PUnifiedNER: A Prompting-based Unified NER System for Diverse Datasets

Jinghui Lu[1], **Rui Zhao**[1], **Brian Mac Namee**[2,3], **Fei Tan**[1*]

[1]SenseTime Research
[2]The Insight Centre for Data Analytics, University College Dublin
[3]School of Computer Science, University College Dublin
{lujinghui1, zhaorui, tanfei}@sensetime.com, brian.macnamee@ucd.ie

Abstract

Much of named entity recognition (NER) research focuses on developing dataset-specific models based on data from the domain of interest, and a limited set of related entity types. This is frustrating as each new dataset requires a new model to be trained and stored. In this work, we present a "versatile" model—the **P**rompting-based **Unified NER** system (PUnifiedNER)—that works with data from different domains and can recognise up to 37 entity types simultaneously, and theoretically it could be as many as possible. By using prompt learning, PUnifiedNER is a novel approach that is able to jointly train across multiple corpora, implementing intelligent on-demand entity recognition. Experimental results show that PUnifiedNER leads to significant prediction benefits compared to dataset-specific models with impressively reduced model deployment costs. Furthermore, the performance of PUnifiedNER can achieve competitive or even better performance than state-of-the-art domain-specific methods for some datasets. We also perform comprehensive pilot and ablation studies to support in-depth analysis of each component in PUnifiedNER.

Introduction

The Named Entity Recognition (NER) task involves the automatic recognition of entities in text with specific meaning, and so includes both entity extraction and entity classification. The recent rise of transformer-based, pre-trained language models (Devlin et al. 2019; Raffel et al. 2020; Tan et al. 2020, 2021; Lu et al. 2022a; Mao et al. 2022) has led to significant performance improvements in NER for various scenarios (Fries et al. 2022; Parmar et al. 2022).

Modern NER models perform self-supervised learning on large unlabelled text datasets to learn generic linguistic representations, and then are fine-tuned for a specific NER task on domain-specific datasets. The output of these approaches, however, is still a single dataset-specific model capable of recognising a few entity types, rather than a "versatile" model that can handle multiple domains and a large number of entity types simultaneously. This is unsatisfactory in practice: for example, an NER model trained on an e-commerce dataset can only extract entity types related to the e-commerce domain such as "company" and "commodity" but not named entities from other domains such as "location", "organisation", etc. As a result, such NER methods can not scale up well as each new dataset demands a new model to be trained and stored.

We believe that it is more appealing to have a versatile model handle all scenarios. Besides, a unified model can achieve better performance than dataset-specific models if the unified model can be trained jointly, incorporating label information from different datasets. To be specific, although most NER datasets focus on corpora from various domains with different entity types, the underlying semantics or entity recognition are shared across corpora and exploiting this shared semantic information can enhance model robustness.

We hypothesise that this enhancement is derived from two aspects: commonality and diversity of label information. The enhancement from commonality refers to the fact that the model can see more relevant phrases from the same entity type. For example, several datasets with the "location" entity can capture most of the keywords belonging to the "location" entity in the real world. The enhancement from diversity means that the model can perceive more polysemous words. For example, "apple" could be a "company" entity (Apple Inc.) in a financial NER dataset, but may be a "commodity" entity (fruit apple) in an e-commerce dataset. Access to both senses during training will potentially strengthen the robustness of a model with regard to the polysemy problem. Moreover, when applying an NER model to specific text, users are not always interested in all of the entity types embedded in the text. Existing NER models typically always output all entity types that they can recognise, which is not an ideal delivery mode. One on-demand model that flexibly recognises requested entity types should be attractive to many users.

In this work, we develop a versatile model that can handle multiple NER datasets simultaneously, addressing all of the challenges mentioned above — *the **P**rompting-based **Unified NER** system (PUnifiedNER)*. Empowered by prompt learning, PUnifiedNER is built upon the recently proposed T5 language model (Raffel et al. 2020), and an overview of the model is shown in Figure 1. We jointly train the model on eight different datasets, supporting a total of 37 entity types. An obvious benefit of a prompting-based design is that the prompts can be served as instructions to guide the model to provide different outputs depending on the entity types of interest to the user, examples of which are shown in the red

*Corresponding author.

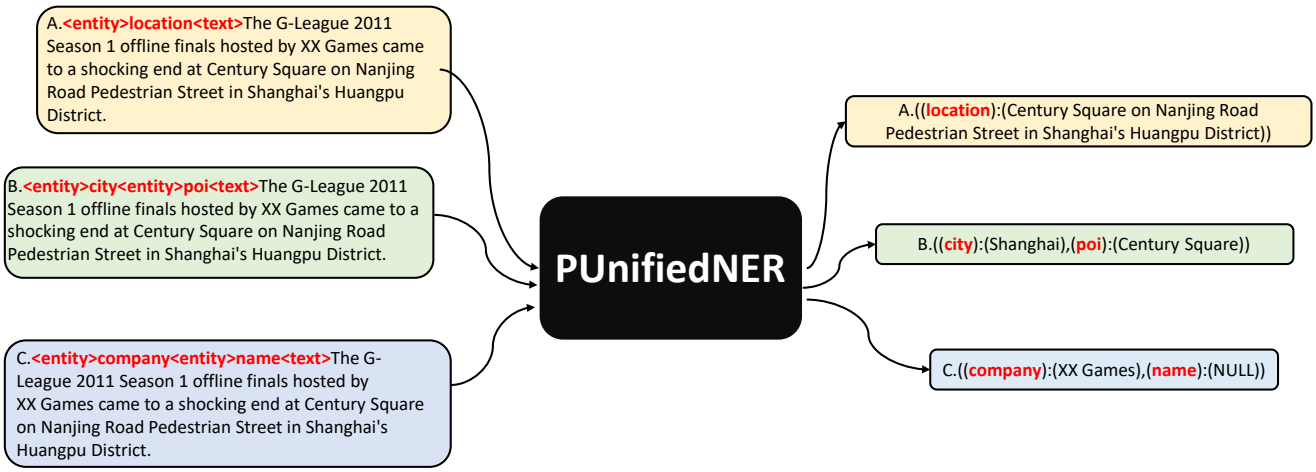


Figure 1: An overview of PUnifiedNER which reframes NER as a seq2seq task where red texts are prompts to suggest which entities users are interested in. For the same text input, PUnifiedNER returns different results conditioned on prefixed prompts.

prompt text on the left panels of Figure 1.

