Order-Agnostic Data Augmentation for Few-Shot Named Entity Recognition

Huiming Wang[*1] **Liying Cheng**[2] **Wenxuan Zhang**[2] **De Wen Soh**[1] **Lidong Bing**[2]
[1]Singapore University of Technology and Design, Singapore [2]DAMO Academy, Alibaba Group
huiming_wang@mymail.sutd.edu.sg dewen_soh@sutd.edu.sg
{liying.cheng, saike.zwx, l.bing}@alibaba-inc.com

Abstract

Data augmentation (DA) methods have been proven to be effective for pre-trained language models (PLMs) in low-resource settings, including few-shot named entity recognition (NER). However, existing NER DA techniques either perform rule-based manipulations on words that break the semantic coherence of the sentence, or exploit generative models for entity or context substitution, which requires a substantial amount of labeled data and contradicts the objective of operating in low-resource settings. In this work, we propose *order-agnostic* data augmentation (OADA), an alternative solution that exploits the often overlooked *order-agnostic* property in the training data construction phase of sequence-to-sequence NER methods for data augmentation. To effectively utilize the augmented data without suffering from the one-to-many issue, where multiple augmented target sequences exist for one single sentence, we further propose the use of ordering instructions and an innovative OADA-XE loss. Specifically, by treating each permutation of entity types as an ordering instruction, we rearrange the entity set accordingly, ensuring a distinct input-output pair, while OADA-XE assigns loss based on the best match between the target sequence and model predictions. We conduct comprehensive experiments and analyses across three major NER benchmarks and significantly enhance the few-shot capabilities of PLMs with OADA, in both fine-tuning and in-context learning scenarios.

1 Introduction

Named entity recognition (NER) (Tjong Kim Sang and De Meulder, 2003; Doddington et al., 2004) has been one of the most long-standing and fundamental tasks. However, the effectiveness of NER systems is often constrained by the need for substantial high-quality, annotated datasets, which are costly and labor-intensive to acquire, necessitating innovative approaches to data scarcity (Rijhwani et al., 2020; Yang and Katiyar, 2020).

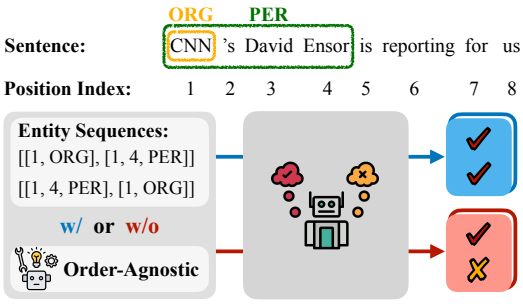


Figure 1: An example sentence from ACE-2005. In a **fixed order** setting (Yan et al., 2021), only one entity sequence ("[1, ORG], [1, 4, PER]") is deemed correct. However, **employing** the order-agnostic property encourages both entity sequences ("[1, ORG], [1, 4, PER]" and "[1, 4, PER], [1, ORG]") to be acceptable.

By introducing more reasonable samples, data augmentation (DA) methods have been proven to be effective solutions in scenarios where labeled data is scarce (Şahin and Steedman, 2018; Kobayashi, 2018; Wei and Zou, 2019). Existing DA approaches for NER can be roughly divided into two categories: (1) rule-based manipulations and (2) text-to-text generations. Among them, rule-based manipulations utilize predefined rules for automatic modifications in texts, including word deletion, reordering, and substitution (Min et al., 2020). However, due to the discrete nature of natural language, these techniques struggle to maintain the semantic coherence. Conversely, text-to-text DA techniques such as DAGA (Ding et al., 2020), MELM (Zhou et al., 2022), and ENTDA (Hu et al., 2023) augment texts by substituting entities or their contextual elements using predictions from pre-trained language models (PLMs). Nevertheless, to execute effective augmentations, these approaches require a substantial amount of labeled data to train the augmentation model for generating synthetic text, which poses a challenge in scenarios with scarcer labeled data, such as few-shot NER tasks,

[*]Work done while Huiming Wang was an intern at DAMO Academy, Alibaba Group.

and violates the fundamental goal of operating in low-resource settings.

In this work, we propose an innovative Order-Agnostic Data Augmentation framework OADA as an alternative solution, and demonstrate that utilizing a fundamental yet often overlooked aspect (i.e., the inherent unordered nature of target entities, which we refer to as the "order-agnostic property") of NER tasks can be extremely effective for data augmentation. To exploit this property for DA, we firstly notice the gap between the training data construction phase and prediction evaluation phase of traditional sequence-to-sequence (seq2seq) NER systems. As shown in Figure 1, given a source sentence "CNN's David Ensor is reporting for us", conventional seq2seq NER methods (Yan et al., 2021) will perceive entities with a fixed order and utilize only "[1, ORG], [1, 4, PER]" as the target sequence during training, while its equivalent "[1, 4, PER], [1, ORG]" is disregarded. However, when "[1, ORG], [1, 4, PER]" and "[1, 4, PER], [1, ORG]" are treated as the model predictions for evaluation, both of them will be recognized as correct generations. This fixed order assumption and the gap between training and evaluation phases cause the loss of many viable samples. Thus, in OADA, we seek to recognize the entities within a sentence as an unordered set, and treat different entity arrangements all as equivalent and also accurate generations. This perspective significantly broadens the range of acceptable target sequences for a given sentence, thereby introducing a novel and effective DA method. As shown in Figure 1, when incorporating the order-agnostic property into training data construction, both entity sequences are taken into consideration as reasonable target sequences.

In OADA, it is hypothesized that different entity arrangements provide equivalent information. Therefore, we further propose the use of ordering instructions and an innovative cross entropy (XE) loss OADA-XE to jointly fine-tune PLMs on these sequences together, without suffering from the one-to-many issue (Gu et al., 2018), where multiple possible entity sequences exist for the same sentence. For example, in Figure 1, if we make no distinction between "[1, ORG], [1, 4, PER]" and "[1, 4, PER], [1, ORG]" and pair them directly with the same input sentence, a PLM will have trouble deciding whether to generate "4" or "ORG" after "1". To address this issue, we show that our proposed strategies can effectively discriminate between different entity arrangements in two different aspects (i.e., inter-type and intra-type) as illustrated in Figure 2. Concretely, we first prioritize entity types as our primary factor of arranging entities. By treating each permutation of entity types like "LOC, ORG, MISC, PER" as an ordering instruction, we concatenate it with the input sentence and arrange the entity sequence following this instruction, ensuring a unique input-output pair. Second, within individual entity types, OADA-XE will align loss based on the best match between the target sequence and the model predictions. For example, in Figure 2, the first two predictions are both acceptable since it is possible to assign a match between them and the gold sequence, while the third prediction is wrong as it contradicts the ordering instruction and unruly rearranges the entities.

Categories	Methods	Coher.	Sub.	No-Train.
Text-to-text	DAGA	✓	✗	✗
	MELM	✓	✗	✗
	ENTDA	✓	✓	✗
Rule-based	Token Manipulation	✗	✓	✓
	Entity Replacement	✗	✗	✓
	OADA	✓	✓	✓

Table 1: Comparison of different DA methods. "Coher." means "Semantic Coherence", "Sub." means "Various NER Subtasks" and "No-Train." represents whether a method requires training additional models.

In summary, our work presents several key contributions, including: (1) To the best of our knowledge, we for the first time investigate to perform DA from the order-agnostic perspective and propose a novel DA framework OADA for few-shot NER. (2) We propose the ordering instruction and an innovative OADA-XE loss, which enable jointly fine-tuning PLMs on various entity arrangements together. (3) To demonstrate the effectiveness and generalization ability of OADA, we conduct comprehensive experiments and extensive analysis on three NER datasets, including one for nested NER task, with five representative PLMs (e.g., BERT (Devlin et al., 2019), BART (Lewis et al., 2020), Flan-T5 (Chung et al., 2022), LLaMA2 (Touvron et al., 2023) and ChatGPT (OpenAI, 2022)), and show notable improvements over established baselines.

2 Related Work

2.1 Data Augmentation for NER

As shown in Table 1, we compare OADA with two main categories of existing DA methods for NER.

