DAML: Chinese Named Entity Recognition with a fusion method of data-augmentation and meta-learning

Anonymous ACL submission

Abstract

Overfitting is still a common problem in NER with insufficient data. Latest methods such as Transfer Learning, which focuses on storing knowledge gained while solving one task and applying it to a different but related task, or Model-Agnostic Meta-Learning (MAML), which learns a model parameter initialization that generalizes better to similar tasks. However, these methods still need rich resources to pre-train. In this work, we present new perspectives on how to make the most of in-domain and out-domain information. By introducing a fusion method of data augmentation and MAML, we first use data augmentation to mine more information. With the augmented resources, we directly utilize out-domain and in-domain data with MAML, while avoiding performance degradation after domain transfer. To further improve the model's generalization ability, we proposed a new data augmentation method based on a generative approach. We conduct experiments on six open Chinese NER datasets (MSRANER, PeopleDairyNER, CLUENER, WeiboNER, Resume NER, and BOSONNER). The results show that our method significantly reduces the impact of insufficient data and outperforms the state-of-the-art.

1 Introduction

NER is one of the common problems in Natural Language Processing(NLP), which aims at dividing the elements in text into predefined categories, such as person names, place names, organizations, or any other classes of interest. Despite being conceptually simple, NER is not an easy task. In recent years, papers applying deep neural networks (DNNs) to the task of NER have successively advanced the state-of-the-art (SOTA) (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Peters et al., 2017, 2018). However, the more parameters you want the model to learn or as complex as the problem at hand so does the data required for training increase. Otherwise, the problem of having more dimensions yet small data results in over-fitting. For instance, on the OntoNotes-5.0 English dataset, whose training set contains 1,088,503 words, a DNN model outperforms the best shallow model by 2.24% as measured by F1 score (Chiu and Nichols, 2016). On the other hand, for comparatively small CoNLL-2003 English dataset, whose training set contains 203,621 words, the best DNN model enjoys only a 0.4% advantage. To make deep learning more broadly useful, it is crucial to reduce its training data requirements. Generally, the annotation budget for labeling is far less than the total number of available (unlabeled) samples. For NER, getting unlabeled data is practically free. However, when facing customrized labels, it is especially expensive to obtain annotated data for NER since it requires multi-stage pipelines with sufficiently well-trained annotators (Kilicoglu et al., 2016; Bontcheva et al., 2017).

In such cases, many methods are introduced to tackle this problem. Data augmentation methods explored for NER tasks differ from NLP tasks, either create augmented instances by manipulating a few words in the original instance, such as label-wise token replacement (Dai and Adel, 2020), mention replacement, and using neural generative network (Ding et al., 2020). Despite data augmentation can help extend the amount of data, it still has a limited effect on low-resource data sets. Adapting the meta-learning approach (Finn et al., 2017) to NER can transfer rich-resource domain knowledge to the low-resource domain. However, it not only needs two domains that are relevant but also needs a rich-resource domain to train the model.

In this paper, we use the recently proposed MAML approach (Finn et al., 2017) and extend it with neural generative data augmentation methods to open Chinese data sets. We selected parts of data from few open Chinese data sets to simulate low-resource domain, and firstly propose a neural network augmentation method to extend low-resource domain data sets, after that, we use a meta-learning algorithm to find a good model parameter initialization with those extended open data sets and that could fast adapt to new tasks. When it comes to the adaptation phase, we regard each test example as a new task, build a pseudo training set with data augmentation for it, and fine-tune the meta-trained model before testing.

To summarize our contributions:

- We propose a meta-learning-based approach to tackle Chinese NER with minimal resources.
- We propose an augmentation before the meta-learning approach to augment low-resource training datasets. To our best knowledge, this is the first successful attempt in adapting data augmentation with meta-learning in Chinese NER.
- We evaluate our approach over 5 open Chinese data sets target languages, which cross different source domains. We show that the proposed approach significantly outperforms existing SOTA methods across the board.

2 Related Work

In this section, we review related work in three parts: NER, meta-learning, and data augmentation.