In addition to the ability to deal with various entity types and domains, our experimental results show that PUnifiedNER can achieve better performance than models of the same architecture trained using single datasets. This is a promising result, as different NER datasets vary greatly in corpus domain, entity annotation, and scale, and it is a difficult challenge to have a single versatile model capable of handling multiple scenarios simultaneously. In other words, joint training on multiple datasets mutually benefits each other rather than degrades each other, which has been viewed as a notorious challenge (Lu et al. 2020; Kamath et al. 2021; Li et al. 2022c). Furthermore, PUnifiedNER is on par with state-of-the-art methods on OntoNotes 4.0 dataset (Pradhan et al. 2013) and achieves a new state-of-the-art result on the Resume dataset (Zhang and Yang 2018). To conclude, our contributions are as follows:

- We propose PUnifiedNER a new method that supports the extraction and recognition of up to 37 entity types. We also show that given different prompts, the model is able to generate entity types of interest to the user, enabling on-demand entity recognition (Figure 1).
- We demonstrate that a single PUnifiedNER model works well on eight publicly available Chinese NER datasets. This represents a reduction from 2.72 billion parameters (parameter calculation based on the BERT-Large model) to 220 million parameters compared to using a set of dataset-specific models.
- Experimental results show that PUnifiedNER achieves significantly better performance than dataset-specific models with the same architecture. On average, PUnifiedNER results in an improvement of 1.33 points against those dataset-specific models on eight datasets, while handling multiple datasets simultaneously. Furthermore, PUnifiedNER creates a new state-of-the-art result on the Resume dataset (f-score 96.65 $\rightarrow$ 97.18) and achieves a result comparable to the state-of-the-art on the OntoNotes 4.0 dataset (f-score 83.08 vs. 82.56).
- Comprehensive ablation studies and pilot studies are conducted to verify the effectiveness of each component in PUnifiedNER, including Language Model Adaptation, Dataset-Dependent Prompting, etc.

Related Work

Prompting-based Approaches for NLP: The essence of prompt learning is making better use of pre-trained language model by adding additional "hints" (Liu et al. 2021a; Wang et al. 2022). Inspired by this, a large number of prompting methods are proposed in the literature to reformulate downstream tasks into pre-training ones to further leverage pre-trained language models. These prompting approaches can be categorised according to their model architectures and associated pre-training tasks, including methods based on bidirectional encoder-only models like BERT (Sun et al. 2021; Lu et al. 2022b), causal language decoder-only models like GPT-3 (Brown et al. 2020; Xie et al. 2021), and encoder-decoder models like T5 (Khashabi et al. 2020; Lester, Al-Rfou, and Constant 2021).

There are also some studies exploring using prompting to unify various tasks, which is inline with our usage of prompting. However, they focus on various classification (Lester, Al-Rfou, and Constant 2021) or QA tasks (Khashabi et al. 2020) instead of NER problems.

Deep Learning Approaches for NER: Most work considers NER as a sequence labelling task, where each token is assigned a pre-defined tag (e.g., BIO scheme). In this line of work, usually deep neural networks such as bidirectional LSTM (Zhang and Yang 2018; Liu et al. 2019) or pre-trained transformer-based language models (Ma et al. 2020; Liu et al. 2021b) is combined with Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira 2001) and has been

widely used in solving *flat NER*, where neither overlapped entities nor non-adjacent entities will appear in the text.

Inspired by the success of using deep learning model in flat NER task, many work in the literature attempts to solve *nested NER* (i.e., entities have overlaps with each other) or *discontinuous NER* (i.e., entities consist of non-consecutive text sequence) by reformulating the token-level sequence labelling methods used in flat NER to span-level sequence labelling methods (Katiyar and Cardie 2018; Yu, Bohnet, and Poesio 2020; Shen et al. 2021; Li et al. 2021), where text spans instead of tokens are enumerated and classified. Seq2Seq model is alsoapplicable for NER, which takes as input a sentence and generates a sequence of entity offsets, span lengths as well as labels (Fei et al. 2021).

There is an important research thread attempting to unify three sub-tasks in NER into one framework. Li et al. (2022a) propose W^{2}NER framework that reframes NER as a word-word relation classification problem and reach the state-of-the-art performance in 14 NER datasets. Yan et al. (2021) apply seq2seq model to generate a sequence of entity start-end indexes and types. More recently, Lu et al. (2022c) propose a text-to-structure framework based on seq2seq model to unify not only NER tasks but other information extraction tasks (i.e., event, relation and sentiment extraction).

Differences between PUnifiedNER and Other Unified Approaches: Though unifying various sub-tasks, most of existing work essentially tries to build up a dedicated model for every single dataset. The most significant difference is that, we focus on training a versatile model that handles a large scale of entity types as well as datasets simultaneously, which is more compelling.

Other differences are (1) PUnifiedNER vs. W^{2}NER: we reframe NER as seq2seq while W^{2}NER recasts NER as word-word relation classification, which is totally different, and PUnifiedNER outperforms W^{2}NER in dataset Resume. (2) PUnifiedNER vs. other seq2seq-based NER approaches: most seq2seq methods still consider identifying more accurate entity boundaries, thus, the output is entity offsets and span lengths, or entity start-end indexes. PUnifiedNER elegantly generates entity types and corresponding text spans using natural language, which also indicates the potential of tackling nested and discontinuous NER. We leave these explorations for future work. Besides, enhanced by prompt-learning, our model implements the on-demand named entity recognition which has not been fully explored in existing studies.

PUnifiedNER

Our unified NER system is based on the recent seq2seq frameworks, i.e., T5 (Raffel et al. 2020).[1] We first present the model architecture and describe how we reframe the NER task on different datasets into a unified seq2seq format with prompts. We then present the datasets used in our experiments. Finally, we detail the way PUnifiedNER is trained.

[1]We use the T5-v1.1-base-chinese checkpoint pre-trained by UER: https://huggingface.co/uer/t5-base-chinese-cluecorpussmall.

Model Architecture

Our model is mainly implemented by reusing the pre-trained T5 model. T5 follows the vanilla transformer encoder-decoder architecture (Vaswani et al. 2017) with some minor modifications, described in Raffel et al. (2020). The encoder, consisting of multiple transformer encoder blocks, uses the bidirectional multi-head self-attention mechanism to encode the input text sequence. The decoder, consisting of multiple transformer decoder blocks, receives the output hidden state from the encoder and uses the unidirectional attention mechanism to autoregressively generate the output text sequence. The predicted token generated at each step is compared to the ground truth token, supervised using a cross entropy loss.

