	922	231	288
	NYT	56,196	5,000	5,000
RE	TACRED*	68,124	22,631	15,509
SRL	CoNLL-12	253,070	35,297	24,462
ID	ATIS	4,478	500	893
	SNIPS	13,084	700	700
DST	MultiWOZ	62,367	7,371	7,368

Table 8: Number of examples in each multi-task dataset. TACRED (Zhang et al. 2017) and TACREV (Alt, Gabryszak, and Hennig 2020) contain the same amount of data, but vary in their dev and test splits.

C Implementation Detail

We use the Flan-T5 (Chung et al. 2022), a family of instruction-tuned models based on T5 (Raffel et al. 2023) from HuggingFace hub (Wolf et al. 2020) as our base model. As shown in Figure 1, we used "Extract *target* from: *sentence*" as our input prompt, where *target* is determined by task. For **CE+task** and **CE+data** models, we add additional format information between task and sentence. Our models are based on the base sized checkpoint with 220M parameters, trained with batch size 16 per GPU on 4 NVIDIA A100 GPUs with 40G memory. For SLGM hyperparameters, we used $w_{ce} = 0.5$, $w_{st} = 0.2$, $w_{miss} = 0.33$, and $w_{sl} = 0.3$. We used greedy decoding when generating answers.

D Detailed experiment result

This section shows full result of Section 5. We report scores and format errors on every dataset for each settings.

D.1 Formatted decoding

Table 11 shows the full result using formatted decoding (Section 5.2), averaged on 5 runs. Comparing with Table 4, formatted decoding greatly reduces format errors on NER and

Type	Rel count	Type	Rel ount
country	3,233,250	member of	1,768,459
located in	1,657,574	product of	1,034,479
date of birth	1,009,859	occupation	937,797
instance of	879,955	place of birth	737,735
educated at	496,122	cast member	446,835
works at	353,896	date of death	300,792
awarded	299,818	contains	253,338
distributor	162,945	part of	146,352
gender	142,878	place of death	122,463
start date	110,483	sibling	92,805
child	84,473	parent	81,600
owner	77,977	genre	76,344
spouse	76,093	nominated	69,833
winner	61,688	position	60,142
created	49,232	background	42,186
capital	39,123	twin towns	38,671
capital of	31,669	president	20,402
family	15,840		

Table 9: Manually compiled relation supertypes. For example, relations ‘country’, ‘sovereign state’, ‘historical country’ are mapped to ‘country’.

Original type	Count	Map	Example
Human	9.7M	Person	Umberto II of Italy
Country	4.2M	Loc.	England
Taxon	1.2M	–	Natuna Island surili
Association football club	954k	Org.	FC Baden
Film	515k	Product	Avengers
Summer Olympic Games	185k	Event	2024 Summer Olympics

Table 10: Entity type mapping examples.

JER tasks. It also shows great score improvements on low resource data like CoNLL-04. Even if the model knows about dataset information, formatted decoding still upgrades score and reduces format errors.

D.2 Fine-tuning

Table 12 shows full result of fine-tuning (Section 5.4) on single run. Comparing with Table 11, fine-tuning shows 5 to 7 points better F1 score than formatted decoding. Surprisingly, best setting utilizing fine-tuning is using only format loss, which is better than using both formatted decoding. Model with format loss win over model with dataset information on almost every dataset. This is evidence that our format loss can reduce problem as easy as classification. However, using both fine-tuning and formatted decoding seems making conflict to each other. This is because some gold instances violates format restriction (e.g. `<SOURCE>` restriction in named entity recognition). Without formatted decoding, model can infer tokens that did not appeared in source sentence, when training example exists. However, we think this is not good for real-world usage, because it can be seen as overfitting on erroneous training examples.

