Improving Chinese Named Entity Recognition with Multi-grained Words and Part-of-Speech Tags via Joint Modeling

Chenhui Dou[1], **Chen Gong**[1]*, **Zhenghua Li**[1], **Zhefeng Wang**[2], **Baoxing Huai**[2], **Min Zhang**[1]
[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, China, [2]Huawei Cloud, China
[1]20215227026@stu.suda.edu.cn [1]{gongchen18, zhli13, minzhang}@suda.edu.cn
[2]{wangzhefeng, huaibaoxing}@huawei.com

Abstract

Nowadays, character-based sequence labeling becomes the mainstream Chinese named entity recognition (CNER) approach, instead of word-based methods, since the latter degrades performance due to propagation of word segmentation (WS) errors. To make use of WS information, previous studies usually learn CNER and WS simultaneously with multi-task learning (MTL) framework, or treat WS information as extra guide features for CNER model, in which the utilization of WS information is indirect and shallow. In light of the complementary information inside multi-grained words, and the close connection between named entities and part-of-speech (POS) tags, this work proposes a tree parsing approach for joint modeling CNER, multi-grained word segmentation (MWS) and POS tagging tasks simultaneously. Specifically, we first propose a unified tree representation for MWS, POS tagging, and CNER.Then, we automatically construct the MWS-POS-NER data based on the unified tree representation for model training. Finally, we present a two-stage joint tree parsing framework. Experimental results on OntoNotes4 and OntoNotes5 show that our proposed approach of jointly modeling CNER with MWS and POS tagging achieves better or comparable performance with latest methods.

Keywords: Chinese NER, Multi-grained word segmentation, POS tagging, Joint model

1. Introduction

Named entity recognition (NER) aims to identify named entities (NE) from raw texts and classify them into pre-defined categories. As a fundamental task in natural language processing (NLP), NER is indispensable for many downstream NLP tasks, including question answering (Zhang et al., 2023), relation extraction (Zhao et al., 2023), and information retrieval (Hu et al., 2023).

In alphabetical languages such as English where words are explicitly separated with spaces, NER can be formulated as a word-based sequence labeling problem, i.e., treating words as the basic processing units. In contrast, Chinese adopts a logographic writing system without delimiters between words, while word information is essential for Chinese NER (CNER) due to the rich boundary information and basic semantic knowledge contained in words.

In order to leverage word information for CNER, a common way in early works, especially before the deep learning (DL) era, is first performing word segmentation (WS), and then recognizing named entities based on the predicted word sequence (Wu et al., 2012). However, these word-based pipeline approaches have the limitation that the inevitable segmentation errors in WS process can be further propagated to CNER, severely affecting CNER performance.

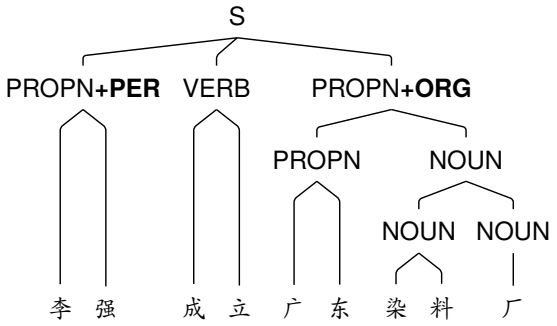

Figure 1: An example sentence of its Chinese MWS-POS-NER tree: "李强 (Li Qiang) 成立 (sets up) 广 东 (Kwangtung) 染 料 (Dyestuff) 厂 (Plant)." The bold fonts are NE labels.

Considering the error propagation issues, there have been few works on word-based approaches for CNER in the DL era. Using character-based (char-based) CNER models as backbones and further integrate word segmentation information into char-based models has been attractive for CNER. Recent methods of leveraging word segmentation information fall into two categories. The first kind of methods is to enhance CNER with implicit task-shared features by simultaneously training CNER with WS, or WS&POS tasks under the multi-task learning (MTL) framework with a task-shared encoder (Wu et al., 2019; Yan et al., 2023). The second one is explicitly integrating word segmenta-

* Corresponding author.

LREC-COLING 2024, pages 8732–8742

tion information as guide features into char-based CNER(He and Sun, 2017; Zhu and Wang, 2019). Although above methods have achieved great improvements, they still face the following two challenges.