Rule-based Manipulation Rule-based DA methods are primarily performing token-level manipulations, including synonym substitution (Wei and Zou, 2019; Cai et al., 2020), word deletion (Kobayashi, 2018) and reordering (Min et al., 2020). By utilizing their predefined rules, these methods can generate large amounts of synthetic texts efficiently. However, they might suffer seriously from the token-label misalignment issue (Zhou et al., 2022), where an entity token might be replaced with alternatives that mismatch its original label. In addition, due to the discrete nature of natural language, token-level manipulations will introduce incoherent replacement to the text.

To avoid the token-label misalignment issue, Dai and Adel (2020) proposed to randomly replace the whole entity mentions with alternative entities of the same type. Despite the effectiveness on flat NER tasks, its application to complex NER subtasks like nested NER remains challenging and they still inevitably introduce incoherent substitution (Hu et al., 2023). In this work, we make augmentations to only target entity sequences while preserving the input sentences intact, ensuring the semantic coherence of the texts.

Text-to-Text Generation Text-to-text generation based augmentations in NER tasks are mostly inspired by back-translation (Sennrich et al., 2016; Fadaee et al., 2017; Dong et al., 2017; Hou et al., 2018; Xia et al., 2019), which aims to transfer texts between languages while preserving their original meaning. When applied to token-level NER tasks, approaches like DAGA (Ding et al., 2020) and MELM (Zhou et al., 2022) explored to fine-tune PLMs using linearized sequences that merge token labels with tokens, and create augmented texts on corrupted sentences. However, these methods face challenges in handling nested entities, limiting their applicability across varied NER tasks. The more recent ENTDA (Hu et al., 2023) offers a context replacement strategy that adapts well to various NER tasks. Nevertheless, a common limitation among all these text-to-text generation methods is their reliance on a substantial amount of labeled data for training augmentation models, which contradicts the objective of low-resource settings.

On the contrary, by leveraging the order-agnostic property shared across NER tasks, our approach OADA possesses the strengths of both rule-based and text-to-text DA methods and can effectively augment data without the need for additional model training, while introducing no change to the sentence thereby maintaining the semantic coherence.

2.2 Few-Shot NER

Few-shot NER is a challenging task of identifying entities using only a small number of labeled examples (Wiseman and Stratos, 2019; Yang and Katiyar, 2020; Ding et al., 2021). Recent advancements in this field fall into mainly two categories: metric-based and prompt-based methods. This categorization also aligns with the two diverse few-shot NER settings. Metric-based methods aim to identify entities by learning a feature space and classifying test samples using nearest class prototypes or neighbor samples (Snell et al., 2017; Fritzler et al., 2018; Yang and Katiyar, 2020; Das et al., 2022; Zhang et al., 2023). However, most of these studies assume a rich-resource source domain, which is in contrast to the real world application scenarios that only very limited labeled data is available.

Following Gao et al. (2021), the more practical few-shot setting makes minimal assumptions about available resources and only provides few samples each class for training. By leveraging linguistic prompts to adapt PLMs to NER tasks, prompt-based methods (Cui et al., 2021; Ma et al., 2022; Lee et al., 2022; Shen et al., 2023; Xu et al., 2023) demonstrate impressive performance in few-shot scenarios, which can be further enhanced with more substantial data. In this work, we follow the more challenging few-shot setting where only few samples are provided for each entity type, and implement OADA over prompt-based methods, with a comprehensive comparison in Section 4.3.

Additionally, there has recently been a remarkable development in large language models (LLMs) such as GPT series (Brown et al., 2020; OpenAI, 2022), which show impressive capabilities in few-shot prompting and in-context learning (ICL) (Jimenez Gutierrez et al., 2022; Chen et al., 2023). Thus, in this work, we specifically examine OADA's generalization ability on LLMs and include the results in Table 4.

3 Order-Agnostic Data Augmentation

3.1 Formulation

The NER tasks aim at detecting all the spans that can represent entities within a given sentence X. The N entities in sentence X form the corresponding entity set $E = \{y_1, y_2, ..., y_N\}$. In OADA, we view the entities as the basic units and only

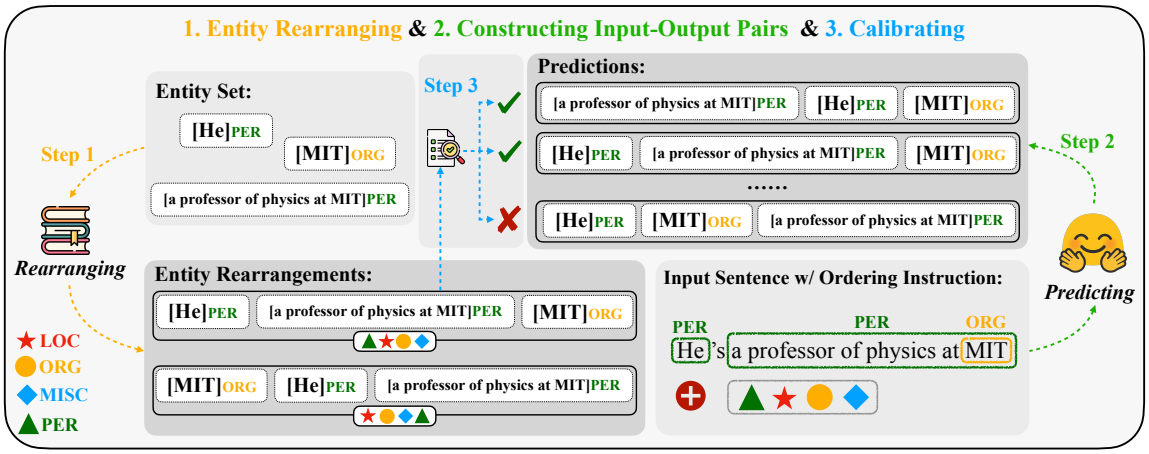


Figure 2: Overview of our proposed OADA. **Step 1: Entity Rearranging.** For each specific permutation of entity types, the entities in the entity set will be grouped by their types and be arranged into a unique rearrangement accordingly. **Step 2: Constructing Input-Output Pairs.** We view each permutation of entity types as an ordering instruction and concatenate it with the input sentence. The entity rearrangement following this permutation will be uniquely paired with the input sequence, as the input-output pair. **Step 3: Calibrating.** We propose OADA-XE and measure each prediction based on its best alignment with the target entity sequence.

perform entity-level rearrangement, preserving the integrity of individual entities. An entity y_i can be represented as a tuple $y_i = (s_i, t_i)$, where s_i, t_i represent the entity span and type of y_i respectively. The generation procedure can be formulated as:

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \log P(y_i|X, Y_{<i}). \quad (1)$$

3.2 Augmenting Data via Entity Rearranging

In OADA, we define that two entity sequences are equivalent if and only if they possess same entity sets, thus two equivalent sequences can vary greatly in how they arrange sequence components (i.e., entities). For the given sentence X, a target sequence Y^i is defined as a specific arrangement of entities from the set E, such as $Y^i = [y_{N-1}, y_N, ..., y_1]$. We further define the arrangement space as $\mathbf{O} = \{O^1, ..., O^I\}$, which encompasses all possible arrangements of entities from E. According to our definition, Y^i can be uniquely determined when provided with the corresponding O^i. This implies that the cardinality of the set $\mathbf{O}$, denoted as $I = |\mathbf{O}|$, directly controls the size of our augmented texts on E.

By randomly shuffling and rearranging the entity set E, we will acquire $N!$ different arrangements in $\mathbf{O}$, which are computationally infeasible to be included into training. Moreover, it is hard to simultaneously model the relations between X and a set of target sequences like Y^i via standard XE (Shao et al., 2019). To tackle this one-to-many issue, we introduce the ordering instructions, which complements the construction of input-output pairs and precisely controls the quantity of $\mathbf{O}$.