2.1 NER

Generally, NER technology is divided into three stages according to the technological development path: early methods (Sekine and Nobata, 2004)(based on rules and dictionaries), traditional machine learning methods (Morwal and Chopra, 2013; Mccallum and Wei, 2003), and deep learning methods (Li et al., 2020c). We will mainly introduce the research progress of deep learning in NER and focus on recent research trends. When it comes to deep learning, better performance, higher efficiency, and lower transfer cost are the advantages, which are mainly due to its powerful feature representation capability. Deep learning models can automatically learn features that require manual design in traditional methods, which greatly reduces the effort of designing features. At the same time, innovations in the architecture of deep learning methods in other applications can often be applied to current tasks and achieve good results. Components of NER architecture based on deep learning include data representation, context encoder, and tag decoder. In data representation, although one-hot encoding is simple and effective, the representation vector is extremely sparse and difficult to optimize. At present, the word-embedding method is more commonly used, which considers contextual semantic information while avoiding the curse of dimensionality. And there are many open-source word vector models, such as Google Word2Vec (Mikolov et al., 2013), Stanford GloVe (Pennington et al., 2014), etc., which can be used to improve efficiency even performance. Certainly, you can choose whether to train yourself (Yao et al., 2015) or to use open-source (Shen et al., 2017). In addition, in order to solve the problem of new word characterization, (Ma and Hovy, 2016) incorporates character-level characterization methods into word vector characterization. In context encoder, three typical networks are convolutional neural network (CNN), recurrent neural network (RNN), and recursive neural networks. The advantage of CNN is that training and testing are faster when compared with others (Strubell et al., 2017). However, RNN has natural advantages and can learn contextual information. At the same time, the LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Yang et al., 2016) architectures can partially solve the problem of efficiency. Unfortunately, CNN and RNN are not good at dealing with ambiguity problems. At this time, the recursive neural network worked. (Li et al., 2017) introduced a recursive neural network to learn deep structured information, the phrase structures of sentences. In tagger decoder, MLP+Softmax (Akbik et al., 2018) is introduced when the NER task is regarded as a multi-class classification problem. And the most commonly used and optimal method in NER is based on the Conditional Random Field (CRF) model (Zhai et al., 2017). In addition, RNN (Shen et al., 2017) and its variants such as pointer network (Vinyals et al., 2015) are also used as NER decoders. (Shen et al., 2017) pointed out that when there are many types of entities, RNN is better and more efficient than CRF. (Zhang and Yang, 2018) uses a pointer network for sequence labeling tasks and performs segmentation and labeling functions at the same time.

2.2 Chinese NER

In the NER task, Chinese is more difficult and more challenging due to its own characteristics compared with other languages such as English, Spanish, French, German, Japanese, and so on. Difficulties lie in (1) there is no explicit boundary identifiers similar to English space which requires word segmentation, another extremely challenging task; (2) Special English entities may appear in Chinese entity types; (3) The proportion of new words is constantly increasing, and the old labeled corpus is difficult to meet the demand; (4) There are many ambiguities and it is difficult to disambiguate. In recent years, Chinese NER technology has also achieved some results, especially based on deep learning methods. Lexicon is one of the commonly used methods. (Zhang and Yang, 2018) investigated a lattice-structured LSTM model to encode input characters and all potential words obtained from a lexicon that explicitly leverages word and word sequence information. A lexicon-based neural graph network with global semantics is introduced by (Gui et al., 2019) to solve the problem of word ambiguities. For efficiency issues, (Ma et al., 2019) designed a simple but effective method for any neural NER model which requires only subtle adjustment of the character representation layer to introduce the lexicon information. Attention mechanism, transfer learning, multi-task learning, etc. are also used alone or in combination. (Cao et al., 2018) proposed a novel adversarial transfer learning framework to make full use of task-shared boundaries information and exploit self-attention to explicitly capture long-range dependencies between two tokens. (Zhu et al., 2019) introduced a convolutional attention network to capture context information by the local-attention layer and a global self-attention layer. In order to adapt limited data, (Dong et al., 2019) presented a novel multitask bi-directional RNN model combined with deep transfer learning to get transferring knowledge and data augmentation. To solve the problems of out-of-vocabulary and word segmentation errors, a self-attention mechanism is introduced into the BiLSTM-CRF neural network structure to compute similarity on the total sequence consisted of characters and words (Chang et al., 2020). Instead of direct transfer from a source-learned model to a target language while further solving the problem of insufficient data, meta-learning was introduced into Chinese NER. (Wu et al., 2020) utilized a few similar examples to fine-tune the learned model in which a meta-learning algorithm is used to get model parameter initialization. In general, many works show good performance, but problems such as new words and insufficient data still exist.