D.3 Model size

Table 13 shows full result regarding to model size (Section 5.5) on single run. On cross-entropy model, model can distinguish characteristics of dataset when model parameters increase. The power of formatted inference on score is reduced when parameter size increased, yet still effective reducing format errors. Comparing CE+FL with CE, using only format loss showed worse score and more format error on large sized model. However, when mixed with formatted decoding, it showed additional increase comparing with models using single methods. Additionally, when we use every method we tried in this paper (i.e. format loss, formatted decoding, using bigger model, fine-tuning on dataset), we achieved state of the art performance on CoNLL-03 named entity recognition task and ATIS intent detection task. This means our framework still has room for improvement.

D.4 Low resource

Tables 14, 15, 16, and 17 show full result of low resource settings, without and with structural pre-training on single run. As mentioned earlier, when looking at result of CE+FL and SLGM with NYT dataset in Tables 14 and 15, there were weird explosion of format errors on some dataset fractions, which caused decreased score. We think this happened because without structural pretraining, model does not have enough structural understanding ability. With given small examples, the model tried to extract same entities multiple time when sentence goes longer.

Task	Dataset	CE		CE+FD		CE+FL		SLGM		CE+task+FD		CE+data+FD	
		Score	FE	Score	FE	Score	FE	Score	FE	Score	FE	Score	FE
NER	CoNLL-03	69.32	4700	78.09	79	66.17	5127	80.28	43	28.30	2196	87.11	0
	OntoNotes	75.44	766	76.04	288	75.50	572	75.87	142	26.72	5164	81.64	45
	GENIA	67.05	147	66.31	9	70.31	75	69.88	5	64.95	3	66.35	1
RE	TACREV	66.95	6	66.93	2	71.51	12	71.39	2	66.48	2	67.00	2
	TACRED	58.41	6	58.42	2	63.21	12	63.11	2	59.48	2	59.05	2
JER	CoNLL-04	0.00	873	66.73	0	0.00	873	71.74	0	66.02	0	74.50	0
		13.31	413	26.49	3	16.84	493	27.87	2	27.23	2	45.71	2
	NYT	88.68	322	88.45	156	88.84	300	88.80	149	73.79	17	88.36	1
		65.80	17	65.75	0	59.42	33	59.36	20	45.03	8	67.47	2
SRL	CoNLL-12	82.47	161	82.35	3	83.44	19	83.45	0	82.27	1	82.24	1
ID	ATIS	94.09	11	93.96	0	94.14	12	93.96	0	93.85	0	94.07	0
	SNIPS	96.58	0	96.58	0	96.86	0	96.86	0	95.72	0	96.01	0
DST	MultiWOZ	38.16	667	38.21	0	38.85	7	38.87	0	36.68	0	37.14	0
Average		62.79	622.23	69.56	41.69	63.47	579.62	70.88	28.08	58.96	568.85	72.82	4.31

Table 11: Full result of F1 scores and format errors regarding to formatted decoding.

Task	Dataset	CE+FT		CE+FD+FT		CE+FL+FT		SLGM+FT		CE+task+FT		CE+data+FT	
		Score	FE	Score	FE	Score	FE	Score	FE	Score	FE	Score	FE
NER	CoNLL-03	91.63	10	92.29	0	93.65	4	93.65	0	77.99	2733	91.65	7
	OntoNotes	82.26	376	82.23	4	84.22	11	84.21	0	41.93	2860	82.29	365
	GENIA	75.78	5	68.09	0	75.44	6	75.42	0	66.72	18	75.70	4
RE	TACREV	75.18	8	74.57	3	76.95	6	76.90	2	73.12	7	74.58	10
	TACRED	65.78	8	64.94	3	66.87	6	66.82	2	64.96	7	65.51	10
JER	CoNLL-04	83.50	2	82.68	0	85.60	3	85.48	0	0.00	877	84.86	1
		52.43	6	50.85	0	55.71	14	55.24	0	36.35	126	50.36	4
	NYT	90.11	11	90.08	0	90.66	13	90.68	0	85.28	15	90.36	20
		74.70	35	74.47	0	76.78	102	76.46	3	62.83	93	75.15	11
SRL	CoNLL-12	82.75	158	82.11	1	84.44	11	84.45	0	82.21	17	82.99	163
ID	ATIS	97.98	2	98.09	0	98.04	1	97.98	0	97.70	1	97.48	1
	SNIPS	98.43	0	98.43	0	96.72	0	96.72	0	98.00	0	98.15	0
DST	MultiWOZ	39.61	0	39.35	0	41.02	0	41.02	0	38.85	518	39.86	4
Average		77.70	47.77	76.78	0.85	78.93	13.62	78.85	0.54	63.53	559.38	77.61	46.15

