Adversarial Named Entity Recognition with POS label embedding

1st Yuxuan Bai
Nanjing University of Posts and Telecommunications
Nanjing, China
1018041124@njupt.edu.cn

2nd Yu Wang
Nanjing University of Posts and Telecommunications
Nanjing, China
2017070114@njupt.edu.cn

3rd Bin Xia
Nanjing University of Posts and Telecommunications
Nanjing, China
bxia@njupt.edu.cn

4th Yun Li✉
Nanjing University of Posts and Telecommunications
Nanjing, China
liyun@njupt.edu.cn

5th Ziye Zhu
Nanjing University of Posts and Telecommunications
Nanjing, China
1015041217@njupt.edu.cn

***Abstract*—Named Entity Recognition (NER) is dedicated to recognizing different types of named entity. Previous works have shown that part-of-speech, as an important feature, provides complementary syntactical information to NER systems. However, these studies suffer from two limitations: (i) the previous models do not consider the noise from part-of-speech; (ii) the previous models need to re-extract features from token representations. In this paper, we propose a novel approach that can alleviate the above issues as well as make full use of part-of-speech features via attention mechanism and adversarial training. We evaluate our model on three NER datasets, and the experimental results demonstrate that our model achieves a state-of-the-art F1-score of Twitter dataset while matching a state-of-the-art performance on the CoNLL-2003 and Weibo datasets.**

***Index Terms*—Named Entity Recognition, Attention mechanism, Adversarial training.**

I. Introduction

Named Entity Recognition (NER) aims to identify named entities in text and classify named entities into predefined entity types (e.g., *person, organization, location*). NER has been considered as a core step for downstream tasks such as relation extraction [1] and co-reference resolution[2].

Recently, neural models have demonstrated strong abilities in NER task [3], [4]. The standard models [5], [6], [7] are the *encoder-decoder* structure that uses Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) for sentences encoding and Conditional Random Fields (CRF) for label decoding. Moreover, Long Short Term Memory Network (LSTM), as a variant of RNN, is also widely used in Natural Language Processing (NLP) tasks. The LSTM+CRF-based models achieve a higher F1-score in CoNLL-2003 datasets, considering various features like uppercases, lowercases, and affix [8], [9]. In addition, Syntactic features, such as part-of-speech helps the models deal with word disambiguation [10]. Many studies [3], [8], [11] utilize part-of-speech as additional features to improve the performance of NER systems.

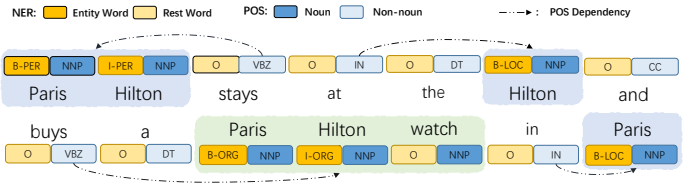


Fig. 1. Example of NER and POS label. For NER system, *PER, LOC*, and *ORG* stand for a person, location, and organization. For POS system, *NP, VP, DT*, and *IN* stand for noun, verb, article, and preposition respectively.

Part-of-speech features have a wealth of complementary information for NER systems: (1) ***Part-of-speech dependencies:*** Due to the polysemy, it is difficult for NER models to classify the entity type of polysemant, while the part-of-speech of rest words [1] can assist systems to complete the task. For example, as shown in Fig.1, we want to recognize *person, location* and *organization* in sentences. "Hilton Paris" may be recognized as a *person, location* or an *organization*. Directly classifying the entity is difficult, but the entity before verb "stay" and "buy" may be recognized as a *person*, while a *location* may appear after the preposition "in" or "at". Therefore, with the assistance of part-of-speech dependencies between different words at the arbitrary position in a sentence, the complexity of the disambiguation problem can be reduced. (2) ***Tasks-shared features:*** Both NER and Part-Of-Speech (POS) are sequence labeling tasks where tags have correlations, especially on boundary information. For example, [Paris Hilton]$_{PER}$ in Fig.1, the noun boundary in POS task and the entity boundary in NER task are identical. Boundary assigns *B-*, *I-* to the Beginning and Inside of the words, respectively. However, POS and NER have task-specific features, such as "Paris Hilton watch" is the words boundary noise that may even inhibit NER systems. (3) ***Noun constraint:*** A Named Entity

[1]rest word: the word is not a Named Entity.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(NE) can only be a proper **noun** at first [12]. Without the noun constraint, NER systems may incorrectly label *non-noun* words as entities words.

Although the part-of-speech features can provide a lot of syntactical information to NER models, there still exist unresolved problems. The capability of different feature embedding methods varies greatly on extracting semantic information. The embedding methods [8], [10] represent tokens by combining word embeddings with part-of-speech embeddings, which leads to mixing the part-of-speech features and noise. Under such circumstances, the models have to re-extract useful information from token representations, whereas the extraction ability cannot be evaluated and controlled.