2.3 Meta-Learning

Different from traditional transfer learning, meta-learning aimed at the model's learning capacity and the obtained general model solve new domain problems by the experiences across other various but data limited domains just like human beings. With its advantages of low resource and strong adaptability, it has become one of the most potential fields of deep learning recently that achieved great success in image classification (Koch et al., 2015), demand prediction (Shi et al., 2020), and reinforcement learning (Finn et al., 2017). There are metric-based models (Strubell et al., 2017), memory-based models (Ravi and Larochelle, 2016), and optimization-based models (Finn et al., 2017) in the pioneering meta-learning studies (Huisman et al., 2021), and we adopted the last one, namely learning adaptable initial parameters of a model. The popular optimization-based technique MAML (Huisman et al., 2021) and the pre-train model were incorporated to address the Chinese NER problems in this paper.

2.4 MAML

MAML proposed a meta-learner and a target-learner, and the gradients that meta-learner accumulated were utilized to update the target learner's gradient. The bilevel optimization strategy of the gradient helped the meta-task with limited data a lot. However, other than neural machine translation (Gu et al., 2018), query generation (Huang et al., 2018) and dialog tasks (Qian and Yu, 2019), there is a limited concentration at such strategies applied in natural language processing. And the applications of MAML are merely at the beginning in the NER field. (Wu et al., 2020) first implemented a cross-lingual NER method based on MAML and achieved SOTA performance over five target languages. Another successful attempt is MetaNER (Li et al., 2020b), a MAML based approach that also demonstrated that the in-domain results could be achieved using only a third of the target data. (Li et al., 2020a) improved MAML in adapting to target tasks with fewer gradient steps via intra-domain, cross-domain and cross-domain three cross-type training.

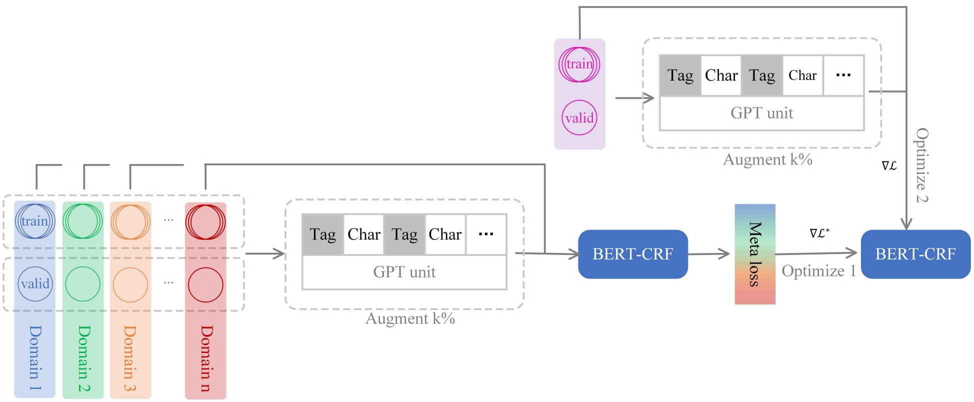

Figure 1: An overview of DAML, which consists of a data augmentation process and a maml training process . During augmentation process, a GPT-2 generation model is used to augment data sets.