Task Reframing

As shown in Figure 1, we reframe NER as a prompting-based seq2seq problem. Formally, given an original text sequence x, we transform x to a source sequence x_{input} by prefixing it with a series of prompts as follows:

$$x_{input} = [s_e, s_{p_1}, ..., s_e, s_{p_n}, s_t, x] \quad (1)$$

where x_{input} is the model input; s_e is the special token *"<entity>"* indicating that the following token is the entity type that we are interested in; s_{p_i} is the entity type, e.g., "city"; and s_t is the special token *"<text>"* indicating that the following text sequence is the sentence from which entities should be extracted. Then the target sequence y_{output} is as follows:

$$y_{output} = ((s_{p_1}) : (ent_1), ..., (s_{p_n}) : (ent_n)) \quad (2)$$

where y_{output} is the model output; s_{p_i} are entity types that are identical to those in Equation 1; and ent_i is the ground truth texts extracted from the input sequence.

For example, suppose we have an input x *"Tom will go to the zoo tomorrow."* and we are interested in entities "time" and "location". Then x_{input} will be *"<entity><time><entity><location><text>Tom will go to the zoo tomorrow."* and y_{output} will be *"((time):(tomorrow),(location):(zoo))"*. Alternatively, if we intend to parse entities "name", then x_{input} will be *"<entity><name><text>Tom will go to the zoo tomorrow."* and y_{output} will be *"((name):(Tom))"*. In this case, we hope prompts can function well as indicators that suggest which entities users are interested in to steer model produce different entities, even given the same input x (also see example A and B in Figure 1). Note that if users assign an entity absent in input x, the ent_i in y_{output} will be a special token *"NULL"* (see example C in Figure 1).

We argue that this design has several advantages: (1) prompts serve as indicators of target entities as discussed above; (2) unifying entities from different datasets into a single model reduces the cost of model deployment compared to using a dedicated model for each dataset; and (3) reusing labelled information of different datasets. Many NER datasets share similar entity types such as "name", "location" and "organisation". Training a model solely on a single dataset can only learn knowledge of the given dataset.

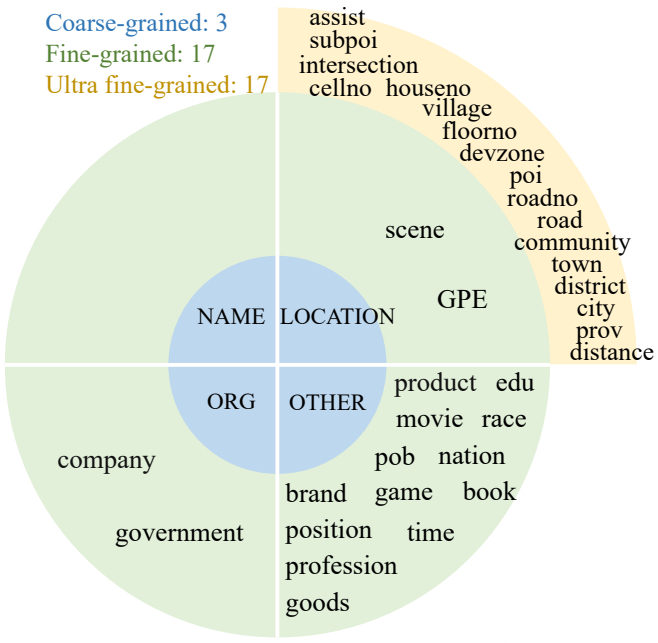


Figure 2: Properties of 37 entities provided by eight public NER datasets. All entities are manually categorised into four groups, i.e., name, location, organisation, and others. Entities in the same quadrant fall into the same group. Entities in different colors indicate different entity granularity, i.e., blue (coarse-grained), green (fine-grained) and yellow (ultra fine-grained). Note that the manually crafted granularity/category information in this figure is for a better visualisation and is not leveraged in model training.

PUnifiedNER can exploit the shared knowledge crossing diverse datasets by using prefixed prompting.

Datasets

We train and evaluate PUnifiedNER on eight existing public NER datasets that target various entity types from different domains including social media, e-commerce, news, postal address, etc. We use the *Ecommerce* (Ding et al. 2019), *MSRA* (Levow 2006), *OntoNotes 4.0* (Pradhan et al. 2013), *People Daily 2014*, *Boson*,[2] *Resume* (Zhang and Yang 2018), *CCKS2021*,[3] and *CLUENER* (Xu et al. 2020) datasets. After preprocessing, these datasets provide 37 unique entities. Figure 2 summarises the properties of these entities. They are categorised into four big groups—name, location, organisation, and other—based on their semantic information and three granularities—coarse-grained, fine-grained, and ultra fine-grained. Note that all texts are originally in Chinese, so we provide translations in this paper to facilitate understanding. Additional details of the datasets and entities are provided in Appendix.

[2]People Daily 2014 and Boson datasets are available at https://github.com/hspuppy/hugbert/tree/master/ner_dataset.

[3]https://tianchi.aliyun.com/competition/entrance/531900/information

Language Model Adaptation

T5 is pre-trained with "reconstructing" masked spans in the source sequence, which are marked with unique sentinel tokens. The target output sequence consists of several sentinels that are followed by the corresponding masked content. Concretely, given a text *"The capital of China is Beijing."*, the source sequence of the pre-trained example might be constructed as *"The <extra0> of <extra1> is Beijing."* and the target sequence will be *"<extra0>capital<extra1>China<extra2>"* where *"<extra$_i$>"* are sentinels and the last sentinel (*"<extra2>"* in this case) is the end of sequence token.

Although this pre-training objective works effectively in Raffel et al. (2020), we believe that this setup is not good enough to shift the pre-trained model to our downstream prompting-based seq2seq NER task. One main obstacle is the objective gap where the pre-trained T5 has never seen a natural and complete input sequence (i.e., text sequence without sentinels) that is crucial for NER tasks. The NER model needs to extract text spans correctly from the original input given the completed context. Therefore, inspired by the idea of continuously pre-training language models (Khashabi et al. 2020; Lu et al. 2020; Lester, Al-Rfou, and Constant 2021; Lu et al. 2022c), we adopt the prefix language modelling objective discussed by Raffel et al. (2020); Lester, Al-Rfou, and Constant (2021) to further adapt the pre-trained language model: we randomly split a given natural text into two substrings and the model must produce the latter substring conditioned on the former substring. For example, given a original sentence *"The capital of China is Beijing."*, the source input of a pre-trained example might be constructed as *"The capital of China"* and the target output will be *"is Beijing."*, respectively.