To effectively utilize complementary part-of-speech information in NER systems, we propose *Adversarial NER with POS label embedding* **(ANP)**, which includes four modules: (i) To capture **part-of-speech dependencies**, we first employ POS label embeddings [13] to represent part-of-speech features, and then *self-attention* mechanism is adopted to capture dependencies between different words at any position in the sentence. (ii) To distill **tasks-shared** features and suppress task-specific noise, we take advantage of the *adversarial training* [14] and *task-attention* mechanism for mapping *tasks-shared* information between POS and NER into shared feature space. (iii) For **noun constraints**, we add the constraint based on the CRF formula of NER task to improve the recall of the model. (iv) Regularization item is also adopted to ensure that the information extracted by the task-specific Encoder differs from the Shared Encoder [15].

We summarize the major contributions as follows:

- We propose a novel adversarial neural network for NER task, which deals with the noise caused by simple combination of embeddings and effectively utilize *tasks-shared* information for different NER models.
- We employ *self-attention* mechanism to capture part-of-speech dependencies, to our best knowledge, it is the first time to extract part-of-speech dependencies via *self-attention* mechanism for NER task. our proposed **ANP** no longer needs to re-extract features from the concatenation token representations.
- We conduct ablation studies to illustrate the effectiveness of part-of-speech dependencies and adversarial training by comparative experiments.

II. Related Works

Recently, neural networks have been used for NER and achieved state-of-the-art results. Collobert [3] proposed first neural network model that achieved 89.31 % F1-score on English CoNLL-2003 dataset. Afterward, plenty of studies [5], [4], [6], [9] used neural network methods and achieved promising results.

The **part-of-speech features** as syntactical information should also be explicitly considered as contextual features of each word in sentences. Part-of-speech were used in early NER systems [16] as well as embedded into words representations [3], [8], [11]. Gustavo Aguilar [10] utilized grammatical information, such as POS and dependency roles, achieved the state-of-the-art on the Workshop on Noisy User-generated Text 2017 dataset (WNUT).

Adversarial training was first proposed by Goodfellow [14] in the image classification task. In recent studies, adversarial training has also been applied in many typical NLP tasks, which mainly contains two types of functions: one is to enhance model robustness [17], [18], [19], and another is to grasp the consistency among different tasks [20], [21], [22].

Attention mechanism was initially applied to end-to-end Neural Machine Translation, where Vaswani [23] proposed self-attention to draw global dependencies and achieved the state-of-the-art result in translation tasks. Afterward, many tasks, like Semantic Role Labeling [24], [25] and Relation Extraction [26], took advantage of self-attention to capture the dependencies between words or sentences. Enlightened by these works, we use *self-attention* mechanism to capture the part-of-speech dependencies of the sentences.

III. Method

Overview

In this section, we introduce the overall architecture of **ANP** in detail. As shown in Fig.2, three different Encoders are employed to encode a sentence (Section III-A2). Then, the *adversarial training* and *task-attention* are used to map the **tasks-shared** features into the shared feature space (Section III-B). Subsequently, the sequence of part-of-speech tags aligned with the sentence is represented by POS label embedding, and we obtain **part-of-speech dependencies** via *self-attention* mechanism (Section III-C1). Finally, Task Decoder, comprised of one CRF for per task, takes the corresponding Encoder features as inputs and then maximize the log-likelihood of the entire label sequence. We control the probability of emission function by applying **noun constraint** into the CRF basic formula (Section III-D).

A. Sentence Representation

1) Word Embedding: Following the [5], given a training set $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$, each input sentence $x_j = \{w_1, w_2, \ldots, w_n\}^T$ is converted into a sequence of embedding vectors $\mathbf{E}^{n \times d_w} = \{e_1, e_2, \cdots, e_n\}^T$ by Continuous Bag-of-words [27], where w_i denotes the i_{th} word in the sentence x_j, and d_w is the dimension of word embedding.

2) Sentence Encoder: As shown in Fig.2, there are three encoders designed to encode sentences into three different high-dimensional spaces (i.e., the NER task, the POS task, and tasks-shared features). In this paper, after obtaining the sentences $\mathbf{E}^{n \times d_w}$, we apply Bidirectional LSTM (Bi-LSTM) to encode, where Bi-LSTM can obtain contextual information for better context-sensitive representations [6].

More specifically, for each word embedding e_i, the hidden state E_i is formed by concatenating the $LSTM^f$ forward di-