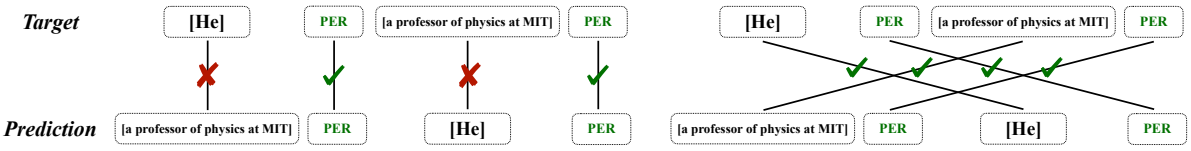
3.3 Constructing Unique Input-Output Pairs

In OADA, we firstly separate the one-to-many mapping into multiple one-to-one mappings in the inter-type aspect, and ensures unique input-output pairs.

Instead of augmenting data with all possible $N!$ arrangements, we propose an alternative strategy: prioritizing entity types as the primary factor of arranging entities.[1] Concretely, consider a certain dataset with a set of entity types $T = \{t_1, t_2, ..., t_l\}$ such as {**LOC**, **ORG**, **MISC**, **PER**}. By arranging entities based on their types, we can define $O_i \in \mathbf{O}$ as a random permutation p of entity types from T (e.g., [**PER**, **LOC**, **ORG**, **MISC**]). In accordance with the entity type arrangement p, entities in E are firstly grouped by their types and be subsequently arranged into Y_p by different groups. For example, in Figure 2, entities in type **PER** and entity "(**MIT**, **ORG**)" are firstly organized into two different groups. And these two groups are then arranged into different entity rearrangements following distinct ordering instructions.

To maintain unique one-to-one mappings between X and a set of its corresponding target sequences like Y_p, we view p as the unique ordering instruction and concatenate X with p as $[p; X]$. In the real generation procedure of a PLM, p will indicate to the model which entity type to focus on at a certain generation step. In this way, we address the one-to-many issue in the inter-type aspect, and the

[1] We discuss alternative rearranging factors in Appendix A.



(a) Standard XE (b) OADA-XE

Figure 3: Illustration of OADA-XE: (a) standard XE performing a per-position penalty, (b) OADA-XE which calculating the loss based on the best alignment between the predictions and the target sequence.

complexity of arrangement space **O** experiences a significant reduction from $\mathcal{O}(N!)$ to at most $\mathcal{O}(l!)$.

For example, in Figure 2, when paired with the ordering instruction "[**PER**, **LOC**, **ORG**, **MISC**]", the entity set can be uniquely rearranged into the target sequence "[(**He**, **PER**), (**a professor of physics at MIT**, **PER**), (**MIT**, **ORG**)]". Thus, the third prediction "[(**He**, **PER**), (**MIT**, **ORG**), (**a professor of physics at MIT**, **PER**)]" will be judged as a wrong prediction.

3.4 Calibrating Predictions with OADA-XE

In this section, we introduce how to alleviate the one-to-many issue within a certain entity type, namely intra-type issue. This issue will be raised since we can not discriminate the entities of a same type with only ordering instructions and standard XE. As an example, in Figure 2, the entities in the first prediction "[(**a professor of physics at MIT**, **PER**), (**He**, **PER**), (**MIT**, **ORG**)]" are precisely following the given ordering instruction, but will be penalized when performing standard XE, since the first two **PER** entities are not positionally aligned with those in the target entity sequence. Unlike inter-type mapping where entities possess different types and can be divided by their types, the only difference between these intra-type entities is their absolute position. Thus, the one-to-many mapping between these intra-type entities arises.

To mitigate this issue, we propose a novel XE loss for OADA as OADA-XE. As shown in Figure 3, standard XE loss requires a strict per-position match between target entities and model predictions, thus will heavily penalize the predicted sequence "[(**a professor of physics at MIT**, **PER**), (**He**, **PER**)]", although it is equivalent to the target entity sequence from our perspective. We define the OADA-XE objective as finding the best ordering $O^i \in \mathbf{O}$ to minimize the XE loss:

$$\mathcal{L}_{\text{OADA-XE}} = \underset{O^i \in \mathbf{O}}{\text{argmin}} \left(-\log P(O^i|X)\right). \quad (2)$$

If a best match can be found between the model prediction and the target entity sequence, this prediction will be regarded as a correct prediction. With OADA-XE, we successfully calibrate the predictions and prevent the intra-type issue.

To make it computationally feasible for Equation 2, we cast this problem as Maximum Bipartite Matching and leverage the efficient Hungarian algorithm (Kuhn, 1955), which reduces the time complexity from $\mathcal{O}(N!)$ to $\mathcal{O}(N^3)$ and is only 1.08 times slower than without OADA-XE in practice. We include the details of how we formulate the problem in Appendix B.

Furthermore, a large portion of rearrangements in **O** are invalid, and log loss requires that models assign probabilities to all potential sequences. Previous study (Kang and Hashimoto, 2020) demonstrated that log loss is sensitive to invalid or noisy sequences and can cause large changes in model behavior. Thus, we start the training with XE loss to make sure the model can effectively deal with the large search space of orderings **O**, and apply OADA-XE loss during training with an annealing schedule (Clark et al., 2019) to gradually teach the model to alleviate the per-position penalty:

$$\mathcal{L} = T_m * \mathcal{L}_1 + (1 - T_m) * \mathcal{L}_{\text{OADA-XE}}, \quad (3)$$

where T_m is the temperature at the m-th epoch of training, which progressively decreases to 0:

$$T_m = \max(0, 1 - c^{m - \lambda M}), \quad (4)$$

where M is the total training epochs, and c and λ are predefined hyperparameters, which we set to 16 and 0.95 throughout our paper.

4 Experiments

Datasets We conduct comprehensive experiments on two flat NER datasets and one nested NER dataset in several few-shot settings. For flat NER datasets, we choose CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and MIT-Movie (Liu et al., 2013) from two different domains. For MIT-Movie, we randomly select 15% samples from its training set as the development set.

Datasets	Models	F1 (Δ)
CoNLL-2003	BART-NER	77.63
	+Token Manipulation	78.38 ↑ 0.75
	+Entity Replacement	78.55 ↑ 0.92
	+DAGA	78.68 ↑ 1.05
	+MELM	79.06 ↑ 1.43
	+EnTDA	81.35 ↑ 3.72
	+OaDA	**82.91** ↑ 5.28
	+EnTDA+OaDA	**83.97** ↑ 6.34
ACE-2005	BART-NER	63.27
	+Token Manipulation	63.69 ↑ 0.42
	+Entity Replacement	63.81 ↑ 0.54
	+EnTDA	65.36 ↑ 2.09
	+OaDA	**66.93** ↑ 3.66
	+EnTDA+OaDA	**67.80** ↑ 4.53

Table 2: Performance comparison between different NER DA methods with 10% training data.

For nested NER, we conduct experiments on ACE-2005 (Doddington et al., 2004), using the same data split as Lu and Roth (2015).

4.1 Comparison with Different DA Methods

We firstly compare OaDA with other NER DA methods introduced in Section 2.1. To make fair comparison, we follow the low-resource setting in **EnTDA** where 10% training data is available, and the results are reported in Table 2. Among them, **DAGA** and **MELM** can not address nested entities and we only include their results on CoNLL-2003. From the results, we can observe that OaDA can be applied to various NER subtasks and achieves the greatest improvement compared with other NER DA methods. Besides, our approach also eliminates the need for training any additional models unlike text-to-text methods. In addition, it is important to note that the entities in EnTDA are still maintaining a fixed order. This implies that their augmented data can be further enhanced with OaDA as demonstrated by the results of "+EnTDA+OaDA", which also shows the generalization ability of our method.

4.2 Experimental Settings

As introduced in Section 2.2, in this work, we follow Ma et al. (2022) where only K samples of each entity type are provided. We conduct experiments in $K = \{5, 10, 20, 50\}$ settings for supervised fine-tuning, and $K = \{1, 2, 3, 5\}$ for ICL with LLMs. For all the settings, we adopt the same sampling strategy as Yang and Katiyar (2020) and report the mean and deviation performance over three splits. To demonstrate that OaDA can be uniformly applied to different models and even other few-shot methods, we implement OaDA over BERT, PromptNER (Shen et al., 2023), BART, BART-NER (Yan et al., 2021)) for supervised fine-tuning, and perform ICL over Flan-T5-XXL, LLaMA2-13B-Chat and ChatGPT.[2]

We compare OaDA with several strong and competitive few-shot methods: Template-NER (Cui et al., 2021), BART-NER, SEE-Few (Yang et al., 2022), Ent-LM (Ma et al., 2022), FIT (Xu et al., 2023) and PromptNER. Please refer to Appendix C for a detailed introduction.