We conduct prefix language modelling training based on T5-v1.1-base-chinese over all NER datasets, encouraging the model to close both the objective gap and the domain gap simultaneously. We have found that training using this self-supervised objective can largely close the objective and domain gaps as well as boosting model performance. This is presented in the pilot study.

Multi-Dataset Joint Learning

After language model adaptation we adopt a simple, yet effective, multi-dataset learning strategy to train the model: for each batch, we randomly select examples from different datasets until the number of examples is identical to the batch size.

Note that during multi-dataset training, it is unfeasible to use all entities as prefixed prompts—prefixing all entity prompts results in around 296 extra tokens (avg. 8 for each entity in Chinese) to the input sequence as well as multiple *"NULL"* tokens in the target output sequence. Therefore, we propose three prefixed prompt setups:

- **Random Prompt:** During training, we randomly sample up to all 37 entities as prefixed prompts. While during inference, we use all entities of the specific testing dataset as prefixed prompts. For

example, the MSRA dataset contains three entities—location, organisation, and name—so during inference the prefixed prompts are always *"<entity>-<location><entity><organisation><entity><name><text>"*.

- **Random Prompt + Exact Match:** During training, each original example is prefixed in two styles (therefore two training examples are generated), one using Random Prompt as discussed above, and the other using the exact entities from the ground truth as prefixed prompts (i.e., Exact Match). During inference, we use all entities of the specific testing dataset as prefixed prompts. Though this setting suffers from information leakage where ground truth entities can help the model narrow down the entity space to reduce difficulty during training, we do not use Exact Match during inference. Thus it is still a fair comparison. Also, we hope the use of Random Prompt can partially alleviate the information leakage problem.
- **Dataset-Dependent Prompt:** Training and inference both use all entities from the specific dataset as prefixed prompts.

The effectiveness of these three prefixed prompt settings is explored in the ablation study.

Pilot Study: The Benefits of Language Model Adaptation

We first ask a question: *Is it necessary to apply language model adaptation before multi-dataset learning?* To answer this question, we first train a T5-v1.1-base-chinese checkpoint on the CLUENER dataset using the Dataset-Dependent Prompt approach discussed in previous section. The best validation f-score achieved is 48.67 which is far from ideal.

Then we continuously pre-train T5 model with the prefix language modelling objective over eight public datasets. The training/validation split is the same as the original setting. We also use the sampling strategy discussed in the previous section to construct training batches and continuously pre-train T5 up to 50K steps, evaluating every 1K steps (other training details are provided in the Appendix). Figure 3 shows the prefix language modelling validation loss for the eight datasets. We find that for most datasets, the model achieves the lowest validation loss with at least 5K steps.

Finally, we select the model after 6K steps, which has on average a lower validation loss across all eight datasets, and retrain it with the CLUENER dataset. The best validation f-score is 74.78, outperforming the previous f-score of 48.67 by a large margin. This demonstrates the necessity and effectiveness of applying language model adaptation to close the gap between T5 pre-training and downstream prompting-based seq2seq NER tasks. Given the benefit brought by language model adaptation, we will use the model at prefix language modelling step 6K as our backbone model unless otherwise stated.

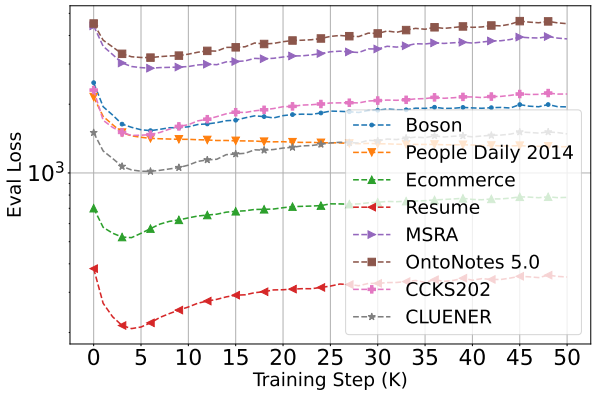
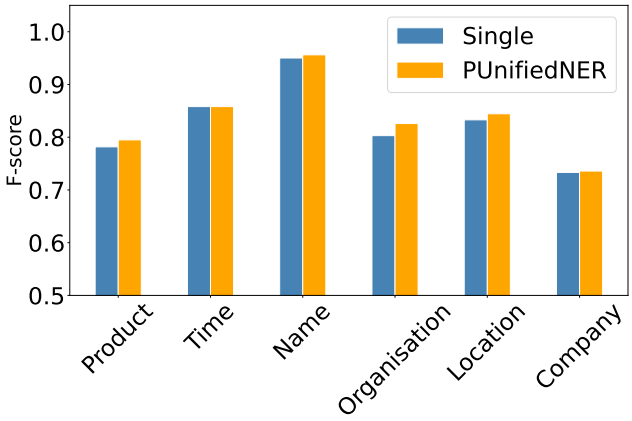
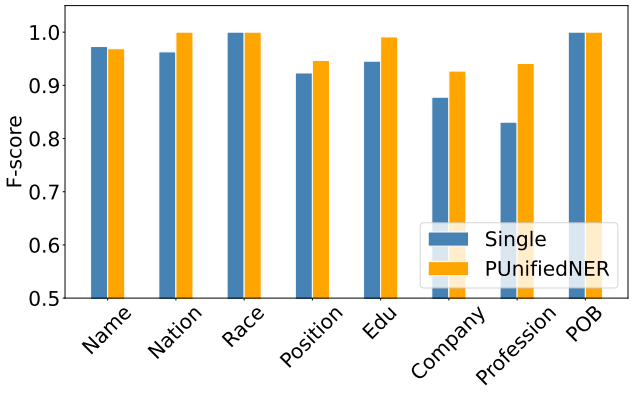


Figure 3: Prefix language modelling validation loss on eight datasets. Loss is log scaled to make trends more noticeable.



(a)



(b)

Figure 4: Fine-grained named entity f-score of (a) Boson (b) Resume where Edu and POB denote education and place of birth, respectively.

Experiments

Comparison to Single Dataset Performance

Our intention is to answer the question: *Can a jointly*

Methods	MSRA*	OntoNotes	Resume	CLUENER	Ecom	PD	Boson	CCKS*	Avg.	# models
Single-Dataset	**87.78**	78.44	93.91	74.57	69.38	**96.99**	82.60	84.26	83.49	8
PUnifiedNER	87.07	**82.56**	**97.18**	**77.00**	**71.32**	94.49	**83.58**	**85.34**	**84.82**	1

Table 1: Comparison of PUnifiedNER to single-dataset training. "*" indicates that this dataset does not contain the test set, thus we report the results of validation set.