2.5 Data Augmentation

Recent works always focused on back translation (Sennrich et al., 2015) and auto augmentation (Cubuk et al., 2018) methods including synonym substitution, random insertion, and random exchange, which generates new corpus by introducing noises or relying on additional knowledge bases. More recently, (Ding et al., 2020) proposed a novel data augmentation approach on NER and POS tagging with the main idea that linearizing labeled sentences. Specifically, they inserted the significant tag in front of the word physically and obtained superior performance after an LSTM based language generation model. Our method is in line with the above approach that fuses both manual labels and semantic information. The difference is a pre-trained generation model was adopted to obtain more abundant synthetic data with labeled sentence linearization, making it more suitable for Chinese datasets.

3 Methodology

NER problem can be seen as a sequence labelling problem which refers to assigning labels or tags to each element of a sequence being passed as an input using an algorithm or machine learning model. This sequence can be words of a sentence passed in the same order as in the sentence. At training steps, given $D_s = \{D_1, D_2, \cdots, D_N\}$, where N refers N low resources from different domains. For each resource D_n, it has annotated raw text X_k as input and a corresponding domain-specific label set Y_k with the BIO schema. Meanwhile, for a target task, which is unseen in training steps. Our ultimate goal is to learn fast and get a good result on the target dataset with low resources. In this section, we present the general form of our algorithm, and the approach we fusion MAML with data augmentation.

3.1 Overview of DAML

Figure 1 shows an overview of our approach, which consists of an augmentation step with the GPT-2 model and a training step with MAML. The main purpose of our method is to learn initial parameters based on various low-resource tasks, such that the model can learn how to quickly solve new tasks with only small training resources. Furthermore, considering training phases with low resources D_s. DAML can (1) Produce good generalization performance on a new task with a small amount of training data. (2) Use low resources with similar labels. Meanwhile, synthetic data using the limited datasets are generated beforehand and later fed into MAML processes to train the model. In addition, the N-way K-shot MAML mechanism allows DAML to learn meta-knowledge and label dependencies from the learning experience across many different low-resource tasks that share the same labels.

In our scenario, we want our model to be able to get a tag sequence Y_k for a raw text input dataset which only providing a few labeled examples for each entity class. In MAML's K-shot learning setting, new tasks with low resources are first augmented by a pre-trained GPT-2 generation model. More specifically, it first linearizes labeled sentences, and a pre-trained language model can be used to learn the distribution of words and tags to generate synthetic training data for the next step. During meta-training, our base model is trained with K samples which contained augmented data and feedback from a corresponding Loss L_k, and

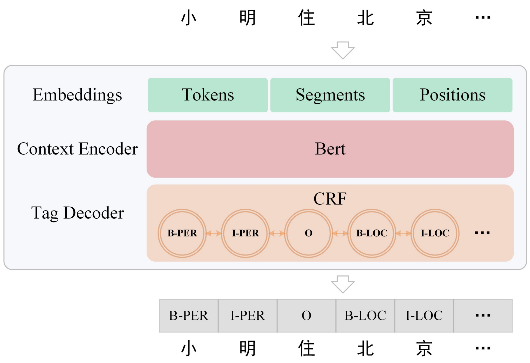

Figure 2: An overview of BERT-CRF constructure.

then the model can improve by the test error. At the end of meta-training, target tasks are augmented by GPT-2 model as well, and meta-performance is measured by the model's performance after learning from K samples. Generally, each task used for meta-testing is held out during meta-training.

3.2 Base Model

Some works have been done with Bert-BiLSTM-CRF which replaces the full connectivity layer in the Bert-CRF with the BiLSTM layer. However, it shows that there was no significant performance difference between Bert-BiLSTM-CRF and Bert-CRF. Besides the network structure of Bert-BiLSTM-CRF takes more resources on the computation. So, in this section, we first give a brief introduction to the BERT-CRF model, which we leverage as the base model in our approach. It produces a clear base structure for the deep learning NER model and it has shown great improvements across various NLP tasks. Figure 2 gives an overview of deep learning-based NER structure. Basically, the structure is mainly divided into two parts, the first part is the BERT structure, with the BERT pre-training language model, each word in the input sentence is converted into a low-dimensional vector form. The second part is the CRF structure, which aims to solve the dependency between the output tags to obtain the global optimal annotation sequence of the text.