Table 12: Full result of F1 scores and format errors regarding to dataset specific fine-tuning.

Task	Dataset	CE			CE+FD			CE+FL			SLGM			CE+data			SLGM+FT		
		S	B	L	S	B	L	S	B	L	S	B	L	S	B	L	S	B	L
NER	CoNLL-03	48.13	64.03	85.69	58.69	80.36	86.37	54.70	72.76	83.75	70.04	84.18	89.72	71.69	89.79	92.72	88.17	93.65	**94.80**
		5684	5843	1756	9	43	5	6108	4437	2531	36	25	18	374	15	45	0	0	0
	OntoNotes	48.80	71.89	84.46	49.46	78.20	86.36	61.89	81.30	85.02	61.73	81.64	85.33	66.95	83.84	87.28	76.87	84.21	87.89
		2080	717	555	360	191	42	1236	411	291	139	99	81	98	83	375	1	0	0
	GENIA	45.97	0.00	77.13	41.75	66.19	72.48	50.74	72.37	76.00	50.22	72.24	75.89	46.18	67.37	76.99	64.08	75.42	76.99
		193	1137	5	8	15	7	197	31	16	12	7	4	92	35	0	1	0	0
RE	TACREV	12.05	54.37	79.24	9.30	65.21	77.74	41.47	72.97	79.34	41.49	72.93	79.35	13.15	66.72	78.57	62.22	76.90	79.10
		7	5	9	8	2	2	56	7	9	4	2	4	4	7	14	4	2	2
	TACRED	10.72	49.18	69.34	8.24	56.92	67.53	36.81	64.35	69.23	36.80	64.32	69.25	10.86	57.70	68.73	55.35	66.82	69.24
		7	5	9	8	2	2	56	7	9	4	2	4	7	7	15	4	2	2
JER	CoNLL-04	0.00	0.00	40.31	49.30	67.57	75.22	0.00	6.88	40.74	57.55	71.17	77.39	48.24	74.43	85.56	77.16	85.48	88.18
		690	794	643	0	0	0	702	874	635	0	0	0	420	11	1	0	0	0
		0.00	4.87	45.77	7.98	27.28	45.07	0.00	23.32	38.61	9.32	33.90	44.84	0.00	46.43	59.69	26.00	55.24	60.02
		470	530	151	91	0	7	883	425	208	244	2	2	981	99	19	1	0	1
	NYT	80.06	85.35	93.59	79.68	88.26	88.36	81.24	89.79	93.66	81.25	89.80	93.69	82.78	88.83	93.76	85.10	90.68	93.85
		564	475	93	322	232	939	446	215	77	200	95	47	16	37	0	0	0	0
		45.68	54.79	84.66	45.20	66.74	84.98	29.99	65.01	85.22	29.63	64.75	85.21	49.72	66.01	86.06	38.95	76.46	86.66
		298	82	3	5	1	2	600	120	8	59	11	1	68	38	5	45	3	0
SRL	CoNLL-12	65.54	80.22	87.30	65.07	82.51	86.93	72.80	84.72	87.67	72.81	84.72	87.67	65.95	82.15	87.42	73.72	84.45	87.12
		167	20	161	2	1	0	83	14	9	4	0	1	164	16	162	4	0	1
ID	ATIS	79.81	92.64	97.87	81.38	93.72	97.87	85.25	96.51	97.92	85.78	96.52	97.76	83.92	94.36	97.92	96.19	97.98	**98.32**
		134	31	4	17	0	0	123	8	3	5	0	0	96	10	5	0	0	0
	SNIPS	89.68	95.22	97.50	89.87	96.43	97.43	92.86	96.72	97.43	93.01	96.72	97.43	88.24	96.86	97.43	96.29	96.72	98.15
		6	1	1	0	0	0	2	0	0	0	0	0	8	0	0	0	0	0
DST	MultiWOZ	28.42	37.75	43.72	26.88	38.89	42.00	32.53	40.96	42.72	32.58	40.96	42.72	27.56	39.33	42.69	35.08	41.02	42.23
		106	10	4	0	0	0	36	0	0	0	0	0	155	4	4	0	0	4
Average		42.68	53.10	75.89	47.14	69.87	77.57	49.25	66.74	75.18	55.56	73.37	78.94	50.40	73.37	81.14	67.32	78.85	81.73
		800.46	742.31	261.08	63.85	37.46	77.38	809.85	503.77	292.00	54.38	18.69	12.46	191.00	27.85	49.62	4.62	0.54	0.77