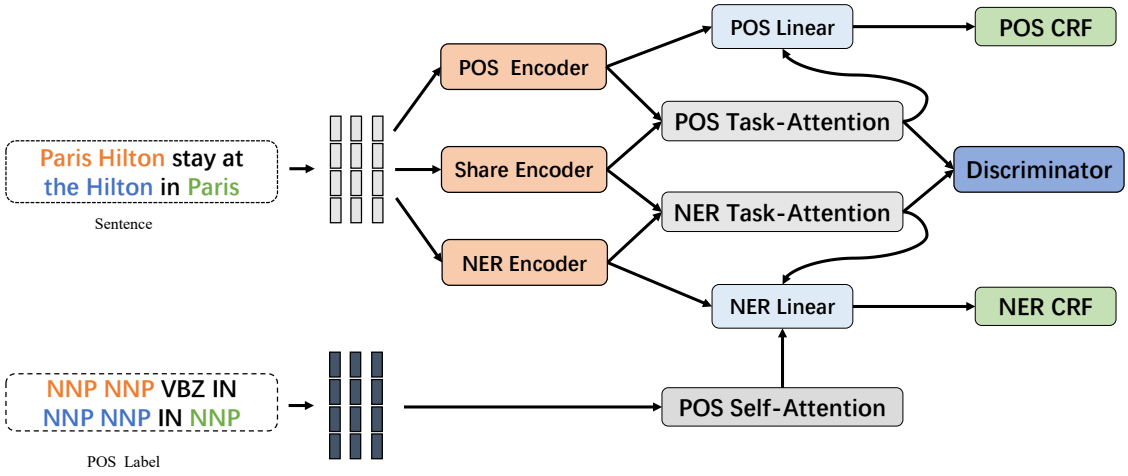


Fig. 2. The main architecture of **ANP**.

rection hidden output h_i^f and the $LSTM^b$ backward direction hidden output h_i^b.

$$h_i^f = LSTM^f(e_i, h_{i-1}), \tag{1}$$

$$h_i^b = LSTM^b(e_i, h_{i+1}), \tag{2}$$

$$E_i(\theta_E) = h_i^f \oplus h_i^b, \tag{3}$$

where θ_E is the parameters of Sentence Encoder and $\oplus$ is the operation of concatenation.

For simplicity, we denote E^{ner}, E^{pos}, and E^{sha} as the representations of sentences Encoders for NER, POS, and Task-Shared respectively.

B. Adversarial training and task-attention

To extract *tasks-shared* information and suppress task-specific noise, we use the idea of the *adversarial training* [18], [21] into **ANP**, as shown in Fig. 2. We apply the Share Encoder as a generative network to map the *tasks-shared* features to the shared feature space. Discriminator tries its best to distinguish which task the training sentence comes from, and the *tasks-shared* features are devoted to mislead Discriminator.

Besides, to strengthen the discriminability of Discriminator, we add the disturbance from task-specific features on tasks-shared features, where the disturbance fuses E^{task} (Section III-A2) with E^{sha} via *task-attention*. After *adversarial training*, the POS task and NER task convergence, and the output $\mathcal{R}$ of *task-attention* can successfully confuse Discriminator, where $\mathcal{R}$ is the authentic *tasks-shared* features:

$$\mathcal{R} = \text{softmax}(\frac{E^{sha} \cdot E^{task^T}}{\sqrt{v}}) \cdot E^{task}, \tag{4}$$

where $task = \{POS, NER\}$, and $\frac{1}{\sqrt{v}}$ is a scaling factor [23].

Discriminator takes $\mathcal{R}$ as input features to identify the sentence comes from the NER task or POS task. All *task-attention* features $\mathcal{R}$ are merged via a max-pooling layer $Max(\cdot)$ and a Nonlinear function ReLU. Finally, after the softmax layer, $\hat{r}^{task}$ indicates the probability of the training sentence comes from the current task. For simplicity, these operations are denoted $\mathcal{J}(\cdot)$.

$$\hat{r}^{task}(\theta_{\mathcal{J}}) = \text{softmax}(Relu(Max(\mathcal{R}))) = \mathcal{J}(E^{task}; E^{sha}), \tag{5}$$

where $\theta_{\mathcal{J}}$ represents the parameters of Discriminator.

First, we need a convergent Discriminator to distinguish which task the training sentence comes from. At the same time, we expect the Share Encoder to produce sentences representations that the Discriminator cannot distinguish which task the sentence comes. In addition, the POS Encoder and NER encoder produce sentences representations to complete the POS task and NER task. So the objective of Discriminator is a min-max process, and the loss function is:

$$\mathfrak{L}_{adv} = \min_{\theta_E}(\max_{\theta_{\mathcal{J}}}(r \cdot \log(\hat{r}^{ner}) + (1-r) \cdot \log(\hat{r}^{pos})), \tag{6}$$

$$r = \begin{cases} 1 & if \ \ task \ \ is \ \ NER \\ 0 & if \ \ task \ \ is \ \ POS. \end{cases}$$

Moreover, we apply regularization item to assure that the information extracted by the Tasks-Specific Encoder E^{task} differs from the Shared Encoder E^{sha} [15], [21],

$$\mathfrak{L}_{regu} = \left\| E^{task} \cdot E^{sha^T} \right\|_F^2, \tag{7}$$

where $\|\cdot\|_F^2$ represents the squared Frobenius norm.

C. Part-of-speech Dependencies Representation

For each given NER task training sentence x_j, we want to capture part-of-speech dependencies between words w_i and w_k. We first search the sequence o_j of part-of-speech tags corresponding x_j in lexicon (Section IV-A). Then we represent the part-of-speech tags using POS label embeddings [13], which calculate the distribution of tags. Finally, **ANP** adopts *self-attention* mechanism to capture the dependencies. Note that part-of-speech dependencies do not appear in the POS task.