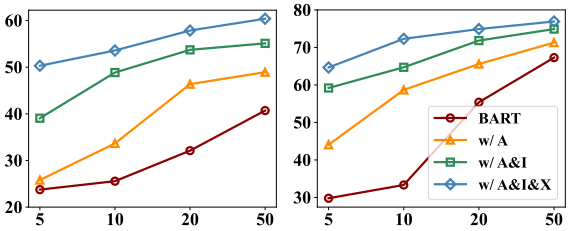
4.3 Main Results

Table 3 shows the results of our proposed OaDA compared with these baselines. Based on the results, we have the following observations: **(1) OaDA consistently improves the performance of both discriminative and generative PLMs.** Despite the varying baseline performances of models like BERT and BART (e.g., BERT's superiority in 10 out of 12 settings), OaDA achieves notable improvements across all few-shot settings. Particularly in 5-shot settings, OaDA enhances BERT's F1 score by 22.60 and 14.29 on CoNLL-2003 and ACE-2005, and those of BART by 16.26 and 22.07, respectively. **(2) OaDA can be generally applied to various tagging schemes, further advancing the capabilities of existing few-shot NER methods.** To further demonstrate the generalization ability of our approach, we also apply OaDA to previous SOTA few-shot NER methods with different tagging schemes. For instance, while BART-NER utilizes start and end indexes to represent entity spans which differs significantly from its pretraining corpus, OaDA effectively adapts to this scheme. The enhanced performance of methods like PromptNER and BART-NER with OaDA underscores its generalization potential across diverse NER applications. **(3) Among the prompt-based methods, OaDA shows to be the most efficient and effective.** Typical prompt-based methods Template-NER and FIT suffer from slow inference due to span enumeration, while our inference speed is 20.17x and 15.90x faster compared to them. PromptNER, while efficient in inference, requires lengthy sequences combining multiple templates with the input sentence, leading to increased mem-

[2]We implement OaDA over BERT based on the seq2seq LM architecture of UniLM (Dong et al., 2019) and also compare with BERT of sequence-tagging.

Models / Datasets	CoNLL-2003				MIT-Movie				ACE-2005			
	K=5	*K=10*	*K=20*	*K=50*	*K=5*	*K=10*	*K=20*	*K=50*	*K=5*	*K=10*	*K=20*	*K=50*
BERT	36.79	43.25	60.57	70.15	42.17	51.96	57.14	70.49	24.17	26.87	32.74	40.10
+BERT-tagger	41.87	59.91	68.66	73.20	39.57	50.60	59.34	71.33	–	–	–	–
+SEE-FEW	55.21	61.99	68.21	72.59	50.35	56.19	61.07	69.58	25.58	36.36	51.31	56.28
+Ent-LM	49.59	64.79	69.52	73.66	46.62	57.31	62.36	71.93	–	–	–	–
+Ent-LM+Struct	51.32	66.86	71.23	74.80	49.15	59.21	63.85	72.99	–	–	–	–
+FIT	45.02	59.24	63.85	68.84	52.21	58.78	63.46	71.43	37.74	42.25	52.71	56.11
+PromptNER	48.36	62.17	73.29	76.40	48.31	56.70	65.40	74.15	27.79	41.33	54.18	60.64
+OADA	59.39	67.15	75.51	77.05	54.80	64.71	70.23	76.41	38.46	42.94	56.19	63.47
+PromptNER+OADA	**63.61**	**70.20**	**77.76**	**80.76**	58.73	69.76	72.05	**78.19**	41.36	**45.58**	**56.81**	**65.11**
BART	36.08	42.67	54.61	59.16	43.64	48.75	58.96	69.64	18.06	25.23	28.53	31.40
+Template-NER	43.04	57.86	66.38	72.71	45.97	49.30	59.09	65.13	21.09	28.61	37.25	39.08
+BART-NER	38.03	43.09	59.26	69.09	50.43	58.53	65.87	70.99	18.87	31.04	41.54	51.81
+OADA	52.34	60.15	68.27	74.18	55.77	66.35	70.02	74.27	40.13	44.29	53.91	59.17
+BAER-NER+OADA	58.54	67.72	76.75	78.91	**62.21**	**70.17**	**73.18**	77.24	**43.25**	45.06	55.86	64.47

Table 3: Performance of fine-tuning on three datasets in different few-shot settings ($K = 5, 10, 20, 50$). We report the mean results (deviations in Appendix C) over 3 different splits for each cell.



(a) ACE-2005 (b) CoNLL-2003

Figure 4: Ablation studies of different components in OADA(BART) with F1 scores on the development sets of two datasets in $K = 5, 10, 20, 50$ settings reported. **A**: augmenting entity sequences (Section 3.2); **I**: using ordering instructions (Section 3.3); **X**: asigning loss with OADA-XE (Section 3.4).

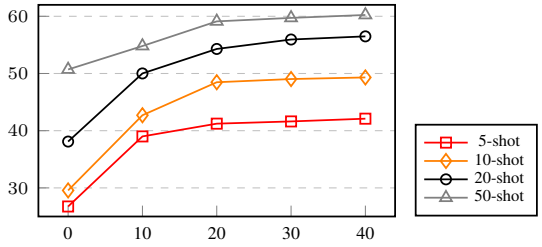


Figure 5: Performance of OADA(BART) on the valid set of ACE-2005 with different number of permutations.

ory demand. Our approach OADA, by leveraging the order-agnostic property, not only streamlines this process but also achieves superior performance across different PLMs compared to these existing methods. A more detailed analysis to the computational efficiency is included in Appendix D.

5 Analysis

5.1 Ablation Study

We conduct ablation experiments on fine-tuning BART to analyze the contributions of individual components of OADA. Results in Figure 4 show that, while directly incorporating augmented data can already improve the performance, addressing one-to-many issue with our proposed ordering instructions and OADA-XE can further boost the effect by a large margin, which demonstrates the effectiveness of each component of OADA. Results of ablation experiments in fine-tuning BERT are included in Appendix C.

5.2 Analysis of Permutations on T

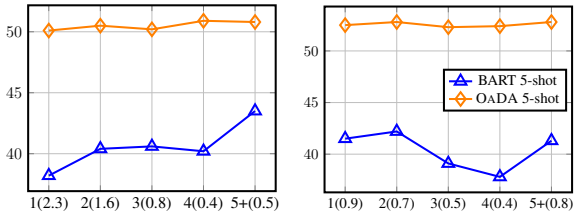
As introduced in Section 3.3, we choose to rearrange the entity sequences given the permutations of the entity types and reduce the arrangement space to at most $l!$, where l is the number of entity types. However, for the datasets like ACE-2005 and MIT-Movie with 7 and 12 entity types separately, acquiring $l!$ entity rearrangements is still impractical. Thus, we conduct experiments to investigate the effect of the number of permutations. From Figure 5, we can observe that most improvement comes from the first 20 permutations, and subsequent improvement is marginal with a great increase in training time. In real practice, we randomly select 20 rearrangements for ACE-2005 and MIT-Movie and all ($4! = 24$) for CoNLL-2003.