Flat NER Methods	MSRA	OntoNotes	Resume	CLUENER	Ecommerce	# Params
Lattice LSTM (Zhang and Yang 2018)	93.18	73.88	94.46	-	-	-
TENER (Yan et al. 2019)	92.74	72.43	95.00	-	-	-
LGN (Gui et al. 2019)	93.71	74.45	95.11	-	-	-
FLAT (Li et al. 2020)	96.09	81.82	95.86	-	-	N*110M
SoftLexicon (Ma et al. 2020)	95.42	82.81	96.11	-	-	N*110M
LEBERT (Liu et al. 2021b)	95.70	82.08	96.08	-	-	N*110M
State-of-the-art Methods						
W^2NER (Li et al. 2022b)	**96.40**	**83.08**	96.65	-	-	N*110M
NEZHA-BC (Zhang et al. 2022)	-	-	-	-	**78.69**	N*110M
DML-NEZHA-Large	-	-	-	**83.30**	-	N*340M
Our Method						
PUnifiedNER	87.07	82.56	**97.18**	77.00	71.32	1*220M

Table 2: Comparison of the performance of PUnifiedNER to recent state-of-the-art methods. All state-of-the-art methods are dataset-specific models. Estimated number of parameters used for model deployment in realistic scenarios are shown in the rightmost column, where N is the number of datasets. The number of parameters for some methods are estimated from the corresponding paper.

Prompt	MSRA	OntoNotes	Resume	CLUENER	CCKS2021	Avg. Score
Random Prompt	69.93	46.88	95.48	69.62	84.30	73.24
Random Prompt + Exact Match	69.21	55.47	90.20	66.30	80.37	72.31
Dataset-Dependent Prompt	**87.07**	**82.56**	**97.18**	**77.00**	**85.34**	**85.83**

Table 3: Ablations on our Prefixed Prompts design choices.

Step	MSRA	OntoNotes	Resume	CLUENER	CCKS	Ecom	People Daily	Boson	Avg. Score
Step 6K	**87.07**	**82.56**	97.18	**77.00**	85.34	**71.32**	94.49	**83.58**	**84.82**
Step 10K	86.94	82.09	96.62	76.16	**86.01**	71.23	95.27	79.69	84.25
Step 50K	86.64	81.86	**97.24**	75.21	84.83	70.52	**96.95**	82.84	84.51

Table 4: Ablations on language model adaptation steps.

trained PUnifiedNER model learn shared information among datasets and hence outperform seq2seq NER models trained on single datasets?

As with the pilot study, we first establish baseline performance of single-dataset training by fine-tuning the T5 backbone on each NER dataset independently. The fine-tuning procedure is the same as discussed in the pilot study. Then we jointly train another T5 backbone on multiple NER datasets using Dataset-Dependent Prompt settings and the same sampling strategy to construct batches described in the pilot study. For single-dataset training, we select the best performing model on a validation set and report its test f-score. For multi-dataset training, we select the model with best mean f-score on all eight NER datasets and report its test f-scores on each dataset separately. Other hyper-parameters settings are detailed in the Appendix.

Experimental results are shown in Table 1. It shows a clear pattern that the jointly trained PUnifiedNER model can learn shared information between datasets, surpassing single-dataset trained models by an average improvement of 1.33 points. To be specific, in six out of eight datasets, PUnifiedNER exceeds its single-dataset trained counterparts by a large margin especially in OntoNotes 4.0 (82.56 vs. 78.44), Resume (97.18 vs. 93.91) and CLUENER (77.00 vs. 74.57), and with a reasonable improvement in Ecommerce (71.32 vs. 69.38), Boson (83.58 vs 82.60) and CCKS2021 (85.34 vs. 84.26). In MSRA, the performance of PUnifiedNER is also comparable. The only exception is the People Daily 2014 dataset where the single-dataset trained model exceeds PUnifiedNER by 2.5 points. Given the fact that the People Daily 2014 dataset includes more than 400K instances, we believe that PUnifiedNER suffers from insufficient training regarding the People Daily 2014 dataset. This is probably because we select the checkpoint that is generally useful for all datasets according to performance on validation sets, where models are converged in all datasets other than People Daily 2014. In our follow-up observations, we found that the validation set performance of the PUnifiedNER model was

still improving after the "best" checkpoint, and more details can be seen in the Appendix. We suspect that this can be alleviated by some multi-training strategies such as Curriculum Learning and Dynamic Stop-and-Go (Lu et al. 2020), which we will address in future work.

Fine-grained Analysis: We further report fine-grained named entity f-scores for the Boson and Resume datasets in Figure 4. First, for the Boson dataset the f-score achieved by PUnifiedNER for almost all entity types is improved compared to the corresponding single-training counterpart. This is consistent to our expectation because all entity types in Boson have appeared in other datasets and joint training can take advantage of "seeing" more data. It is surprising to find out that the prompting-based joint training can also bring additional discriminative ability for entity types that only appear in one dataset, which is evidenced by the f-scores for the Education and Profession entities in the Resume dataset as shown in Figure 4(b).

Comparison to State-of-the-art Approaches

In Table 2 we compare the performance of PUnifiedNER with recent state-of-the-art approaches, as well as other flat NER approaches including **Lattice LSTM** (Zhang and Yang 2018), **TENER** (Yan et al. 2019), **LGN** (Gui et al. 2019), **FLAT** (Li et al. 2020), **SoftLexicon** (Ma et al. 2020), and **LEBERT** (Liu et al. 2021b). We draw special comparison with the recent **W^{2}NER** approach (Li et al. 2022b) as it demonstrates very strong performance on several Chinese NER datasets. We also compare PUnifiedNER with **DML-NEZHA-Large**[4] and **NEZHA-BC** (Zhang et al. 2022) since they achieve state-of-the-art results for the CLUENER and Ecommerce datasets, respectively. Our single multi-dataset jointly trained model achieves competitive performance with these state-of-the-art dataset-specific models in two out of five datasets. In particular, on the Resume dataset, PUnifiedNER surpasses the state-of-the-art (96.65 -> 97.18).