On the one hand, the integration of CNER and word segmentation information through above guide-feature or MTL methods is indirect and shallow. On the other hand, existing methods usually only consider single-grained word segmentation (SWS), i.e., a sentence is split into a single word sequence, which ignore the fact that the segmentation granularities for Chinese word segmentation (CWS) are diverse from different linguistic perspectives. Compared with the word information obtained from SWS, words of different granularities can provide richer boundary information for CNER. For example, we examine the entities coverage ratio of OntoNotes4 (Weischedel et al., 2011) in SWS and multi-grained words (MWS) following Gong et al. (2020) respectively, and find that MWS has a much higher entities coverage ratio of 91.45% than that in SWS (85.54%).

With the above considerations, in this work, we propose to enhance the performance of CNER by representing and modeling words of different granularities and CNER in a unified structure, which can integrate rich multi-grained word information with CNER more closely and straightforward. Figure 1 shows an example of the tree-structure representation, where the non-terminals correspond to all words of different granularities, including named entities.

Moreover, NER is also closely correlated with POS tagging since named entities usually have the POS tags of proper noun. In fact, as aforementioned, researchers have tried to leverage POS tags for NER by taking POS tags as extra features (Dang et al., 2018; Nie et al., 2020).

In this work, we naturally integrate POS tags in our unified tree structure as non-terminal labels. As shown in Figure 1, each non-terminal is assigned with a POS tag. For the named entities, we further attach extra NE labels on them. For example, "李强" is a proper noun and also a person entity, so we assign the "PROPN" POS tag and "PER" NE label to it independently.

We conduct extensive experiments on two widely-used NER datasets, i.e., OntoNotes4 and OntoNotes5 (Pradhan et al., 2013), to verify the effectiveness of our method. Detailed analysis are also conducted to gain more insights on our proposed approach. We have released our codes at https://github.com/Huangmang3/JointNER.

The main contributions of this paper can be summarized as follows:

- We propose a unified representation for MWS, POS and CNER by capturing them in a single tree structure, and automatically construct the corresponding MWS-POS-NER data based on the tree representation, in order to take full advantage of word information for CNER.
- We propose to jointly model MWS-POS-NER with a two-stage parsing approach, which first parse MWS tree with POS tags then classify NE labels of predicted proper nouns, to improve CNER with the interactive knowledge of multi-grained words and POS tags.
- Extensive experiments and in-depth analysis verify the effectiveness of our proposed method of improving CNER with MWS and POS via joint modeling.

2. Related Work

Typically, there are two mainstream methods for NER. The first one is sequence labeling method (Hu et al., 2022b; Zhou et al., 2022; Zheng et al., 2022) which classifies each character or word in the sequence into an NE label, and is the most widely-used method in flat NER. The second one is span-based method (Wan et al., 2022; Zhu and Li, 2022; Lou et al., 2022) which performs classification on all the possible spans of the sentence to identify whether a span is an entity with pre-defined NE types, and is usually more preferred by nested NER. In this work, we adopt the span-based method. The main reason is that the span-based method can naturally model our proposed MWS-POS-NER tree structure, since a tree can be decomposed into its corresponding constituent spans.

For CNER, considering the rich boundary information and basic semantic knowledge contained in words, researchers usually incorporate word information to CNER models for better performance. In early traditional machine learning era, word-based sequence labeling methods are usually used for CNER (Wu et al., 2012), which first segment the sentence into a word sequence, and then classify each word into an NE label to identify entities. However, the word-based method can cause error propagation problem. Therefore, current works turn to incorporate word information into char-based CNER models with lexicon-enhanced method and joint modeling method, in addition to word segmentation guide-feature method and MTL method as discussed in Section1.

Lexicon-enhanced Model. In recent years, many researchers propose lexicon-enhanced CNER models to leverage word information. The main idea is first matching words in the sentence according to lexicons, and then integrating the matched words into char-based CNER models with various model architectures(Li et al.,

2020; Liu et al., 2021; Wu et al., 2021; Hu et al., 2022a). Although great improvements are achieved by lexicon-enhanced models, the helpful context-aware word information are ignored in lexicon, and it is impossible for lexicon to cover all the words in datasets. Thus, the word information obtained from lexicon is limited.

Joint Model. To utilize context-aware word information, Wang et al. (2019) propose to improve CNER by jointly representing and modeling CNER, SWS, POS tagging and constituent parsing. In their work, they reformulate parsing to height-limited constituent parsing by cutting nodes which exceed a certain height limit in the constituency tree and thus only reserving constituency subtrees for each sentence. They represent joint SWS-POS-CNER-Parsing in unified height-limited constituent parsing subtrees structure, where the NE labels and POS tags are merged into the same label space. Their approach have improved CNER performance. However, one limitation of their work is that they only focus on leveraging words of single granularity, without considering the fact that word segmentation granularities can be diverse from different linguistic perspectives. Moreover, they merge POS tags and NE labels into the same label space, ignoring the incompatibility and unbalance of these two label sets.