Table 13: Full result regarding to model size. S, B, L stands for small, base, large, respectively. For each dataset, upper row is F1 score, and lower row is format errors. Text with bold means state of the art performance.

Task	Dataset	CE				CE+FD				CE+FL			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
NER	CoNLL-03	26.19	42.80	42.71	45.30	51.65	55.67	54.10	55.24	24.72	41.60	44.79	44.32
		10169	7812	7820	7853	2	33	32	30	9937	7516	7820	7822
	OntoNotes	32.09	36.05	36.42	37.78	29.55	34.76	35.25	35.66	39.97	40.62	40.25	44.26
		3599	3442	3452	3392	480	752	612	586	4705	3628	3402	3472
	GENIA	42.69	45.87	45.62	42.71	38.07	39.58	39.54	38.24	45.61	44.59	46.95	46.48
		379	319	236	217	82	51	43	89	399	290	301	236
RE	TACREV	0.06	0.00	0.00	0.00	1.08	2.98	0.89	1.69	1.70	6.99	1.52	3.02
		30361	29257	30610	33136	1144	914	623	1264	39	17	15	20
	TACRED	7.87	5.39	10.67	7.10	3.68	2.78	1.31	0.42	1.48	6.07	1.43	2.78
		75	12	16	16	3	3	3	2	39	17	15	20
JER	CoNLL-04	0.00	0.00	0.00	0.00	39.00	41.08	41.64	41.35	0.00	0.00	0.00	0.00
		762	714	716	740	0	0	1	1	782	702	711	704
		0.00	0.00	0.00	0.00	0.44	1.34	1.92	1.85	0.00	0.00	0.00	0.00
		464	701	691	665	151	42	27	30	446	846	602	943
	NYT	77.64	80.34	80.66	78.57	77.22	80.24	80.32	78.35	79.33	81.23	80.92	80.56
		1143	341	459	788	739	244	363	622	1101	348	493	541
		48.18	31.60	33.65	30.24	47.83	31.15	32.04	28.94	52.19	25.60	41.50	20.19
		134	280	354	292	6	60	22	42	154	572	407	1482
SRL	CoNLL-12	49.40	56.52	60.59	65.31	48.36	56.17	60.26	64.83	54.16	60.45	64.43	68.33
		960	413	232	121	30	12	3	4	1393	680	190	120
ID	ATIS	79.54	80.07	79.45	80.64	81.20	79.48	80.99	80.59	82.70	82.68	81.39	82.29
		121	117	129	109	13	10	11	12	142	115	145	124
	SNIPS	85.37	86.69	87.56	89.57	86.59	87.45	88.59	90.30	87.84	89.37	89.02	90.34
		57	27	27	22	0	0	0	0	47	28	27	25
Average		37.42	38.78	39.78	39.77	42.05	42.72	43.07	43.12	39.14	39.93	41.02	40.21
		4018.67	3619.58	3728.50	3945.92	220.83	176.75	145.00	223.50	1598.67	1229.92	1177.33	1292.42

Table 14: Full result of low resource experiment without structural pre-training (Part 1): CE, CE+FD, and CE+FL. For each dataset, upper row is F1 score, and lower row is format errors.