Though the performance of PUnifiedNER is less satisfactory for some datasets e.g., MSRA, we should keep in mind that PUnifiedNER is a single model for diverse datasets while the other methods train dedicated models for each dataset, which means more resources are required when deploying models in realistic situations. To be specific, W^{2}NER and DML-NEZHA-Large are based on BERT-Base (110 million parameters) or BERT-Large (340 million parameters) and include additional layers (e.g., Convolution Layer and Co-Predictor Layer in Li et al. (2022b)). Specifically, if we need to deploy an NER model for N datasets, the number of model parameters is at least $N * 340$ million. However, if we use PUnifiedNER, the number of model parameters is equal to the number of T5-base parameters, i.e., 220 million, which is $N * 1.55$ times smaller than in W^{2}NER (BERT-Large version) or DML-NEZHA-Large. Also, as reported in previous work (Raffel et al. 2020; Lester, Al-Rfou, and Constant 2021; Lu et al. 2022c), increasing the model size of T5 architecture can further improve performance. Thus, we believe that PUnifiedNER is able to compensate the performance if we change the backbone to T5-Large/XL/XXL without significant increase in the number of parameters needed.

Ablation Study

Ablations on Prefixed Prompts Setups: To verify our prefixed prompt design choices, we perform ablations for different prefixed prompt settings as discussed in the pilot study and report results on five datasets: MSRA, OntoNotes 4.0, Resume, CLUENER and CCKS2021.

The results are shown in Table 3. We also report overall average performance in the rightmost column. Our default setting is Dataset-Dependent Prompt. We compare this with two ablations: Random Prompt and Random Prompt+Exact Match. We observe that Dataset-Dependent Prompt leads to much better performance compared to Random Prompt (avg. 85.83 vs. 73.24) and Random Prompt+Exact Match (avg. 85.83 vs. 72.31). This makes sense since Exact Match suffers from information leakage in training, thus the performance degrades during inference. While Random Prompt results in too many *"NULL"* examples with few or even zero positive examples where the model learns nothing about ground truth. Dataset-Dependent Prompt is more useful in balancing positive and *"NULL"* examples, reducing the risk of information leakage and improving data efficiency.

Ablations on Language Model Adaptation Steps: The pilot study has shown that language model adaptation can largely close the objective gap and boost downstream task performance on the prompting-based seq2seq NER task. In this subsection, we go a step further to verify whether selecting the model with lowest validation loss on language model adaptation can benefit the downstream multi-dataset joint learning. We perform another multi-dataset joint training with all eight NER datasets based on model after 10k steps and after 50K steps (full training). All training settings are the same as described in the pilot study. Other hyper-parameters settings are presented in the Appendix.

Table 4 shows the results where our default setting of 6K steps surpasses the other two settings: after 10K steps (avg. 84.82 vs. 84.25) and after 50K steps (avg. 84.82 vs. 84.51). This demonstrates that selecting the model with the lowest validation loss is a useful approach. However, other settings outperform the default for some datasets. For example, the model after 50K steps achieves an f-score of 97.24, which is a new state-of-the-art for the Resume dataset. We notice from Figure 3 that the model after 50K steps has a high validation loss as compared to the model after 6K steps for the Resume dataset , which suggests that validation loss might not be the only model selection criterion that should be used. We leave this exploration for future work.

On-demand Named Entity Recognition

Our code and a demo interface for PUnifiedNER have been made available,[5] which demonstrates the capability of on-demand named entity recognition.

[4]State-of-the-art performance is reported at https://www.cluebenchmarks.com/ner.html as of July 1st, 2022.

[5]All resources are available at: https://github.com/GeorgeLuImmortal/PUnifiedNER.

Conclusion

In this work, we present a novel Prompting-based Unified NER system (PUnifiedNER) that can recognise a large set of entity types in data from various domains and support on-demand entity recognition. To achieve this, we first recast the NER task to a seq2seq task where a prefix language modelling objective is introduced to reduce the gap between pre-training and fine-tuning. A pilot study shows that prefix language modelling is very effective in adapting pre-trained language models. Dataset-Dependent Prompt is designed to unify data from all datasets into a united format that enables joint training crossing multiple datasets for PUnifiedNER. Besides on-demand named entity recognition, experimental results show that the multi-dataset training empowered by prompting can also lead to significant performance gains over single-dataset training, while dramatically reducing model deployment cost. Further, the jointly trained PUnifiedNER model sets a new state-of-the-art performance level for the Resume dataset, and is comparable to other state-of-the-art dataset-specific NER methods in some cases. Lastly, extensive ablation studies are performed to clarify the design choices of PUnifiedNER.

Acknowledgements

We would like to thank **Hongyu Lin** from Chinese Academy of Sciences for his thoughtful discussion, as well as the many others who have helped. We would also like to thank anonymous reviewers for their insightful comments to help improve the paper. This publication has emanated from research conducted with the support of SenseTime Research.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; 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.; 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. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota.

Ding, R.; Xie, P.; Zhang, X.; Lu, W.; Li, L.; and Si, L. 2019. A neural multi-digraph model for Chinese NER with gazetteers. In *Proceedings of the ACL 2019*, 1462–1467.

Fei, H.; Ji, D.; Li, B.; Liu, Y.; Ren, Y.; and Li, F. 2021. Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 12785–12793.

Fries, J. A.; Weber, L.; Seelam, N.; Altay, G.; Datta, D.; Garda, S.; Kang, M.; Su, R.; Kusa, W.; Cahyawijaya, S.; et al. 2022. Bigbio: A framework for data-centric biomedical natural language processing. *arXiv preprint arXiv:2206.15076*.

Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; and Huang, X. 2019. A Lexicon-Based Graph Neural Network for Chinese NER. 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)*, 1040–1050. Hong Kong, China.

Kamath, A.; Singh, M.; LeCun, Y.; Synnaeve, G.; Misra, I.; and Carion, N. 2021. MDETR-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1780–1790.

Katiyar, A.; and Cardie, C. 2018. Nested Named Entity Recognition Revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 861–871. New Orleans, Louisiana.

Khashabi, D.; Min, S.; Khot, T.; Sabharwal, A.; Tafjord, O.; Clark, P.; and Hajishirzi, H. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1896–1907. Online.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 282–289. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic.

Levow, G.-A. 2006. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 108–117. Sydney, Australia.

Li, F.; Lin, Z.; Zhang, M.; and Ji, D. 2021. A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition. In *Proceedings of the ACL 2021 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4814–4828. Online.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2022a. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10965–10973.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2022b. Unified named entity recognition as word-word relation classification. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Li, L. H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. 2022c. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10965–10975.