We start with BERT (Devlin et al., 2018), or Bidirectional Encoder Representations from Transformers here. BERT is a language model learned with the transformer encoder (Vaswani et al., 2017). It reads the input sequence at once and is effective in automatically learning useful representations and underlying factors from raw data. BERT uses masked language models to enable pre-trained deep bidirectional representations. Given a sentence input, we first use character-based tokenization for Chinese input and then comprise corresponding position embeddings, segment embeddings, and token embeddings as an input representation. All the embeddings will be fine-tuned during the training process. At the output, the low-dimensional vector token representations are fed into the CRF layer for sequence labeling.

There are two phases of model training: pre-training and fine-tuning. For the pre-training phase, this model directly loads BERT-Base-Chinese, a pre-trained model from google which is pre-trained base on entire Chinese Wikipedia $25M$ sentences, raw text without formatting. The structure of the model has 12-layer, 768-hidden, 12-heads, and $110M$ parameters. In fine-tuning phase, we simply train the BERT model with specific inputs and outputs and fine-tune all the parameters end-to-end.

We use the CRF (Lafferty et al., 2001) layer as tag decoders. CRF combines the advantage of graphical modeling and takes the previous context into account when making multivariate output predictions. A CRF layer has a state transition matrix as parameters. With such a layer, we can efficiently use past and future tags to predict the current tag. The probability distribution for CRF can be defined as:

$$P(y_1, \cdots, y_n|X) = \frac{1}{Z(X)} \exp(h(y_1|X) + \sum_{k=1}^{n-1}[g(y_k, y_{k+1}) + h(y_{k+1}|X)] \quad (1)$$

where $Z(x)$ is a normalization factor over all possible tags of x, and $h(y_k|X)$ indicates the probability y_k of the tag at position k which is calculated by the previous softmax layer. $g(y_k, y_{k+1})$ is the transition probability of a tag from states y_k to y_{k+1}.To apply Maximum Likelihood on the negative log function $-logP(y_1, y_2, \cdots, y_n|X)$, we will take the argmin and lean the transition probability.

3.3 Data Augmentation

Retained the label linearization part in (Ding et al., 2020), pre-processed operations are illustrated in Figure 3, in which paired $< tok^1, tag^1 >< tok^2, tag^2 > \cdots$ is converted into a line $< tag^1, tok^1, tag^2, tok^2, \cdots >$ with deleting all the "O" tags and inserting the remaining valid tags starting with "B-" or "I-" before the corresponding characters. After adding special tokens (< BOS > and < EOS >) to the beginning and the end of each sentence, all the sentences were tokenized before feeding into the model. Given that the transformer decoder-based Generative Pre-Training (GPT) model performs better on long text as have been extensively reported (Radford and Narasimhan, 2018; Radford et al., 2019), the pre-processed corpus was put into the GPT-2 model for training and generating. The architecture of GPT-2 small is shown in Figure 3 with 12-decoders. The implementation of the GPT model mainly depends on predicting the next character with only one “Masked Multi Self Attention” block before the “Feed Forward” block in each decoder. For training, two main stages, pre-train and fine-tune are implemented successively with object functions shown as formula 2 where i is set to 1 and 2 correspondings to pre-train and fine-tune respectively. For both large-scale pre-train datasets C_1 and our own fine-tune labeled datasets C_2, the current word y was predicted via m words before it. In pre-train, only L_1 is optimized with the large-scale unsupervised datasets. Taking both L_1 and L_2 into account, the weight parameter λ was set, and L_{total} was calculated as the fine-tune basis for optimization. In this paper, we adopted the GPT-2 model with 24 layers and 345 million parameters and set the embedding size to 1024.