1) POS label embedding: We also take advantage of Mikolov's word2vec model to pre-train the tags o_j and obtain POS label embedding sequence $\mathbf{P}_j^{n\times d_p} = \{p_1, p_2, \ldots, p_n\}^T$, where d_p is the dimension of the embedding.

2) Part-of-speech Dependencies: Polysemy often appears in sentences, where the meaning of the word, as well as the entity type, changes with the contexts. Directly classifying the entity type of polysemant is difficult. However, the adjacent rest words like verb, preposition, and article can assist the model to complete the classification. Thus, we utilize part-of-speech dependencies to describe the part-of-speech correlation between words.

Self-attention mechanism is widely used in NLP due to its ability to draw global dependencies [23]. In fact, the attention represents a relation between words or sentences. Therefore, *self-attention* mechanism is suitable for capturing part-of-speech dependencies between different words at arbitrary positions in the sentence.

Overall, we follow Vaswani [23], where the $Q = K = V =$ POS label Embedding. In detail, for a sentence x_j and its corresponding POS label embeddings $\mathbf{P}_j = \{p_1, p_2, \ldots, p_n\}^T$, the weights of part-of-speech dependencies for each pair of words are defined as follows,

$$\mathbf{D}(p_i, p_k) = \frac{\exp(p_i p_k^T)}{\sum_{i=1}^{n} \exp(p_i p_k^T)}, \tag{8}$$

$$\mathbf{A}(p_i) = \sum_{k=1}^{n} \mathbf{D}(p_i, p_k) \cdot p_k. \tag{9}$$

Then we obtain attention weights matrix $\mathbf{D}^{n\times n}$, where $\mathbf{D}(p_i, p_k)$ represents part-of-speech dependencies of the position i and k. $\mathbf{A}$ is the weighted sum of POS label embedding.

In addition, to obtain sufficient grammatical information and representations focusing on different dependencies, *multi-head attention* is leveraged in our model. The *multi-head attention* consists of q heads $(h^1, h^2, \cdots, h^q)$, where each head is expected to learn a function of different dependency. Following Vaswani [23], we linearly project the POS label embedding $\mathbf{P}$ and then calculate the dependencies $\mathbf{A}$ of each head. Finally, these dependencies are concatenated and projected again. The operation is described as follows:

$$\mathcal{A}(\theta_{\mathcal{A}}) = Concat(\mathbf{A}^{h_1}, \cdots, \mathbf{A}^{h_q}) W^o, \tag{10}$$

$$\mathbf{A}^{h_q} = \mathbf{A}(\mathbf{P} \cdot W^q), \tag{11}$$

where W^q, W^o are linear function. In the end, we denote $\mathcal{A}$ as the part-of-speech dependencies and $\theta_{\mathcal{A}}$ as all parameters in the module.

D. Task Decoder

We have obtained the task-specific representations E^{task} (section III-A2), *tasks-shared* features $\mathcal{R}$ (section III-B) and part-of-speech dependencies $\mathcal{A}$ (SectionIII-C2).

In this section, we adopt CRF as POS / NER Decoder [6], [5]. CRF is an undirected probabilistic graph model, which attempts to model the conditional probability of multiple variables given observation values. With the transition feature function G and status feature function M, CRF is widely used to solve the sequence labeling.

Given input sequence $\mathbf{x}$ and output sequence $\mathbf{y}$, a score function is defined as follows:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^{|\mathbf{y}|} G_{y_{i-1}, y_i} + \sum_{i=1}^{|\mathbf{x}|} u_i M_{x_i, y_i}, \tag{12}$$

where G_{y_{i-1}, y_i} is the score of a transition from y_{i-1} to y_i, and M_{x_i, y_i} is the emission score from x_i to y_i.

In this paper, the input sequence $\mathbf{x}$ of CRF is $\mathcal{T}$, which is defined as follows:

$$\mathcal{T}(\theta_{\mathcal{L}^{task}}) = \begin{cases} \mathcal{L}^{ner}(E^{ner}; \mathcal{A}; \mathcal{R}) & if\ task\ is\ NER \\ \mathcal{L}^{pos}(E^{pos}; \mathcal{R}) & if\ task\ is\ POS, \end{cases} \tag{13}$$

where $\mathcal{L}(\cdot)$ is a fully connected neural network, and $\theta_{\mathcal{L}^{task}}$ is the $\mathcal{L}^{task}$ parameters.

In order to avoid incorrectly labeling *non-nouns* as entity words, we use **noun constraints**. When the task is NER task, we add u_i as soft weights to constrain the emission score in Eq. 12 by checking if the word w_i is a noun, which effectively improves the recall of **ANP**. As well as the u_i equals 1 when the task is POS task.

$$u_i = \begin{cases} 1, & if\ w_i\ is\ noun \\ \frac{1}{|W|} & if\ w_i\ is\ not\ noun, \end{cases} \tag{14}$$

where $|W|$ is the number of *non-noun* words in the lexicon (Section IV-A).