Li, X.; Yan, H.; Qiu, X.; and Huang, X. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the ACL 2020*, 6836–6842. Online.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, W.; Fu, X.; Zhang, Y.; and Xiao, W. 2021b. Lexicon Enhanced Chinese Sequence Labeling Using BERT Adapter. In *Proceedings of the ACL 2021 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5847–5858. Online.

Liu, W.; Xu, T.; Xu, Q.; Song, J.; and Zu, Y. 2019. An Encoding Strategy Based Word-Character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2379–2389. Minneapolis, Minnesota.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Lu, J.; Goswami, V.; Rohrbach, M.; Parikh, D.; and Lee, S. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10437–10446.

Lu, J.; Yang, L.; Namee, B.; and Zhang, Y. 2022a. A Rationale-Centric Framework for Human-in-the-loop Machine Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6986–6996. Dublin, Ireland: Association for Computational Linguistics.

Lu, J.; Zhao, R.; Mac Namee, B.; Zhu, D.; Han, W.; and Tan, F. 2022b. What Makes Pre-trained Language Models Better Zero/Few-shot Learners? *arXiv preprint arXiv:2209.15206*.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022c. Unified Structure Generation for Universal Information Extraction. In *Proceedings of the ACL 2022 (Volume 1: Long Papers)*, 5755–5772. Dublin, Ireland.

Ma, R.; Peng, M.; Zhang, Q.; Wei, Z.; and Huang, X. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the ACL 2020*, 5951–5960. Online.

Mao, Z.; Zhu, D.; Lu, J.; Zhao, R.; and Tan, F. 2022. SDA: Simple Discrete Augmentation for Contrastive Sentence Representation Learning. *arXiv preprint arXiv:2210.03963*.

Parmar, M.; Mishra, S.; Purohit, M.; Luo, M.; Mohammad, M.; and Baral, C. 2022. In-BoXBART: Get Instructions into Biomedical Multi-Task Learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 112–128. Seattle, United States.

Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards Robust Linguistic Analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 143–152. Sofia, Bulgaria.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Shen, Y.; Ma, X.; Tan, Z.; Zhang, S.; Wang, W.; and Lu, W. 2021. Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. In *Proceedings of the ACL 2021 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2782–2794. Online.

Sun, Y.; Zheng, Y.; Hao, C.; and Qiu, H. 2021. NSP-BERT: A Prompt-based Zero-Shot Learner Through an Original Pre-training Task–Next Sentence Prediction. *arXiv preprint arXiv:2109.03564*.

Tan, F.; Hu, Y.; Hu, C.; Li, K.; and Yen, K. 2020. TNT: Text Normalization based Pre-training of Transformers for Content Moderation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4735–4741. Online.

Tan, F.; Hu, Y.; Yen, K.; and Hu, C. 2021. BERT-Beta: A Proactive Probabilistic Approach to Text Moderation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8667–8675. Online and Punta Cana, Dominican Republic.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Arunkumar, A.; Ashok, A.; Dhanasekaran, A. S.; Naik, A.; Stap, D.; et al. 2022. Super-NaturalInstructions:Generalization via Declarative Instructions on 1600+ Tasks. In *EMNLP*.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Xu, L.; Dong, Q.; Liao, Y.; Yu, C.; Tian, Y.; Liu, W.; Li, L.; Liu, C.; Zhang, X.; et al. 2020. CLUENER2020: fine-grained named entity recognition dataset and benchmark for chinese. *arXiv preprint arXiv:2001.04351*.

Yan, H.; Deng, B.; Li, X.; and Qiu, X. 2019. TENER: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.

Yan, H.; Gui, T.; Dai, J.; Guo, Q.; Zhang, Z.; and Qiu, X. 2021. A Unified Generative Framework for Various NER Subtasks. In *Proceedings of the ACL 2021 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5808–5822. Online.

Yu, J.; Bohnet, B.; and Poesio, M. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the ACL 2020*, 6470–6476. Online.

Zhang, X.; Jiang, Y.; Wang, X.; Hu, X.; Sun, Y.; Xie, P.; and Zhang, M. 2022. Domain-Specific NER via Retrieving Correlated Samples. In *Proceedings of the 29th International Conference on Computational Linguistics*, 2398–2404. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the ACL 2018 (Volume 1: Long Papers)*, 1554–1564. Melbourne, Australia.

Dataset	Sentence				Mention			
	#All	#Train	#Dev	#Test	#All	#Train	#Dev	#Test
Ecommerce	4,987	3,989	500	498	15,216	12,109	1,540	1,567
MSRA	50,729	44,364	4,365	-	80,884	74,703	6,181	-
OntoNotes 4.0	24,371	15,724	4,301	4,346	28,006	13,372	6,950	7,684
People Daily 2014	286,269	-	-	-	751,229	-	-	-
Boson	16,753	-	-	-	23,172	-	-	-
Resume	4,759	3,819	463	477	16,565	13,438	1,497	1,630
CLUENER	12,091	10,748	1,343	1,350	27,043	23,971	3,072	-
CCKS2021	10,825	8,855	1,970	-	53,620	43,644	9,976	-

Table 5: Dataset Statistics. "#" denotes the amount. For MSRA and CCKS2021, we report the result on validation set. For People Daily 2014 and Boson, we sample 2,000 examples and 2,000 examples from all examples for evaluation and testing, respectively. For CLUENER, we submit prediction of test set to the competition and report the score provided by the website.

Dataset	#Entity	Entity
Ecommerce	2	{'HP(brand)':'品牌','HC(commodity)':'商品'}
MSRA	3	{'LOC':'地点','PER':'名称','ORG':'组织'}
OntoNotes 4.0	4	{'GPE':'地缘政治实体','LOC':'地点','PER':'名称','ORG':'组织'}
People Daily 2014	4	{'LOC':'地点','PER':'名称','ORG':'组织','T(time)':'时间'}
Boson	6	{'product_name':'产品','time':'时间', 'person_name':'名称', 'org_name':'组织', 'location':'地点', 'company_name':'公司'}
Resume	8	{'NAME':'名称', 'CONT':'国籍', 'RACE':'民族', 'TITLE':'职位', 'EDU':'学历', 'ORG':'公司', 'PRO':'专业', 'LOC(place of birth)':'籍贯'}
CLUENER	10	{'name':'名称','company':'公司','game':'游戏','organization':'组织','movie':'电影', 'address':'地点','position':'职位', 'government':'政府','scene':'景点','book':'书籍'}
CCKS2021	17	{'prov':'省份', 'city':'城市', 'district':'区', 'town':'街道', 'community':'社区', 'poi(point of interest)':'兴趣点', 'road':'路', 'roadno':'路号', 'subpoi':'次兴趣点', 'devzone':'产业园', 'houseno':'楼号', 'intersection':'路口', 'assist':'方位', 'cellno':'单元', 'floorno':'楼层', 'distance':'距离', 'village_group':'村组'}

Table 6: Entity tag of each dataset and the conversion from tag used in dataset to corresponding Chinese natural language. For some tags that are hard to understand, we provide their meaning in brackets. "#" denotes the amount of entity types.