$$L_i(C_i) = \sum_{(x,y)} logP(y|x_1, x_2, ..., x_m), i = 1, 2 \tag{2}$$

$$L_{total}(C) = L_1(C_1) + \lambda * L_2(C_2) \tag{3}$$

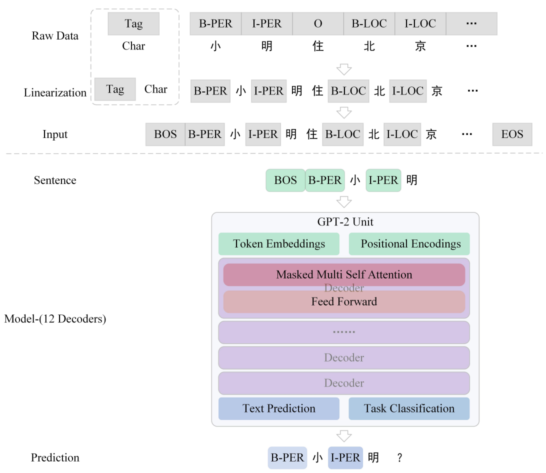

Figure 3: Illustration of Data augmentation with the pre-process pipeline and the augmentation model.

3.4 MAML

In this section, we describe the detail of the MAML approach. The MAML strategy consists of two core phases: a meta-training phase and a meta-adapting phase. First, we elaborate on the meta-training phase and how we set our MAML training tasks up. In effect, augmented data sets are used to enhance the performance of the model in this meta-training process. Then, we describe how to adapt the learned model to the final target task, also known as the meta-adapting phase. The whole process is shown in Algorithm1.

Algorithm 1 Training and Adapting DAML

META-TRAINING
Input: $D_s = \{D_1, D_2, \cdots, D_N\}$, α, β, base model Initialize parameters θ.
Output: base model parameters $\theta*$.
Initialize a deep copy model with the pre-trained base model M_{init}.
while not done **do**
  Sample batch of source training data D_i from D_s.
  for All D_i **do**
    Evaluate $\nabla_\theta L_{T_i}(f_\theta)$ with respect to N examples' evaluation data.
    Compute adapted parameters with gradient descent: $\theta_i^{'} = \theta - \alpha \nabla_\theta L_{T_i}(f_\theta)$
  end for
  Aggregate gradient descent: $mamlgradient = \beta \nabla_\theta \sum_{D_0 \sim D_N} L_{T_i}(f_{\theta_i^{'}})$
end while
Update base model's parameter θ with MAML gradient.

3.4.1 Meta-training Phase

Formally, we divide our data sets into meta-training data sets D_s, the low resources we use to improve our model performance, and final target data sets T_f, the target data we want our model to be able to adapt to. For each data set, it has been split into training parts and evaluation parts. In our scenario, consider the base NER model denoted as f_θ with parameters θ. In the meta-training phase, our approach is going to learn adaptation parameters from the meta-training tasks and its associated dataset $(D_{(i)train}, D_{(i)test})$. The parameters of the temporary model are adapted by AdamW with one or more steps. To achieve a good generalization across a variety of tasks, the model would like to find the optimal θ^* so that the task-specific fine-tuning is more efficient. The loss, denoted as $L_{T_i}(f_\theta)$), depends on the tasks.

3.4.2 Meta-adapting Phase

After meta-training, the model has already learned a model with parameters θ^* with the meta-training domains D_s. The meta-adapting phase tries to learn the distribution between the source domains Dtr and simulated target domains D_{val} using the learned temporary model. It mimics the process of the temporary model being adapted to unseen domains. More specifically, the outer meta-validation loss is computed on the task T_j from the meta-validation domains D_{val} by L_{val}.

4 Experiment

In this section, we first describe our experimental settings. Then, we present our experiment details for the approach used in this paper. Finally, we detail the result on the MSRA dataset and give a comparison for experiments based on various amounts of augmentation data.