5.3 Entity Recall in Different Positions

We examine the recall performances of entities at different positions within a sentence. The detailed results are illustrated in Figure 6. For vanilla BART on flat CoNLL-2003 5-shot, entities appearing later are more likely to be recalled. Conversely, in nested ACE-2005, entities in the middle exhibit notably lower recall probabilities. We hypothesize that this pattern arises because, for flat NER, the dependen-

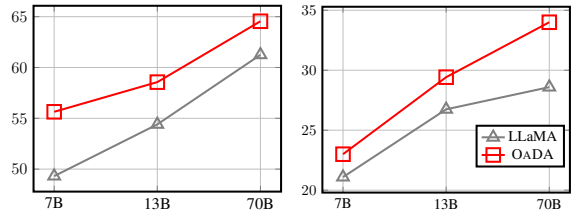
LLMs / Datasets	CoNLL-2003			MIT-Movie			ACE-2005		
	K=1	*K=2*	*K=3*	*K=1*	*K=2*	*K=3*	*K=1*	*K=2*	*K=3*
Flan-T5-XXL	52.33	56.30	61.74	47.87	53.50	56.41	20.34	24.79	26.25
+OADA	62.56	65.86	68.11	53.23	58.60	65.24	30.39	34.21	36.21
LLaMA2-13B	54.40	56.28	65.06	63.22	63.58	63.70	26.74	26.97	28.13
+OADA	58.56	61.94	67.88	66.60	67.48	70.01	28.95	29.42	38.34
ChatGPT	65.96	80.27	81.33	72.65	76.78	77.85	40.43	44.28	44.61
+OADA	**67.63**	**80.96**	**82.10**	**73.71**	**77.23**	**78.31**	**43.49**	**45.75**	**46.91**

Table 4: Performance of ICL with LLMs in ($K = 1, 2, 3$)-shot settings (deviations in Appendix F.2).



(a) CoNLL-2003 (b) ACE-2005

Figure 6: The recall of entities at different positions with the number of entities in that position in the bracket, over two development sets (the unit is 1000).



(a) CoNLL-2003 (b) ACE-2005

Figure 7: Performance comparison of LLaMA with different model scales, in the 1-shot setting.

cies among entities are less pronounced. While for nested NER, where latter entities may encompass the former entities, leading to a cascading effect where errors in earlier entities adversely impact the recall of subsequent ones. With OADA, we disrupt the arrangement of entities, so that subsequent entities are possible to appear before their preceding entities, thereby reducing the dependency of an entity's recall on previous entities. We further include some related case studies in Appendix E.

5.4 ICL of LLMs with OADA

Recently, there has been a remarkable development of LLMs. However, the huge number of parameters, coupled with their significant demand for computational resources, make ICL a more practical approach. To verify our applicability, we conduct experiments over distinct LLMs: Flan-T5-XXL, LLaMA2-13B and ChatGPT (i.e., gpt-3.5-turbo). The results are shown in Table 4. From the table, we can see that, although the results of LLMs are relatively higher than fine-tuning small-scale PLMs, OADA can be still valid and improve their performance further, which demonstrates the efficacy of OADA in utilizing ICL with LLMs. We include the details of our formatting and the ablation results of ICL in Appendix F.1 and F.3.

5.5 The Impact of Model Scales

We study how sensitive ICL with OADA is to the model scales as shown in Figure 7. Although we can observe significant performance improvements as the scale of LLaMA model increases, our method is consistently effective on these models with different scales. As shown in the results of LLaMA-70B, OADA can still be valid even when its model scale is much larger than other versions, and improves its base performance on CoNLL-2003 by more than 3.00 F1 scores. Besides, we can also observe from the results on CoNLL-2003 that, the larger base LLaMA models we use, the improvement coming from OADA appears to be slowly declining. We assume that this trend comes from the saturation of the demonstration used by ICL. Since the instruction tuning process of LLMs will incorporate data from NER tasks (Longpre et al., 2023), they show superiority performance on CoNLL-2003. To demonstrate this, we further employing OADA over LLaMA on ACE-2005 which LLMs will be less familiar with (Zhang et al., 2024). And the results indicate that our method shows good robustness in this scenario where more demonstrations can be helpful for LLMs to understand the tasks, rather than including extra but redundant demonstrations.

6 Conclusions

In this paper, we propose a novel data augmentation method OADA by leveraging the often-overlooked order-agnostic property of NER. Furthermore, to jointly utilize the augmented data together without suffering from the one-to-many issue, we introduce the use of ordering instructions and an innovative OADA-XE loss, tackling the issue in inter-type and

intra-type aspects separately. Experiments on three major NER benchmarks, and extensive analyses demonstrate the effectiveness of OADA.

Limitations

Despite the effectiveness of OADA, there are still some potential directions worth exploring and we leave as future work.

Reordering Factors In our work, to reduce the huge search space of **O** and provide a clear distinguishing criteria, we choose entity types as the basic reordering factor, and also discuss other potential candidates in Appendix A. From our analyses, we know that the original "left-to-right" order achieves the worst performance among them, which also demonstrates our claim that the strict order assumption does not need to be maintained. There are still other reordering factors worth discovering, which may further improve the effect. Maybe our current choice (i.e., entity types) is not the optimal solution, but our work also provides enough clues for subsequent work based on this, and the design we proposed to solve the one-to-many mapping problem will still have enough application scenarios in future work.

More Diverse Inference Strategies and Best Order During the training stage of OADA, we unlock the capability of generative models to perform diverse inference. As introduced in Section 3.3, we will construct a unique ordering instruction for each specific permutation of entity types to tackle inter-type mapping problem. Thus, for each input sentence, we will use $|\mathbf{O}|$ different ordering instructions, each indicating a specific generation priority. Besides, there are already some works (Mitchell et al., 2022; Wang et al., 2023) demonstrating that performing consistency-check will largely improve the performance of ICL with LLMs. Thus, we also conduct experiments on majority-voting based inference strategy and the results are shown in Appendix G. From the results, we found that the performance of applying majority-voting is not significantly better than the inference guided by the best ordering instruction. Furthermore, we also conduct some studies to select the best order and include them in Appendix H. From the results, we also can not observe significant improvement by selecting the "top-k" order. In our work, we consistently adopt all entity arrangements for fine-tuning and the original order for decoding, as reported in the main results of OADA. We believe that if there is a proper algorithm that can select entities inside all target sequences generated with different instructions, the performance will be further improved.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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 *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6334–6343, Online. Association for Computational Linguistics.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675, Toronto, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition us-

ing BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, M. Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Neural Information Processing Systems*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2018. Few-shot classification in named entity recognition task. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S. Yu. 2023. Entity-to-text based data augmentation for various named entity recognition tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9072–9087, Toronto, Canada. Association for Computational Linguistics.

Zhanming Jie and Wei Lu. 2019. Dependency-guided LSTM-CRF for named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3862–3872, Hong Kong, China. Association for Computational Linguistics.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

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

Jingjing Liu, Panupong Pasupat, Yining Wang, D. Scott Cyphers, and James R. Glass. 2013. Query understanding enhanced by hierarchical parsing structures. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77.

S. Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Large-scale language model for conversational ai. *OpenAI Blog*.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime Carbonell. 2020. Soft gazetteers for low-resource named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8118–8123, Online. Association for Computational Linguistics.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2019. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *AAAI Conference on Artificial Intelligence*.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Neural Information Processing Systems*.

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Yuanyuan Xu, Zeng Yang, Linhai Zhang, Deyu Zhou, Tiandeng Wu, and Rong Zhou. 2023. Focusing, bridging and prompting for few-shot nested named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2621–2637, Toronto, Canada. Association for Computational Linguistics.

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

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zeng Yang, Linhai Zhang, and Deyu Zhou. 2022. SEE-few: Seed, expand and entail for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2540–2550, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. 2024. In-context learning for few-shot nested named entity recognition. *ArXiv*, abs/2402.01182.

Shan Zhang, Bin Cao, Tianming Zhang, Yuqi Liu, and Jing Fan. 2023. Task-adaptive label dependency transfer for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3280–3293, Toronto, Canada. Association for Computational Linguistics.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.

Appendix

A Analysis of Rearranging Factors

In this paper, we propose to leverage the order-agnostic property for data augmentation. Unlike traditional NER systems that perceive entities as sequences with a fixed *left-to-right* order, we rearrange the entity set following a certain permutation of entity types. Another natural choice for rearrangement is based on the positions of entities, for example, specifying the 3rd entity in a *left-to-right* order to be generated at the beginning. Besides, some research (Jie and Lu, 2019; Yu et al., 2020) has demonstrate the effectiveness of dependency relationships in modeling the relations between the entities. Thus, we compare these four rearranging factors as shown in Figure 8. Since there is only one unique target sequence following a specific order (i.e., *left-to-right* or dependency relation), we only augment data with factors as positional information and entity types. Our findings indicate that rearranging entities based on types provides a clearer distinction among them, leading to enhanced performance. Furthermore, our proposed OADA-XE, addresses the one-to-many issue prevalent in both augmenting scenarios, resulting in improved effect. This evidence strongly supports our claim that adopting an order-agnostic approach, especially when combined with entity type-based rearrangement, significantly benefits NER systems.