In this work, we propose to improve CNER with multi-grained words and POS tags by joint modeling. The differences between our work and Wang et al. (2019) are three-fold. First, compared with their work which only consider word information from single-grained segmented words, we propose to utilize words of multiple granularities, in order to gain richer word information from different linguistic perspectives for CNER. Second, we represent joint MWS-POS-NER as a unified tree structure, rather than subtrees structure. Third, we propose a two-stage approach to handle POS tags and NE labels separately, instead of merging them into the same label space.

3. Representing MWS-POS-NER as Unified Char-level Tree

The key idea of this work is to jointly represent and model MWS, POS tagging and NER, in order to improve NER performance by leveraging the shared and interactive features of multi-grained words and POS tags.

In this section, we first describe the unified MWS-POS-NER tree-structure representation, and then introduce how to automatically produce MWS-POS-NER data on OntoNotes4 (Weischedel et al., 2011) and OntoNotes5 (Pradhan et al., 2013) datasets.

3.1. Unified MWS-POS-NER Tree Structure

We propose to represent MWS, POS, and NER in a unified manner by constructing a MWS-POS-NER tree structure, for the purpose of taking advantage of rich word information for NER by closely integrating NER with MWS and POS.

As illustrated in the bottom part of Figure 2, all words of different granularities along with their corresponding POS tags or NE labels are encoded in the hierarchical tree structure. In the tree, each leaf node is a character. When a single character or several consecutive characters composing a word, a non-terminal node is created with a POS tag attached. For example, "最" is a single-char word with a POS tag of "ADV" (short for adverb); whereas "最高" is a multi-char word with a POS tag of "ADJ" (short for adjective). The two words correspond to two non-terminal nodes in the tree with POS tags as the node labels.

For entities, we append an NE label to the POS tag if a word is also a named entity. For example, the word "最高人民检察院政治部" is an organization and its corresponding node is labeled as "PROPN+ORG", in which "PROPN" (short for proper noun) is the POS tag, and "ORG" is the NE label. Please note that the parsing model handles POS tags and NE labels separately, as discussed in Section 4.1.

3.2. Constructing MWS-POS-NER Data

We propose a two-step method to automatically construct a unified MWS-POS-NER tree for each sentence in the training data of OntoNotes4 and OntoNotes5 by additionally making use of two existing widely-used heterogeneous datasets, i.e., the People Daily Corpus of the Peking University (PPD)(Yu, 2003) and the Microsoft Research Word Segmentation Corpus (MSR) (Huang et al., 2006). Among the four above mentioned datasets, OntoNotes4 and OntoNotes5[1] are two popular NER datasets which contain annotated named entities, CWS, and POS tags simultaneously, PPD is annotated with both WS labels and POS tags, and MSR is only annotated with WS labels. Moreover, OntoNotes, PPD, and MSR use heterogeneous annotation guidelines.

First, we automatically obtain MWS-POS tree structure for each sentence in OntoNotes by utilizing the heterogeneous SWS-POS annotations in OntoNotes and PPD, and the SWS annotations in

[1]Since the workflow of constructing MWS-POS-NER trees for OntoNotes4 and OntoNotes5 are similar, we refer to OntoNotes4 and OntoNotes5 as OntoNotes later in this section.