4.1 Data Sets

We evaluated the effectiveness of our method on subsets of six wide used Chinese datasets, MSRANER (Levow, 2006), PeopleDairyNER[1], CLUENER (Xu et al., 2020), WeiboNER (Peng and Dredze, 2015), Resume NER (Zhang and Yang, 2018) and BOSONNER (Min et al., 2015), with longer sentences and context-dependent semantics as well, which originated from the newspaper, social media, news, commentary, and financial domains. In particular, in order to verify the effectiveness of our method in the Fewshot scenario, the number of sub-datasets of this article is 2000 (all if less than 2000).

4.2 Implementation Details

To verify the effectiveness of our method in the supervised datasets, we set MSRANER as the target data set. For the base model, we fine-tune on MSRANER data set based on bert opensource model. At the same time, we fine-tune PeopleDairyNER, CLUENER, WeiboNER, and Resume NER four open datasets(we call them training sets in the following parts of the paper.) without any augmented data based on bert opensource model as our "Pre-Train" model. After that, we augment training sets with 50% more amount of sentences with LSTM model and GPT-2 model to fine-tune bert opensource model in both MAML training steps and model fine-tune process as "GPT2+MAML", "LSTM+Pre-Train" and "GPT2+Pre-Train". Next, we augment 0%,25%,50%,75%,100% amount of MSRANER training sentences with LSTM model and GPT-2 model to show the final comparison. As mentioned above, 2000 sentences are randomly split from the original development and test data to verify our methods.

The total experiments used the same hyper-parameters. The models were trained using the AdamW optimizer with a bert learning rate of $3e-5$ and a CRF learning rate of $1e-3$. mMx sequence's length for training data is 128 and 512 for evaluating data. And for MAML processes, we used $\alpha = 0.99$ and $\beta = 0.99$ as well.

We use exact match to evaluate our precision/recall/f1-score result where roughly describing precision is the percentage of correct named-entities found by the NER system, and recall is the percentage of the named-entities in the golden annotations that are retrieved by the NER system. The formula is shown as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.3 Experimental Results

We report the $f1$ results of 7 approaches in Table 1. Our method shows consistent performance improvement for GPT-2 model and MAML combined approach, especially for the smaller sampled sets. For details, firstly, compared with the base model, all other methods show advantages which show advantages for the combination of out-domain and in-domain information. Secondly, compared with the LSTM augmentation method, the GPT2 augmentation method shows advantages. Thirdly, compared with Pre-Train and augmentation method, MAML and augmentation method shows advantages. At last, with GPT-2 augmentation in MAML

[1]https://github.com/zjy-ucas/ChineseNER

Methods	Datasets	0%	25%	50%	75%	100%
Base	MSRA	0.857	-	-	-	-
LSTM+Pre-Train+LSTM	MSRA	0.860	0.870	0.881	0.875	0.870
GPT2+Pre-Train+GPT2	MSRA	0.902	0.909	0.910	0.918	0.913
GPT2+MAML+GPT2	MSRA	0.909	0.912	0.917	0.921	0.915
Pre-Train+LSTM	MSRA	0.869	0.867	0.867	0.879	0.887
Pre-Train+GPT2	MSRA	0.899	0.904	0.904	0.909	0.900
MAML+GPT2	MSRA	0.907	0.906	0.913	0.913	0.901

Table 1: Experiments Results for datasets of MSRA, People's Daily, Weibo, Resume and CLUE. Seven methods are listed with 0%,25%, 50%,75% and 100% datasets.

and Pre-Train stage show advantages when compared with augmentation only in fine-tune stage. Augmentation with the LSTM model shows disadvantages when added in MAML and Pre-Train stage, for the effectiveness of the augmentation quality. Especially, we conduct "Pre-Train+GPT2" and "MAML+GPT2" models to test the BOSON-NER data set, the f1 scores are 0.684 and 0.761 which verifies the effectiveness of the GPT-2 model and MAML combined approach.

5 Conclusion

In this paper, we have shown that the fusion of data augmentation and MAML work well in the NER task. Besides, our method takes full use of out-domain and in-domain information which can apply to low-resource tasks. Continued work can be focused on high-quality data augmentation methods. We hope that DAML will encourage future research to transfer advanced for different tasks.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Kalina Bontcheva, Leon Derczynski, and Ian Roberts. 2017. Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*, pages 875–892. Springer.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192.