Appendices

A Evaluation Metrics

Regarding evaluation metrics, we follow prior work (Yan et al. 2021; Li et al. 2022b; Lu et al. 2022c) and employ macro F1-score. A predicted entity is counted as true positive only if its text span and entity types match those of a gold entity.

B Dataset Statistics

We evaluate our framework on 8 Chinese flat NER datasets. In Table 5, we present the detailed statistics. The details of entity type of each dataset is presented in Table 6.

C Implementation Details

In this section, we provide more details of our experiments. Hyper-parameter settings are listed in Table 7. We adopt AdamW (Loshchilov and Hutter 2018) optimizer. Our model is implemented with PyTorch, language model adaptation is trained with 24 NVIDIA 1080Ti GPUs, multi-dataset training is trained with 8 Tesla V100 GPUs. Note that we use beam width equals to 5 in evaluation but 10 in testing. Single-dataset training is trained with one Tesla V100 GPU and we use beam width equals to 10 for both evaluation and testing. Hyper-parameter tuning is based on validation set.

E Multi-dataset Joint Training Performance on Validation Set

The validation macro f-score of each checkpoint is presented in Figure 5. We select checkpoint on step 30K since it is generally ideal for all datasets. However, we can observe that after the chosen checkpoint, the validation performance on People Daily 2014 (the red line) consistently increases and reaches the best f-score at step 100K. This explains the suboptimal performance of PUnifiedNER on dataset People Daily 2014.

Language Model Adaptation		Multi-dataset Training		Single-dataset Training	
Hyper-parameter	**Value**	**Hyper-parameter**	**Value**	**Hyper-parameter**	**Value**
Warm-up step	1K	Warm-up step	5K	Warm-up step	1K
Total step	50K	Total step	100K	Total epoch	40
Eval step	10K	Eval step	20K	Eval epoch	1
Max input length	256	Max input length	512	Max input length	512
Max output length	256	Max output length	512	Max output length	512
Learning rate	1e-4	Learning rate	1e-4	Learning rate	1e-5
Batch size	48	Batch size	32	Batch size	4
Beam width	-	Beam width	5	Beam width	10

Table 7: Hyper-parameter settings.

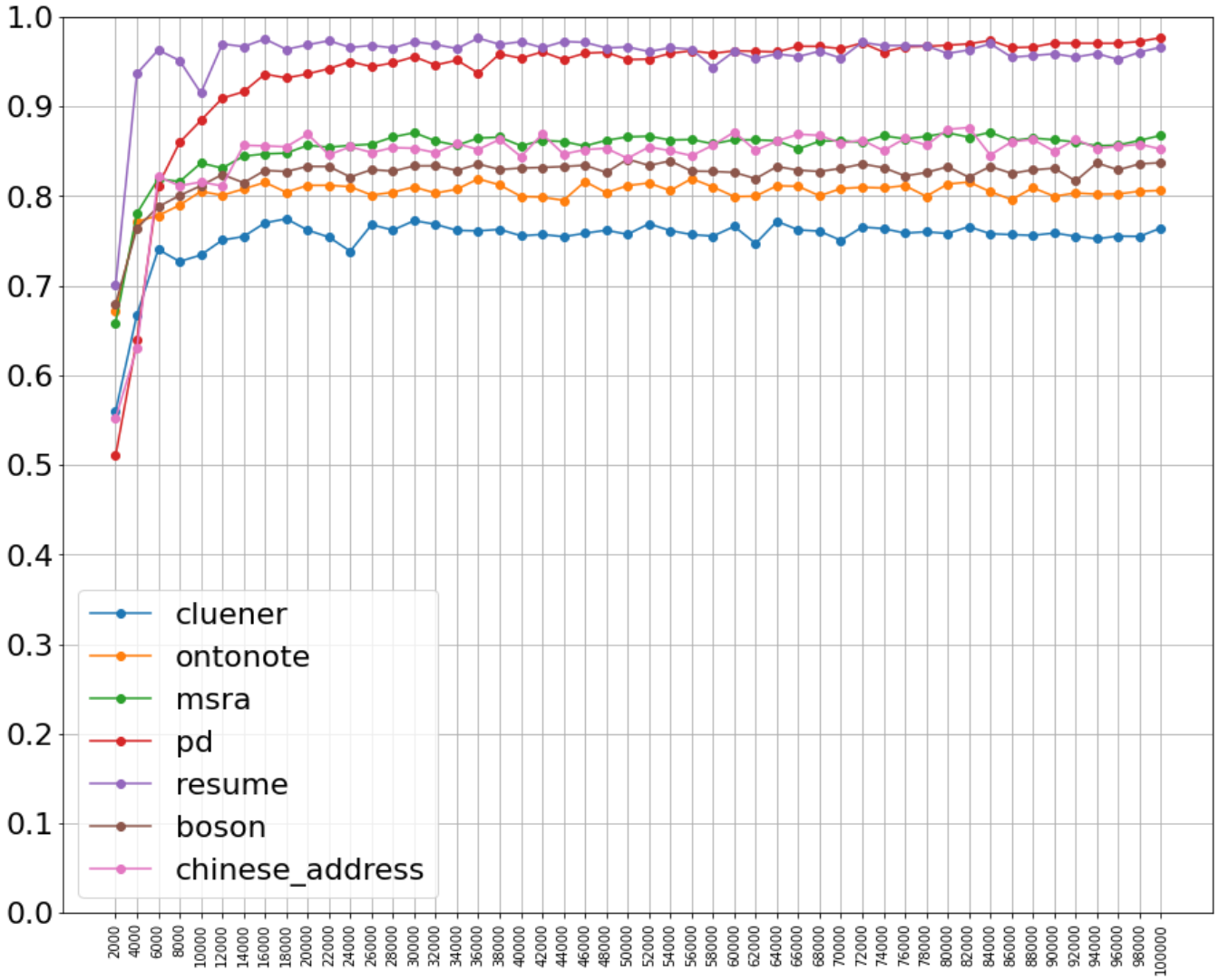


Figure 5: Validation macro f-score of multi-dataset joint training.