Subsequently, a softmax layer is leveraged to calculate the conditional probability of label sequence $\mathbf{y}$ given hidden state $\mathbf{x}$, the probability is described as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp^{s(\mathbf{x}, \mathbf{y})}}{\sum\limits_{\tilde{\mathbf{y}} \in \mathbf{Y}} \exp^{s(\mathbf{x}, \tilde{\mathbf{y}})}}, \tag{15}$$

where $\mathbf{Y}$ is a set of all possible label sequences. Our objective is to find an optimal label sequence by the first-order Viterbi algorithm. NER task and POS task are trained in a supervised manner by minimizing the loss functions:

$$\mathfrak{L}_{task} = -\sum_{\mathbf{x} \in \mathcal{C}^{task}} \log(p(\mathbf{y}^{task}|\mathbf{x})), \tag{16}$$

where $task \in \{NER, POS\}$, $\mathcal{C}^{task}$ is the sampled sentences for NER or POS.

E. Joint Training

We finally combine all the loss functions $\mathfrak{L}_{adv}$ (Eq.6), $\mathfrak{L}_{regu}$ (Eq.7) and $\mathfrak{L}_{task}$ (Eq.16), then jointly optimize the model using Adam [28]:

$$\mathfrak{L} = \sum_{task \in \{NER, POS\}} \mathfrak{L}_{task} + \gamma \mathfrak{L}_{adv} + \lambda \mathcal{L}_{regu}, \tag{17}$$

where γ and λ are hyperparameters.

Note that the input of POS Labeling Embedding is gold tags rather than the output of POS task lest error accumulation. In

Algorithm 1: Training Detail

Data: Dataset $\mathcal{X} = \{x_1, x_2, \cdots, x_m\}$, NER gold tags $\{y_1, y_2, \cdots, y_m\}$ and part-of-speech tags $\mathcal{P} = \{o_1, o_2, \cdots, o_m\}$ aligned with $\mathcal{X}$

Input: Sentence Encoder E with parameters $\theta_{E^{ner}}, \theta_{E^{pos}}, \theta_{E^{sha}}$; Part-of-speech dependencies $\mathcal{A}$ with parameters $\theta_{\mathcal{A}}$; Adversarial Training Discriminator with parameters $\theta_{\mathcal{J}}$; Linear concatenation layer $\mathcal{L}$ with parameters $\theta_{\mathcal{L}^{task}}$; hyperparameters γ, λ; learning rate δ_1, δ_2, $task \in \{POS, NER\}$; iterations T

Initialization parameter $\{\theta_{E^{ner}}, \theta_{E^{sha}}, \theta_{\mathcal{A}}, \theta_{\mathcal{L}^{ner}}\} \triangleq \theta_1$, $\{\theta_{E^{pos}}, \theta_{E^{sha}}, \theta_{\mathcal{L}^{pos}}\} \triangleq \theta_2$.

for $t = 1, 2, \ldots, T$ **do**

  for *task in* {*POS, NER*} **do**

    if *task is NER* **then**

      Random sampling NER training data $\mathcal{X}_1$

      Get $E^{ner}(x_j)$, $E^{sha}(x_j)$, $\mathcal{A}$, $\mathcal{R}$ from each sentence x_j in $\mathcal{X}_1$ and part-of-speech aligned o_j

      Calculate $\mathfrak{L}_{adv}$

      $\theta_{\mathcal{J}} \leftarrow \theta_{\mathcal{J}} + \delta_1 \nabla \mathfrak{L}_{adv}$

      Calculate $\mathfrak{L}_{adv}$, $\mathfrak{L}_{regu}$, $\mathfrak{L}_{ner}$

      $\mathfrak{L}_1 = \mathfrak{L}_{ner} + \gamma \mathfrak{L}_{adv} + \lambda \mathfrak{L}_{regu}$

      $\theta_1 \leftarrow \theta_1 - \delta_2 \nabla_{\theta_1} \mathfrak{L}_1$

    else

      Random sampling POS training data $\mathcal{X}_2$

      Get $E^{pos}(x_j)$, $E^{sha}(x_j)$ from each sentence x_j in $\mathcal{X}_2$

      Calculate $\mathfrak{L}_{adv}$

      $\theta_{\mathcal{J}} \leftarrow \theta_{\mathcal{J}} + \delta_1 \nabla \mathfrak{L}_{adv}$

      Calculate $\mathfrak{L}_{adv}$, $\mathfrak{L}_{regu}$, $\mathfrak{L}_{pos}$

      $\mathfrak{L}_2 = \mathfrak{L}_{pos} + \gamma \mathfrak{L}_{adv} + \lambda \mathfrak{L}_{regu}$

      $\theta_2 \leftarrow \theta_2 - \delta_2 \nabla_{\theta_2} \mathfrak{L}_2$

Output: Our proposed model **ANP**

the implementation process, we randomly sample the training set and take turns to train the POS task and NER task. During training, the Viterbi algorithm [5] is used to infer the label sequence. In each iteration, for NER task, $\theta_{E^{ner}}, \theta_{E^{sha}}, \theta_{\mathcal{A}}, \theta_{\mathcal{J}}$, and $\theta_{\mathcal{L}^{ner}}$ will be updated, while $\theta_{E^{pos}}, \theta_{E^{sha}}, \theta_{\mathcal{J}}$, and $\theta_{\mathcal{L}^{pos}}$ will be updated for POS task.