Ning Chang, Jiang Zhong, Qing Li, and Jiang Zhu. 2020. A mixed semantic features model for chinese ner with characters and words. *Advances in Information Retrieval*, 12035:356.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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. *arXiv preprint arXiv:2011.01549*.

Xishuang Dong, Shanta Chowdhury, Lijun Qian, Xiangfang Li, Yi Guan, Jinfeng Yang, and Qiubin Yu. 2019. Deep learning for named entity recognition on chinese electronic medical records: Combining deep transfer learning with multitask bi-directional lstm rnn. *PloS one*, 14(5):e0216046.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. 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)*, pages 1040–1050.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. *arXiv preprint arXiv:1803.02400*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Mike Huisman, Jan N van Rijn, and Aske Plaat. 2021. A survey of deep meta-learning. *Artificial Intelligence Review*, pages 1–59.

Halil Kilicoglu, Asma Ben Abacha, Yassine Mrabet, Kirk Roberts, Laritza Rodriguez, Sonya Shooshan, and Dina Demner-Fushman. 2016. Annotating named entities in consumer health questions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3325–3332.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

J. Lafferty, A. Mccallum, and Fcn Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gina-Anne Levow. 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*, pages 108–117.

Jing Li, Billy Chiu, Shanshan Feng, and Hao Wang. 2020a. Few-shot named entity recognition via meta-learning. *IEEE Transactions on Knowledge and Data Engineering*.

Jing Li, Shuo Shang, and Ling Shao. 2020b. Metaner: Named entity recognition with meta-learning. In *Proceedings of The Web Conference 2020*, pages 429–440.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020c. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

P. H. Li, R. P. Dong, Y. S. Wang, J. C. Chou, and W. Y. Ma. 2017. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Ruotian Ma, Minlong Peng, Qi Zhang, and Xuanjing Huang. 2019. Simplify the usage of lexicon in chinese ner. *arXiv preprint arXiv:1908.05969*.

X. Ma and E. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf.

A. Mccallum and L. Wei. 2003. Early results for named entity extraction with conditional random fields. *Proc of Conll*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Kerui Min, Chenggang Ma, Tianmei Zhao, and Haiyan Li. 2015. Bosonnlp: An ensemble approach for word segmentation and pos tagging. In *Natural Language Processing and Chinese Computing*, pages 520–526. Springer.

S. Morwal and D. Chopra. 2013. Nerhmm: A tool for named entity recognition based on hidden markov model. *International Journal on Natural Language Computing*, 2(2):43–49.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

M. Peters, W. Ammar, C. Bhagavatula, and R. Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Matthew Peters, M. Neumann, M. Iyyer, M. Gardner, and L. Zettlemoyer. 2018. Deep contextualized word representations.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. *arXiv preprint arXiv:1906.03520*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*.

Jiatu Shi, Huaxiu Yao, Xian Wu, Tong Li, Zedong Lin, Tengfei Wang, and Binqiang Zhao. 2020. Relation-aware meta-learning for market segment demand prediction with limited records. *arXiv preprint arXiv:2008.00181*.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.

Qianhui Wu, Zijia Lin, Guoxin Wang, Hui Chen, Börje F Karlsson, Biqing Huang, and Chin-Yew Lin. 2020. Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9274–9281.

Liang Xu, Qianqian Dong, Yixuan Liao, Cong Yu, Yin Tian, Weitang Liu, Lu Li, Caiquan Liu, Xuanwei Zhang, et al. 2020. Cluener2020: fine-grained named entity recognition dataset and benchmark for chinese. *arXiv preprint arXiv:2001.04351*.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. 2015. Biomedical named entity recognition based on deep neutral network. *Int. J. Hybrid Inf. Technol*, 8(8):279–288.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.

Yuying Zhu, Guoxin Wang, and Börje F Karlsson. 2019. Can-ner: Convolutional attention network for chinese named entity recognition. *arXiv preprint arXiv:1904.02141*.