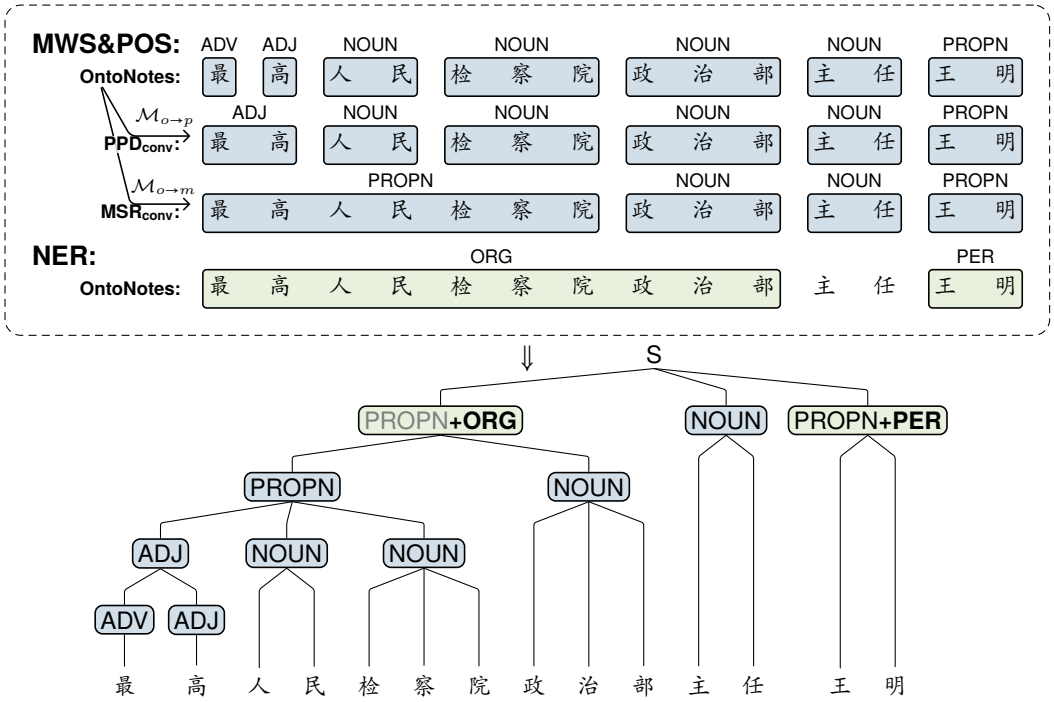

Figure 2: An example sentence of how its Chinese MWS-POS-NER tree is generated by different SWS&POS results and NE labels: "最高 (the Supreme) 人民 (People's) 检察院 (Procuratorate) 政治部 (Political Department) 主任 (director) 王明 (Wang Ming)."

MSR, with the approach of annotation conversion[2] (Gong et al., 2022). Second, we attach NE labels on the MWS-POS tree according to the NE annotations in OntoNotes to form the unified MWS-POS-NER tree for each sentence. The whole workflow is shown in Figure 2.

Step 1: Generating MWS tree with POS tags. In this work, we first generate the hierarchical tree structure of multi-grained words with POS tags on OntoNotes by additionally leveraging PPD and MSR. Since OntoNotes mostly has non-overlapping and heterogeneous annotations with PPD and MSR, and each of these datasets only contains single-side gold labels. We automatically convert the OntoNotes-side gold labels into the PPD-side labels (denoted as PPD_{conv} in Figure 2), and convert the OntoNotes-side gold labels into the MSR-side labels (denoted as MSR_{conv} in Figure 2) respectively with the annotation conversion approach of Gong et al. (2022), in order to obtain the labels according to three different guidelines for each sentence simultaneously.

The upper part of Figure 2 illustrates an example of the above annotation conversion workflow: each sentence in OntoNotes is converted into its corresponding sentence with PPD-side labels via $\mathcal{M}_{o\to p}$ and MSR-side labels via $\mathcal{M}_{o\to m}$, respectively. Please kindly note that we map POS tags in different datasets into Universal Dependencies (UD)[3] according to pre-defined mapping rules, to unify the representation of POS tags under different annotation guidelines. For MSR which has no POS annotations, we produce its POS tags based on the following rules: if a word in MSR also exists in OntoNotes or PPD, it will be assigned with the same POS tag as OntoNotes or PPD, otherwise, its POS tag will be determined according to pre-defined production rules.

Finally, for each sentence in OntoNotes, we can produce the WS and POS results under three different guidelines. There may be situations where the WS results under different guidelines overlap with each other. For example, in MSR_{conv}, the segmentation result for the word "ABC" is "AB/C", but in PPD_{conv}, the segmentation result is "A/BC". According to our preliminary experimental statistics, we found that less than 0.1% of segmented words overlap with other segmented words. For simplicity, we directly adopt the predicted WS results with high confidence (e.g. AB/C) and discard other WS results (e.g. A/BC). Finally, we represent the produced WS and POS results in a MWS-POS tree structure as shown in the bottom part of Figure 2.

[2]We have re-implemented the code and released at https://github.com/SudaLaGroup/CoupledModelwithBERT.

[3]universaldependencies.org/u/pos/

Step 2: Attaching NE labels to MWS-POS tree. In step 2, we attach the manually annotated NE labels in OntoNotes training data to the MWS-POS tree obtained in step 1, in order to form the complete MWS-POS-NER tree for each sentence in OntoNotes training data.