IV. Experiment

A. Datasets

We evaluate **ANP** on three datasets, (i) the CoNLL-2003 NER task [29] for English (a benchmark sequence labeling task); (ii) two social media datasets including Chinese Weibo NER (Weibo) dataset [30], [31] and English Twitter NER (Twitter) dataset [32], [33]. Moreover, the CoNLL-2003 dataset provides gold POS tags. As for Weibo and Twitter datasets, the corresponding POS tags are tagged via LTP-Cloud [2] and Owoputi [34][3]. To take full advantage of POS tags boundary information, we preprocess the tags, such as transfer "NNP NNP" to "B-NNP I-NNP".

TABLE I
Statistics of the datasets.

Dataset	Train	Dev	Test	Type
CoNLL-2003	14999	3464	3679	4
Weibo	1350	270	270	9
Twitter	1900	240	254	11

B. Settings

For evaluation, we adopt the metrics of F1-score, where the label of the entity is correct only when it matches the gold entity exactly. Our model uses two Bi-LSTM layers as sentences Encoder whose hidden dimension is 256. For the Embedding, we search from [100,128,200,256,300] for word Embeddings and POS label embeddings. Adam [28] is employed with learning rate $\beta = 0.01$, decay rate $\rho = 1e^{-4}$ and mini-batch size$= 50$. We utilize the grid search to adjust hyperparameters. Specifically, with other hyperparameters fixed, we vary λ, γ [1,10,50,100], dropout [0.01-0.5], and the number of heads (Section III-C2) [4-8] to optimize algorithm.

We implement several LSTM+CRF-based baselines for comparison with **ANP**. In the first group experiments, we evaluate the performance of with or without POS labels [8], [35]. In the second group experiments, all models are LSTM+CRF-based, such as LSTM+CRF [36], Bi-LSTM+CRF [6], and Bi-LSTM+CRF+ELMo [7]. Moreover, we utilize ablation test to demonstrate the effectiveness of each module in **ANP**. In order to avoid random deviation as much as possible, the experimental results in this paper are the average of multiple experiments.

C. Result

In this section, we show the performances of **ANP** and the other NER systems mentioned above. The comparisons of Shao [8] with Changpinyo [35] are shown in Table II. Changpinyo [35] adds part-of-speech features but gets the opposite effect with Shao [8], illustrating the different part-of-speech utilization methods may result in different effects. Therefore, how to effectively take advantage of the part-of-speech features is very important. Then, we compare **ANP** with previous ones that use Bi-LSTM+CRF as the basic structure. Peters [7] proposes Bi-LSTM+CRF+ELMo structure and achieve LSTM+CRF-based state-of-the-art performance, demonstrating the effectiveness of contextual representations. Language modeling has been shown useful for NLP task. From Table II, we can observe that **ANP** achieves higher F1-score

[2]LTP-Cloud (Language Technology Platform Cloud), is developed by the Research Center for Social Computing and Information Retrieval at Harbin Harbin Institute of Technology (HIT-SCIR).

[3]This POS tagger has custom labels that are suitable to SM data (i.e., the tagger considers emojis, hashtags, URLs and others).

TABLE II
EXPERIMENTAL RESULTS ON CONLL-2003, WEIBO, AND TWITTER DATASETS.

Structures	Methods	with POS	F1(CoNLL)	F1(Weibo)	F1(Twitter)
Bi-LSTM+softmax	[8]	N	87.56	47.24	78.25
		Y	88.35	47.26	78.57
Bi-GRU+CRF	[35]	N	88.24	47.35	79.56
		Y	87.99	47.32	79.88
LSTM+CRF	[36]	N	90.35	50.23	80.03
Bi-LSTM+softmax	[6]	N	91.35	50.31	80.27
	[11]	Y	86.03	-	79.28
Bi-LSTM+CRF+ELMo	[7]	N	92.24	-	81.54
Bi-GRU+CRF+transfer	[33]	N	86.73	-	83.56
BiLSTM-MMNN	[31]	Y	-	54.50	-
LSTM+CRF	[30]	Y	-	**55.28**	-
Bi-LSTM+CRF	**ANP**	Y	**92.86**	54.98	**83.92**

We report the results of **ANP**. The structure of the comparison models are Bi-LSTM+CRF and other similar deployments. In "with POS" column, N denotes without POS tags, and Y denotes with POS tags. Due to the particularity of language and method, for instance, the International Phonetic Alphabet (IPA) features cannot be employed in Chinese, part of the experiments cannot be re-implemented completely.

TABLE III
THE EXPERIMENTAL RESULTS OF ABLATION STUDY ON THREE DATASETS (F1-SCORE).

Models	CoNLL	Weibo	Twitter
ANP	92.86	54.98	83.92
No Frob	92.33	53.57	81.53
No Adv	92.02	53.49	81.48
No pos label	91.68	53.44	80.84
Concat directly	91.05	52.89	80.52
No noun constraint	91.83	53.03	80.89

on CoNLL-2003 dataset and Twitter over the previous Bi-LSTM+CRF state-of-the-art. We will discuss later why **ANP** inferior to the model in [30] on the Weibo dataset in Section V.