B Solving OADA-XE as Maximum Bipartite Matching

In Section 3.4, to make it computationally feasible for the following equation:

$$\mathcal{L}_{\text{OADA-XE}} = \underset{O^i \in \mathbf{O}}{\operatorname{argmin}} \left(-\log P(O^i|X) \right),$$

we cast solving this problem as Maximum Bipartite Matching and leverage the efficient Hungarian algorithm (Kuhn, 1955), which reduces the time complexity from $\mathcal{O}(N!)$ to $\mathcal{O}(N^3)$. In our real practice, we perform matching based on entities and calculate the OADA-XE loss on the best matching between the target entities and the model predictions.

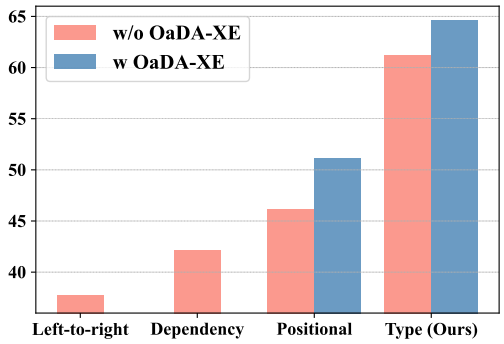


Figure 8: Comparison between different rearranging factors on CoNLL-2003 in 5-shot. We only apply OADA-XE on factors that can guide the augmentation.

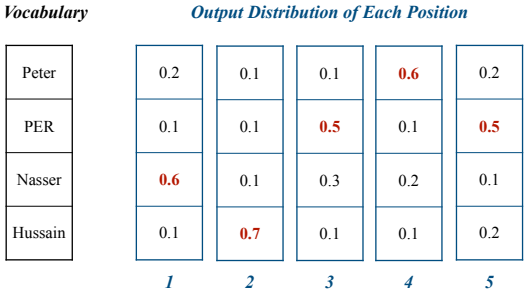


Figure 9: Illustration of bipartite matching to implement the proposed OADA-XE loss. We only show the probabilities of the words in target sequence for better understanding, and highlight in red the prediction selected by OADA-XE for each position. The best match is "[(***Nasser Hussain***, PER), (***Peter***, PER)]".

To simplify the description here, we state the situation that an entity sequence of length N is generated. As shown in Figure 9, we construct the bipartite graph $G = (U, V, E)$ where the vertices U are the set of N output positions and V are the set of N target tokens. Each edge in E is the prediction log probability for the token y in position n. Note that OADA-XE requires no changes to parallel decoding in inference, the training time of the OADA-XE loss is similar to (about 1.27 times slower than) training with cross entropy loss.

C Supplementary Results of Fine-Tuning

We compare OADA with several strong and competitive few-shot methods: **Template-NER** (Cui et al., 2021), the very first prompt-based method for few-shot NER, enumerates all spans within a sentence for entity typing, with extremely high time complexity. **BART-NER** formulates the NER task as an entity span sequence generation task, which can

Datasets	Models	*K=5*	*K=10*	*K=20*	*K=50*
CoNLL-2003	BERT	41.87 ± 9.13	59.91 ± 3.28	68.66 ± 2.35	73.20 ± 1.24
	SEE-Few	55.21 ± 3.93	61.99 ± 1.73	68.21 ± 2.60	72.59 ± 1.39
	Ent-LM	49.59 ± 8.30	64.79 ± 3.86	69.52 ± 4.48	73.66 ± 2.06
	Ent-LM + Struct	51.32 ± 7.67	66.86 ± 3.01	71.23 ± 3.91	74.80 ± 1.87
	PromptNER	48.36 ± 5.29	62.17 ± 2.23	73.29 ± 2.68	76.40 ± 2.47
	BART	36.08 ± 4.80	42.67 ± 5.32	54.61 ± 1.16	59.16 ± 0.18
	Template-NER	43.04 ± 5.15	57.86 ± 5.68	66.38 ± 6.09	72.71 ± 2.13
	BART-NER	38.03 ± 5.16	43.09 ± 7.38	59.26 ± 2.30	69.09 ± 2.19
	OADA (BERT)	59.39 ± 2.16	67.15 ± 1.02	75.51 ± 1.34	77.05 ± 1.68
	OADA (PromptNER)	**63.61 ± 2.16**	**70.20 ± 1.02**	**77.76 ± 1.34**	**80.76 ± 1.68**
	OADA (BART)	52.34 ± 5.39	60.15 ± 4.15	68.27 ± 0.51	74.18 ± 1.05
	OADA (BART-NER)	58.54 ± 3.02	67.72 ± 4.24	76.75 ± 1.85	78.91 ± 0.84
MIT-Movie	BERT	39.57 ± 4.09	50.60 ± 1.28	59.34 ± 0.68	71.33 ± 0.62
	SEE-Few	50.35 ± 6.28	56.19 ± 4.17	61.07 ± 2.09	69.58 ± 1.33
	Ent-LM	46.62 ± 9.46	57.31 ± 3.72	62.36 ± 4.14	71.93 ± 1.68
	Ent-LM + Struct	49.15 ± 8.91	59.21 ± 3.96	63.85 ± 3.70	72.99 ± 1.80
	PromptNER	48.31 ± 5.24	56.70 ± 3.25	65.40 ± 2.94	74.15 ± 2.19
	BART	36.08 ± 4.80	42.67 ± 5.32	54.61 ± 1.16	59.16 ± 0.18
	Template-NER	45.97 ± 3.86	49.30 ± 3.35	59.09 ± 0.35	65.13 ± 0.17
	BART-NER	50.43 ± 4.87	58.53 ± 1.09	65.87 ± 1.94	70.99 ± 1.84
	OADA (BERT)	54.80 ± 2.78	64.71 ± 1.35	70.23 ± 1.14	76.41 ± 0.97
	OADA (PromptNER)	58.73 ± 2.16	69.76 ± 1.02	72.05 ± 1.34	**78.19 ± 1.68**
	OADA (BART)	55.77 ± 3.80	66.35 ± 0.71	70.02 ± 0.23	74.27 ± 0.41
	OADA (BART-NER)	**62.21 ± 1.29**	**70.17 ± 1.58**	**73.18 ± 0.87**	77.24 ± 0.77
ACE-2005	SEE-Few	25.58 ± 4.13	36.36 ± 2.95	51.31 ± 2.08	56.28 ± 1.01
	FIT	37.74 ± 5.33	42.25 ± 10.65	52.71 ± 2.55	56.11 ± 1.75
	PromptNER	27.79 ± 7.41	41.33 ± 7.02	54.18 ± 2.57	60.64 ± 0.40
	BART	18.06 ± 1.32	25.23 ± 0.55	28.53 ± 1.05	31.40 ± 2.18
	Template-NER	21.09 ± 2.79	28.61 ± 2.15	37.25 ± 1.80	39.08 ± 1.22
	BART-NER	18.87 ± 3.33	31.04 ± 2.59	41.54 ± 2.16	51.81 ± 4.34
	OADA (PromptNER)	41.36 ± 1.47	**45.58 ± 3.28**	**56.81 ± 0.92**	**65.11 ± 0.17**
	OADA (BART)	40.13 ± 1.56	44.29 ± 0.61	53.19 ± 1.25	59.17 ± 1.08
	OADA (BART-NER)	**43.25 ± 0.24**	45.06 ± 1.34	55.86 ± 2.13	64.47 ± 1.81

Table 5: Performance of fine-tuning on three datasets in different few-shot settings ($K = 5, 10, 20, 50$).