Task	Dataset	SLGM				CE+data			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
NER	CoNLL-03	55.63	60.61	59.42	61.32	26.19	44.20	45.41	45.42
		14	12	15	20	10169	9828	9686	8658
	OntoNotes	36.00	38.80	38.78	41.39	32.09	42.05	39.90	43.55
		610	569	592	405	3599	7886	8326	7794
	GENIA	44.54	43.99	45.86	46.01	42.69	48.26	47.79	47.00
		57	48	27	68	379	858	879	586
RE	TACREV	1.76	7.04	1.52	3.02	11.49	4.70	12.18	9.75
		4	2	3	3	80	12	19	25
	TACRED	1.53	6.12	1.43	2.78	5.91	3.57	10.88	7.20
		4	2	3	3	16	33	49	16
JER	CoNLL-04	56.38	53.78	53.77	56.11	0.00	0.00	0.00	0.00
		1	15	2	0	762	632	646	645
		4.31	3.92	5.00	2.52	0.00	0.00	0.00	0.00
		19	16	19	29	464	611	506	554
	NYT	79.21	81.17	80.64	80.40	77.64	82.85	82.66	82.03
		440	224	310	369	1143	252	363	546
		51.94	25.46	41.13	19.76	48.18	37.27	39.00	38.39
		1	129	17	293	134	188	127	143
SRL	CoNLL-12	53.68	60.35	64.48	68.34	49.40	60.39	63.50	68.33
		11	78	3	3	960	1005	350	159
ID	ATIS	85.18	84.09	84.22	84.09	79.54	85.85	85.66	84.77
		9	12	10	5	121	92	93	94
	SNIPS	89.02	90.44	89.87	91.16	85.37	87.80	88.33	90.17
		0	0	0	0	57	8	14	9
Average		46.60	46.31	47.18	46.41	38.21	41.41	42.94	43.05
		97.50	92.25	83.42	99.83	1490.33	1783.75	1754.83	1602.42

Table 15: Full result of low resource experiment without structural pre-training (Part 2): SLGM and CE+data. For each dataset, upper row is F1 score, and lower row is format errors.