D. Ablation Study

In ablation study, we use "No Frob", "No Adv", "No pos label" in Table III to represent removing Frobenius norm, Adversarial Training (i.e., POS task) and POS label embedding respectively. Table III describes that the Frobenius norm is critical to prevent redundant information from the *adversarial training* process. In addition, removing the *adversarial training* hurts the performance significantly. Moreover, the critical claim is part-of-speech dependencies in this paper. So we evaluate it by *No pos label*, and the results are worse than **ANP** on each dataset. In order to prove part-of-speech noise caused by concatenation, we concatenate POS embeddings and word embeddings directly as token representation when training NER task. It performs very poorly under the same configurations, as shown in Table III *Concat directly*. As shown in Table III, we can see that the **noun constraint** improve the recall effectively.

TABLE IV
EXPERIMENTAL RESULTS BY ADDING MANUALLY LABEL PART-OF-SPEECH AND EXCLUDE COMPLICATED SENTENCES.

Operation	F1
Weibo	54.98
Weibo+manual	55.12
Weibo+manual+Exclude	55.15

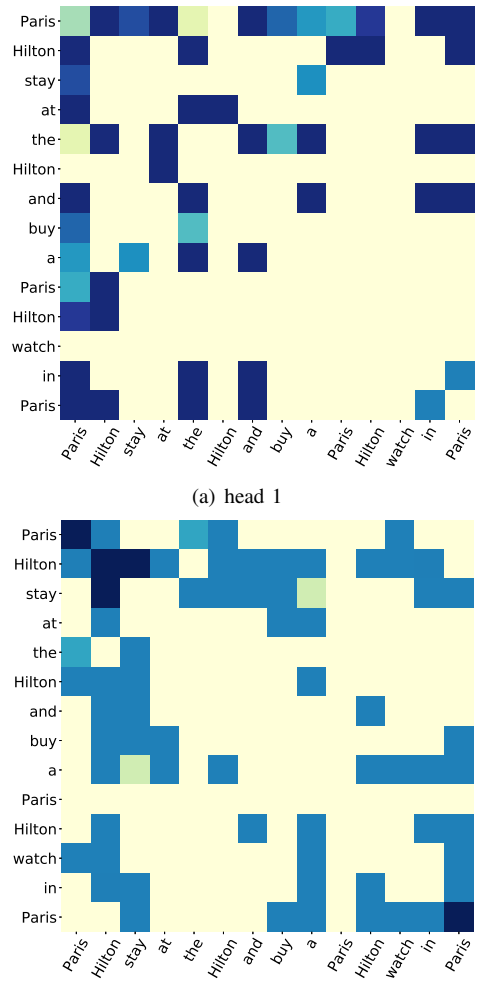
V. ANALYSIS

To better understand our model, we introduce the following questions:

- What does the POS label embedding really learn via *self-attention* mechanism?
- What does the incorrect part-of-speech label cause?
- What is the role of *adversarial training* in our model?

For the first question, as shown in Fig.3, different heads learn the various grammatical dependencies. For example, when the input sentence is "[Paris Hilton]$_{PER}$ stays at the [Hilton]$_{LOC}$ and buys a [Paris Hilton]$_{ORG}$ watch in [Paris]$_{LOC}$", the preposition "in" and "at" focuses the [Hilton]$_{LOC}$ and [Paris]$_{LOC}$ respectively as well as verb "stay" and "buy" correctly picks up [Paris]$_{PER}$, as shown in Fig.3(a). With the help of **part-of-speech dependencies** between rest words and entity words, our model reduces the complexity of identifying the entity type of polysemant.

For the second question, on Weibo dataset in Table II, the improvement of **ANP** is not significant. Considering that there is a certain error in part-of-speech by the LTP-Cloud, we manually label part-of-speech $1/3$ of the Weibo dataset and exclude complicated sentences with more than 20 words to reduce the number of incorrect part-of-speech labeling. Table IV presents that, although we only manually annotated a



(a) head 1

(b) head 2

Fig. 3. Visualization of the attention distribution for the POS label embedding. There are two heads of *attention*.

part of the dataset, we can still see improvement in the results, especially without complicated sentences. The performance improvement demonstrates that if all the correct POS labels are available on Weibo dataset, our model will perform better.

For the last question, to understand what the features are extracted by Share Encoder, we complete POS and NER with the hidden outputs of Share Encoder on CoNLL-2003 dataset after the NAP reaches convergence. Then results are shown in the Table V. The result indicates that the *tasks-shared* features can also complete subtasks, such as POS task and NER task. In addition, the loss figure of model convergence shows the process of *adversarial training*, as shown in Fig. 4. The Discriminator loss reaches the minimum value at the 100 iteration and then oscillates. In other words, as POS task and NER task converge gradually, and the Discriminator loses its ability to distinguish, the features $\mathcal{R}$ (SectionIII-C2) are authentic *tasks-shared* features.