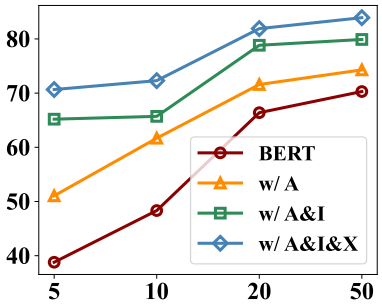


Figure 10: Ablation studies of different components in OADA(BERT) with F1 scores on the development sets of CoNLL-2003 in $K = 5, 10, 20, 50$ settings reported. **A**: augmenting entity sequences (Section 3.2); **I**: using ordering instructions (Section 3.3); **X**: asigning loss with OADA-XE (Section 3.4).

be directly applied to various NER subtasks. **SEE-Few** (Yang et al., 2022) reformulates span classification as a textual entailment task and leverages both the contextual clues and entity type information. **Ent-LM** (Ma et al., 2022) proposes a template-free prompt tuning method and induces the language models to predict label words at entity positions, while **Ent-LM+struct** leverages the viterbi algorithm to further boost the performance. **FIT** (Xu et al., 2023) is a focusing, bridging and prompting framework specially for few-shot nested NER, which also suffers from the huge time complexity because of enumeration.

Text: *staff sergeant tom ridge* opened fire . Entity: **[staff sergeant, PER], [staff sergeant tom ridge, PER]**
Model: BART Prediction: **[staff, ORG]**
Model: OADA(BART) Prediction: **[staff sergeant tom ridge, PER]**
Text: here is *cnn ' s candy crowley* with some war stories . Entity: **[cnn, ORG], [cnn ' s candy crowley, PER]**
Model: BART Prediction: **[cnn, GPE], [candy crowley, PER], [some war stories, ORG]**
Model: OADA(BART) Prediction: **[cnn, ORG], [cnn ' s candy crowley, PER], [war, WEA]**
Text: *he* ' s *a professor of psychiatry at nyu* , *chairman of the forensic panel* . Entity: **[he, PER], [a professor of psychiatry at nyu, PER], [nyu, ORG], [chairman of the forensic panel, PER]**
Model: BART Prediction: **[chairman of the forensic panel, PER]**
Model: OADA(BART) Prediction: **[he, PER], [a professor of psychiatry at nyu, PER], [nyu, ORG], [chairman of the forensic panel, PER]**

Table 9: Texts generated by BART with and without our proposed OADA in 50-shot settings, where [PER, ORG, GPE, WEA] represents entity types [person, organization, government, weapon] respectively.

Model	Time (s)	Memory (MB)	Performance (Δ)
BART-NER	3.4(1.00x)	14081(1.00x)	71.39
+OADA w/o X	32.5(9.56x)	14081(1.00x)	77.82(↑ 7.43)
+OADA	37.7(11.09x)	15179(1.08x)	80.95(↑ 9.56)
PromptNER	2.1(1.00x)	21572(1.00x)	76.58
+OADA w/o X	27.4(13.00x)	21572(1.00x)	80.15(↑ 3.57)
+OADA	33.1(15.76x)	23294(1.08x)	83.04(↑ 6.46)

Table 7: Computational cost of training on 50-shot samples from CoNLL-2003 and its performance on the development set, where X means OADA-XE.

PromptNER unifies entity locating and entity typing in one prompt and reduplicates this prompt for many times in the input sequence, which represent different entities. Detailed experimental results with both the mean values and deviations are reported in Table 5.

In Section 5.1, we provide the results of ablation study on fine-tuning BART. We also conduct ablation study over BERT and show the effectiveness of each component and the results are illustrated in Figure 10.

D Computational Efficiency

As we show in Section 4.3, OADA can be applied to different methods, including various tagging schemes. In addition, as a DA method, we introduce more reasonable data into training, resulting in the inevitable increase of training time. During inference, since we perform standard decoding, we share the similar time complexity with the backbones. Thus, to provide the detailed analyses to the computational efficiency of OADA, we provide the performance comparison in terms of training time, memory against backbone models as shown in Table 7 (with a single NVIDIA V100).

The increase in time is acceptable because the data size is enhanced by 24 times. Meanwhile, with OADA-XE, training time and memory have increased slightly, but it also brings considerable performance improvements. Moreover, as discussed in Section 4.3, PromptNER will combine the duplication of a template with the input sentence, leading to increased memory demand as shown in the table. By comparing the results of BERT-NER+OADA w/o X and PromptNER, our method can enhance the performance of BART-NER by a large margin and even make up for the gap in their performance, with an acceptable time increase.

E Case Study

In this section, we delve into case studies, as detailed in Table 9, and hope to provide some clues for understanding why OADA can help improve the performance of backbone PLMs. As discussed in Section 5.3, we found that without OADA, latter entities exhibit notably lower recall probabilities than former entities, leading to a cascading error effect.

In the first example sentence, vanilla BART will make a mistake and predict "staff" as an "ORG" entity, thus all the following entities will be disregarded. Conversely, our OaDA-enhanced approach, despite missing an internal entity, does not perpetuate this error. It successfully identifies the complete entity "staff sergeant tom ridge", demonstrating a more ro-

Datasets	Models	*K=1*	*K=2*	*K=3*	*K=5*
CoNLL-2003	LLaMA2	54.40 ± 3.63	56.28 ± 3.52	65.06 ± 1.45	61.07 ± 1.82
	Flan-T5	52.33 ± 7.06	56.30 ± 2.97	61.74 ± 2.32	60.65 ± 1.52
	ChatGPT	65.96 ± 2.19	80.27 ± 2.76	81.33 ± 1.97	82.79 ± 1.29
	OADA (LLaMA2)	58.56 ± 3.42 ↑ 4.16	61.94 ± 3.03 ↑ 5.66	67.88 ± 2.09 ↑ 2.82	61.78 ± 0.95 ↑ 0.71
	OADA (Flan-T5)	62.56 ± 3.31 ↑ 10.23	65.86 ± 0.84 ↑ 9.56	68.11 ± 1.79 ↑ 6.37	64.95 ± 1.49 ↑ 4.30
	OADA (ChatGPT)	**67.63** ± **2.10** ↑ 1.67	**80.96** ± **1.99** ↑ 0.69	**82.10** ± **2.09** ↑ 0.77	**83.85** ± **0.85** ↑ 1.06
MIT-Movie	LLaMA2	63.22 ± 1.67	63.58 ± 2.72	63.58 ± 5.89	61.51 ± 1.11
	Flan-T5	47.87 ± 0.64	53.50 ± 0.65	56.41 ± 0.55	54.88 ± 0.85
	ChatGPT	72.65 ± 0.85	76.78 ± 1.37	77.85 ± 2.05	79.10 ± 0.49
	OADA (LLaMA2)	66.60 ± 1.23 ↑ 3.38	67.48 ± 1.80 ↑ 3.90	70.00 ± 4.61 ↑ 6.42	64.43 ± 2.74 ↑ 2.92
	OADA (Flan-T5)	53.23 ± 1.48 ↑ 5.36	58.60 ± 0.94 ↑ 5.10	65.24 ± 0.66 ↑ 8.83	58.02 ± 0.66 ↑ 3.14
	OADA (ChatGPT)	**73.71** ± **0.96** ↑ 1.06	**77.23** ± **1.58** ↑ 0.45	**78.31** ± **2.17** ↑ 0.46	**80.13** ± **1.55** ↑ 1.03
ACE-2005	LLaMA2	26.74 ± 1.22	26.97 ± 1.82	34.83 ± 0.65	32.28 ± 1.80
	Flan-T5	20.34 ± 1.87	24.79 ± 1.60	26.25 ± 1.30	26.84 ± 0.60
	ChatGPT	40.43 ± 0.70	44.28 ± 1.38	44.61 ± 0.94	45.18 ± 1.55
	OADA (LLaMA2)	28.95 ± 0.96 ↑ 2.21	29.42 ± 1.87 ↑ 2.45	38.34 ± 1.92 ↑ 3.51	34.29 ± 0.67 ↑ 2.01
	OADA (Flan-T5)	30.39 ± 5.23 ↑ 10.05	34.21 ± 2.74 ↑ 9.42	36.21 ± 1.48 ↑ 9.96	32.82 ± 0.21 ↑ 5.98
	OADA (ChatGPT)	**43.49** ± **2.10** ↑ 3.06	**45.75** ± **0.47** ↑ 1.47	**46.91** ± **1.35** ↑ 2.30	**48.08** ± **0.90** ↑ 2.90

Table 8: Performance of ICL with LLMs in ($K = 1, 2, 3, 5$)-shot settings.

bust performance in nested NER scenarios. In another example "here is cnn ' s candy crowley with some war stories .", vanilla BART fails to recognize "cnn ' s candy crowley" as a singular entity due to its earlier misclassification of "cnn". Our approach, however, correctly identifies this compound entity, showcasing its superior entity recognition capabilities. Finally, in the sentence "he ' s a professor of psychiatry at nyu , chairman of the forensic panel .", vanilla BART even overlooks the entity "nyu" due to its previous error, while our method accurately identifies all entities within the sentence. Table 9 summarizes these examples, providing a clear comparison between the performance of vanilla BART and our OADA-enhanced model.