Task	Dataset	CE				CE+FD				CE+FL			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
NER	CoNLL-03	25.65	44.89	43.35	44.44	51.74	57.90	57.62	57.52	25.80	45.63	45.04	47.94
		10480	8083	7859	7781	8	25	21	26	10184	7603	7353	7420
	OntoNotes	36.26	41.34	42.19	44.24	36.33	41.39	41.35	43.11	39.09	44.56	44.52	48.91
		3108	2397	2641	2741	369	664	452	539	3884	2200	2510	3069
	GENIA	45.23	51.49	53.09	48.07	41.38	45.92	47.78	45.43	47.84	53.07	55.02	54.70
		528	347	335	300	179	118	88	86	552	372	277	230
RE	TACREV	33.68	39.19	43.60	42.50	31.84	35.56	34.53	33.29	20.42	40.41	40.98	44.76
		94	33	91	59	4	3	3	3	118	68	71	100
	TACRED	38.95	44.61	45.96	44.77	38.59	44.51	46.03	44.24	39.56	45.99	46.20	45.84
		597	579	568	588	3	12	12	3	622	73	31	83
JER	CoNLL-04	0.00	0.00	0.00	0.00	59.66	55.38	55.57	57.02	0.00	0.00	0.00	0.00
		758	758	702	756	0	2	1	1	741	710	731	789
		0.47	0.00	0.00	0.00	4.45	7.03	6.47	8.11	0.00	0.00	0.00	0.00
		656	1037	993	693	208	152	12	12	686	648	593	547
	NYT	78.43	81.58	81.09	79.99	78.19	81.12	81.05	79.72	79.90	83.02	82.95	82.15
		1214	417	421	754	744	158	241	442	1036	326	338	699
		41.44	26.18	26.54	31.34	40.03	26.11	26.01	30.43	43.96	42.66	44.27	48.97
		387	347	470	262	61	114	99	60	458	242	216	180
SRL	CoNLL-12	52.57	61.29	64.54	68.26	51.64	61.03	64.41	67.93	55.36	63.31	66.76	70.22
		1140	353	230	137	6	5	5	1	1182	344	229	141
ID	ATIS	82.21	84.14	83.07	83.76	83.82	83.63	82.43	82.99	83.40	84.56	84.62	85.13
		124	98	105	95	35	19	20	20	107	85	91	73
	SNIPS	85.97	91.03	90.95	91.45	86.59	90.44	90.58	91.30	88.62	92.04	92.13	91.78
		50	9	10	11	0	0	0	0	14	7	4	3
Average		43.41	47.15	47.86	48.23	50.36	52.50	52.82	53.42	43.66	49.60	50.21	51.70
		1594.67	1204.83	1202.08	1181.42	134.75	106.00	79.50	99.42	1632.00	1056.50	1037.00	1111.17

Table 16: Full result of low resource experiment with structural pre-training (Part 1): CE, CE+FD, and CE+FL. For each dataset, upper row is F1 score, and lower row is format errors.

Task	Dataset	SLGM				CE+data			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
NER	CoNLL-03	51.04	62.62	61.09	59.88	23.70	49.10	47.50	52.50
		32	28	29	27	10380	9046	8494	7328
	OntoNotes	38.13	45.45	44.35	46.46	35.89	45.29	44.14	46.36
		594	712	408	733	4272	8931	8713	8221
	GENIA	43.82	47.42	49.64	49.04	46.36	51.73	54.35	52.91
		200	122	97	135	594	841	748	476
RE	TACREV	20.99	40.30	41.20	44.86	5.56	30.49	34.92	34.28
		2	2	2	2	37	22	53	35
	TACRED	38.82	45.92	46.26	46.38	28.07	30.90	25.74	21.07
		5	2	3	3	202	64	36	35
JER	CoNLL-04	61.46	57.19	58.31	58.93	0.00	0.00	0.00	0.00
		0	0	1	0	749	688	664	654
		3.72	5.12	6.64	7.79	0.00	0.00	0.00	0.00
		275	54	78	23	463	524	425	511
	NYT	79.53	83.09	82.86	81.93	79.35	82.34	82.33	82.07
		461	301	172	470	913	621	565	627
		42.64	41.54	42.81	47.67	47.58	47.69	52.20	52.71
		72	40	25	14	133	185	144	151
SRL	CoNLL-12	54.29	62.98	66.61	69.95	52.49	61.58	66.04	69.87
		5	4	2	1	3008	1082	467	214
ID	ATIS	83.18	83.00	83.66	83.66	80.95	86.82	84.73	87.15
		24	13	15	15	145	81	79	76
	SNIPS	88.59	91.87	92.01	91.73	86.01	91.67	91.35	92.34
		0	0	0	0	32	10	14	5
Average		50.52	55.54	56.29	57.36	40.50	48.13	48.61	49.27
		139.17	106.50	69.33	118.58	1744.00	1841.25	1700.17	1527.75

Table 17: Full result of low resource experiment with structural pre-training (Part 2): SLGM and CE+data. For each dataset, upper row is F1 score, and lower row is format errors.