TABLE V
EXPERIMENTAL RESULTS WHEN SHARE ENCODER IS AS SPECIFIC-TASK ENCODER.

CoNLL		Weibo		Twitter	
POS	NER	POS	NER	POS	NER
90.35	88.75	-	45.3	-	81.84

Due to the lack of gold labels for Weibo and Twitter datasets, so we does not show the POS experimental results on Weibo and Twitter datasets.

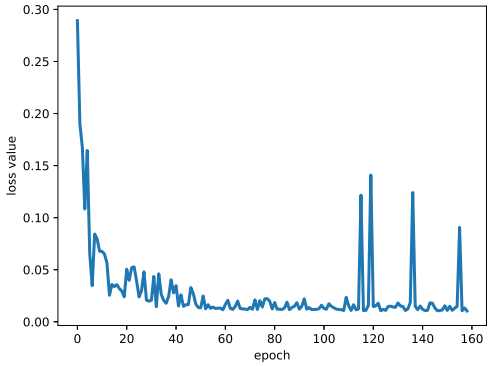


Fig. 4. The loss of Discriminator when the $\gamma = 10$, $\lambda = 1$.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel adversarial NER model via POS label embedding. **ANP** outperforms current Bi-LSTM+CRF-based state-of-the-art models on Twitter dataset for NER task and matches the state-of-the-art performance on the CoNLL-2003 and Weibo datasets. Our analysis shows that utilizing the *self-attention* mechanism to capture part-of-speech dependencies and *adversarial training* to map *tasks-shared* features can improve NER system performance significantly. Also, the regularization item and noun constraint have positive impacts on the performance of our model.

In the future, we will consider other sequence labeling tasks (i.e., Chunk, Semantic Role Label) and explore their relationships. We will also extend the architecture for other NLP tasks, such as relationship extraction and event extraction.

ACKNOWLEDGMENTS

The research work is supported by Natural Science Foundation of China (No. 61603197, 61772284, 61876091), and Graduate research and innovation plan project for the learning in Jiangsu province (No. SJKY19_0766).

REFERENCES

[1] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[2] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Computational Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[4] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[5] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[6] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016, pp. 1064–1074.

[7] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.

[8] Y. Shao, C. Hardmeier, and J. Nivre, "Multilingual named entity recognition using hybrid neural networks," in *The Sixth Swedish Language Technology Conference (SLTC)*, 2016.

[9] V. Yadav, R. Sharp, and S. Bethard, "Deep affix features improve neural named entity recognizers," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018, pp. 167–172.

[10] G. Aguilar, A. P. L. Monroy, F. González, and T. Solorio, "Modeling noisiness to recognize named entities using multitask neural networks on social media," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1401–1412.

[11] P. Le-Hong, M. P. Q. Nhat, T.-H. Pham, T.-A. Tran, and D.-M. Nguyen, "An empirical study of discriminative sequence labeling models for vietnamese text processing," in *Knowledge and Systems Engineering (KSE), 2017 9th International Conference on*. IEEE, 2017, pp. 88–93.

[12] G. Petasis, A. Cucchiarelli, P. Velardi, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Automatic adaptation of proper noun dictionaries through cooperation of machine learning and probabilistic methods," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 128–135.

[13] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2321–2331.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[15] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 343–351.

[16] K. Takeuchi and N. Collier, "Use of support vector machines in extended named entity recognition," in *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, 2002, pp. 1–7.

[17] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *stat*, vol. 1050, p. 6, 2017.

[18] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Adversarial training for multi-context joint entity and relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 2830–2836.

[19] M. Yasunaga, J. Kasai, and D. Radev, "Robust multilingual part-of-speech tagging via adversarial training," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, 2018, pp. 976–986.

[20] P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, "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*, 2018, pp. 182–192.

[21] X. Wang, X. Han, Y. Lin, Z. Liu, and M. Sun, "Adversarial multi-lingual neural relation extraction," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1156–1166.

[22] P. Qin, X. Weiran, and W. Y. Wang, "Dsgan: Generative adversarial training for distant supervision relation extraction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 496–505.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[24] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, "Deep semantic role labeling with self-attention," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum, "Linguistically-informed self-attention for semantic role labeling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 5027–5038.

[26] J. Du, J. Han, A. Way, and D. Wan, "Multi-level structured self-attentions for distantly supervised relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2216–2225.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.

[29] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[30] N. Peng and M. Dredze, "Improving named entity recognition for chinese social media with word segmentation representation learning," in *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016, p. 149.

[31] H. He and X. Sun, "A unified model for cross-domain and semi-supervised named entity recognition in chinese social media," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[32] A. Ritter, S. Clark, O. Etzioni *et al.*, "Named entity recognition in tweets: An experimental study," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1524–1534.

[33] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," *arXiv preprint arXiv:1703.06345*, 2017.

[34] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2013, pp. 380–390.

[35] S. Changpinyo, H. Hu, and F. Sha, "Multi-task learning for sequence tagging: An empirical study," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2965–2977.

[36] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of NAACL-HLT*, 2016, pp. 260–270.