F In-Context Learning with OADA

In this section, we discuss how we perform ICL with OADA.

F.1 Input-Output Template for ICL

We provide the details of our prompts.

Instruction Prompt

Instruction: please extract entities and their types from the input sentence, all entity types are in options.

Instruction:

Instruction: Please extract entities and their types from the input sentence, all entity types are in options.
Options: PER, ORG, LOC, MISC.

Demonstrations:

Sentence: Marcelo Rios of Chile also advanced .
Following the order: LOC, PER, ORG, MISC
Entity: Chile is a LOC, Marcelo Rios is a PER.

Sentence: Marcelo Rios of Chile also advanced .
Following the order: PER, ORG, LOC, MISC
Entity: Marcelo Rios is a PER,Chile is a LOC.

Query:

Sentence: By stumps Kent reached 108 for three .
Following the order: LOC, PER, ORG, MISC
Entity:

Figure 11: Overview of input-output template for conducting ICL with OADA.

Option Prompt

Option: T.

Ordering Instruction Prompt

Following the order: p.

T and p are the set of entity types and a permutation of T. One example is provided in Figure 11.

Datasets	Models	*K=1*	*K=2*	*K=3*
CoNLL-2003	LLaMA2	54.40 ± 3.63	56.28 ± 3.52	65.06 ± 1.45
	Flan-T5	52.33 ± 7.06	56.30 ± 2.97	61.74 ± 2.32
	OADA (LLaMA2) w/o I	56.55 ± 3.39	59.05 ± 4.97	65.06 ± 1.45
	OADA (Flan-T5) w/o I	60.58 ± 2.03	62.79 ± 1.78	65.78 ± 1.82
	OADA (LLaMA2)	58.56 ± 3.42	61.94 ± 3.03	67.88 ± 2.09
	OADA (Flan-T5)	**62.56 ± 3.31**	**65.86 ± 0.84**	**68.11 ± 1.79**
MIT-Movie	LLaMA2	63.22 ± 1.67	63.58 ± 2.72	63.58 ± 5.89
	Flan-T5	47.87 ± 0.64	53.50 ± 0.65	56.41 ± 0.55
	OADA (LLaMA2) w/o I	61.66 ± 3.24	61.24 ± 2.72	64.74 ± 3.44
	OADA (Flan-T5) w/o I	51.24 ± 1.77	55.02 ± 1.26	56.67 ± 0.24
	OADA (LLaMA2)	**66.60 ± 1.23**	**67.48 ± 1.80**	**70.00 ± 4.61**
	OADA (Flan-T5)	53.23 ± 1.48	58.60 ± 0.94	65.24 ± 0.66
ACE-2005	LLaMA2	26.74 ± 1.22	26.97 ± 1.82	34.83 ± 0.65
	Flan-T5	20.34 ± 1.87	24.79 ± 1.60	26.25 ± 1.30
	OADA (LLaMA2) w/o I	27.63 ± 2.01	27.20 ± 3.16	35.28 ± 1.80
	OADA (Flan-T5) w/o I	25.08 ± 0.37	32.12 ± 2.41	32.83 ± 0.21
	OADA (LLaMA2)	28.95 ± 0.96	29.42 ± 1.87	**38.34 ± 1.92**
	OADA (Flan-T5)	**30.39 ± 5.23**	**34.21 ± 2.74**	36.21 ± 1.48

Table 9: Ablation experiments of ICL with LLMs in ($K = 1, 2, 3$)-shot settings. I: using ordering instructions.

	CoNLL-2003	ACE-2005
Random-Order$_{\text{Best}}$	74.81	56.07
Random-Order$_{\text{Worst}}$	73.46	53.88
Original-Order	74.20	55.18
Majority-Voting	74.32	55.41

Table 10: The comparison between different inference ordering on two datasets in 10-shot.

F.2 Supplementary Results of ICL

Detailed results of ICL with Flan-T5-XXL, LLaMA2-13B, ChatGPT are reported in Table 8. From the results, we can observe that although the performance of these LLMs is strong enough, OADA can still improve their effect.

F.3 Ablation Study on ICL

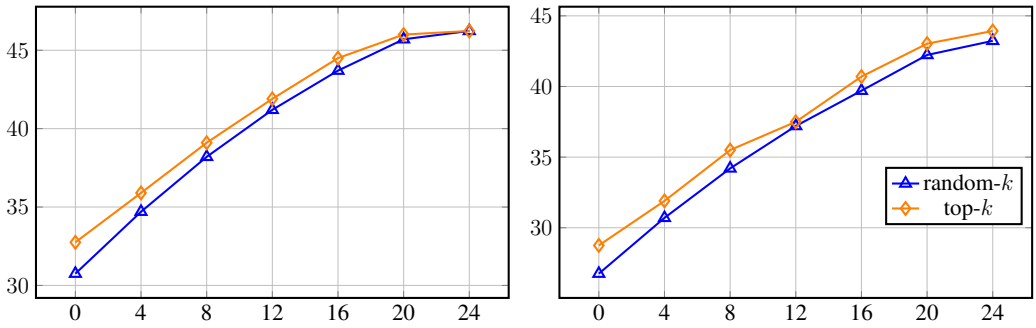
In this section, we conduct some ablation studies on ICL with LLMs. The results are shown in Table 9. From the results, we can see that while augmenting entity sequences can already improve the performance, the ordering instructions we used to prevent inter-type mapping also plays a crucial role in further boosting the performance. And the results also demonstrate that OADA can be easily applied to LLMs.

G Majority-Voting Inference

In Table 10, we compare 3 different ordering instructions for inference: 1. random, for this setting, we run experiments with 10 different random sampled instructions and report the best and worst performance. 2. original-order, we utilize "following the order: from left to right". 3. majority-voting, we perform majority-voting on the generated entity sequences guided by 10 randomly sampled instructions. If an entity appears in more than 5 sequences, it will be considered in the final entity set. From the result, we see that performing majority-voting will not increase the performance by a large margin, and is not efficient. Thus, in our main results, we only adopt the original order for inference, and leave this as future work.

H Best Order Selection

In our work, we regard OADA as a data augmentation method and propose the use of ordering instructions and OADA-XE to address the one-to-many issue. The basic assumption of utilizing all target sequences jointly is that they are equivalent to each other. In this section, we further discuss whether there could be a standing-out order that improves the performance of PLMs trained with the rearrangements of its guidance. Given a specific per-



(a) CoNLL-2003 (b) ACE-2005

Figure 12: Performance of training BART 5-shot with top-k permutations selected with averaged entropy, compared to training with k randomly selected permutations.

mutation of entity types, the output logits are utilized to compute the entropy on the target sequence arranged with this permutation:

$$\mathcal{P}(Y|X) = -\frac{1}{N}\sum_{N}\sum_{|V|} p \log p, \qquad (5)$$

where N is the number of entities in X and $|V|$ is the vocabulary size, p represents the output logits of the decoder.

For each sentence, its target sequence can be rearranged by $l!$ times, where l represents the number of entity types. Thus, we can select top-k different permutations based on the average entropy of all instances constructed by a certain permutation. We show the performance of training PLMs on these selected permutations in Figure 12. From the results, we can see that top-k can outperform random-k in nearly every setting. However, selecting the top-k permutations can not compete with utilizing more permutations, and they already calculated all the entropy loss based on the entity sequences in all orders, which means selecting top-k permutations will not bring a more efficient training or achieve better performance. Thus, we still adopt all ($4! = 24$ for CoNLL-2003, 20 for MIT-Movie and ACE-2005) permutations for training, rather than selecting the best orders.