For the entity which is a word in the MWS-POS tree (accounting for 96% of all the named entities), we directly attach an extra NE label to its corresponding word non-terminal to indicate the word is also an entity. Considering that most of the entities are proper nouns, we define the POS tags of all the entities to be "PROPN" for simplicity. For example, in Figure 2, "王明" has two separate labels of POS tag "PROPN" and NE label "PER" in the non-terminal, meaning that the word "王明" is a person entity with the POS tag of proper noun.

For the entity which is annotated in the original OntoNotes datasets but is not considered as a word in the MWS-POS tree obtained by step 1, we represent it by adding a new non-terminal node for the corresponding entity to the tree. For example, the organization entity "最高人民检察院政治部" is an annotated named entity in OntoNotes but is not a word in MWS-POS tree. To form the complete MWS-POS-NER tree, we add a new non-terminal to the original MWS-POS tree with the NE label "ORG" and the POS tag "PROPN".

Please kindly note that we only construct MWS-POS-NER tree via the above proposed two-step method for each sentence in training data. For dev and test data, we do not provide any gold MWS or POS information, instead, the MWS-POS-NER tree is automatically predicted by our proposed joint model.

4. Joint MWS-POS-NER Parsing

Based on the tree representation that we build in the previous section, as shown in Figure 1, we naturally employ a two-stage tree parsing model to cast the joint modeling of MWS, POS tagging and CNER.

4.1. The Two-stage Parsing Framework

As discussed in Section 3.2, our tree representation allows entities to be only located tagged "PROPN" (i.e., direct children of the root node "S"). It means that entities only appear in a small fraction of tree nodes. In light of that, we propose a two-stage MWS-POS-NER parsing framework, as shown in Figure 3. In the first stage, the joint model predicts a MWS tree with POS tags. In the second stage, the model determines whether the words with "PROPN" tags predicted in the first stage are entities and identifies their NE labels.

Formally, given a character-level sentence $\boldsymbol{x} = c_1, c_2, ..., c_n$, we use (i, j, t) to represent that $c_i...c_j$ corresponds to a word tagged as t in the first stage, and use (i, j, l) to represent that $c_i...c_j$ is an entity labeled as l in the second stage.

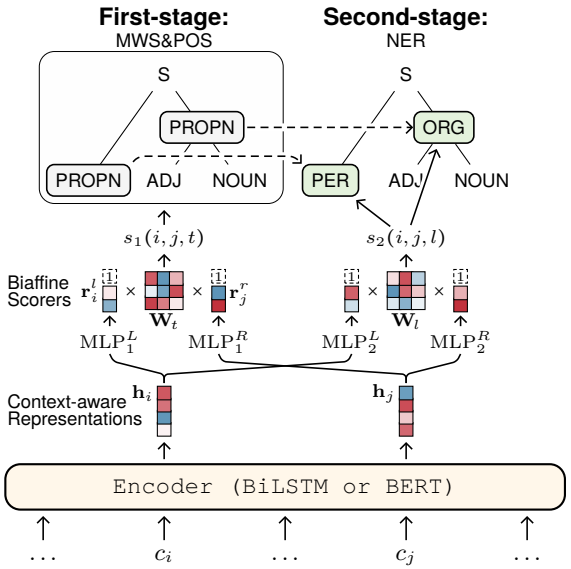

Figure 3: The architecture of the two-stage joint parsing framework.

First-stage: Predicting MWS tree with POS tags. As illustrated by the tree in the upper part of Figure 3, in the first stage, the model aims to produce an MWS tree with POS tags.

Formally, given a sentence $\boldsymbol{x}$, the goal is to find an MWS-POS tree $\hat{\boldsymbol{y}}$ with the highest score. The score of the tree $\boldsymbol{y}$ is calculated by summing up the scores of all its constituent MWS-POS spans.

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(i,j,t) \in \boldsymbol{y}} s(i, j, t)$$
$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} s(\boldsymbol{x}, \boldsymbol{y}) \quad (1)$$

Second-stage: Recognizing named entities. As shown in the upper part of Figure 3, after obtaining an optimal tree $\hat{\boldsymbol{y}}$ that contains all the multi-grained words and their corresponding POS tags, in the second stage, we constrain our model to only recognize NE labels $\hat{l}$ for those words with "PROPN" tags in the predicted tree $\hat{\boldsymbol{y}}$, since all the named entities have the POS tag of "PROPN" as illustrated in Section 3.2.

$$\hat{l} = \arg\max_{l \in \mathcal{N}} s(i, j, l) \quad (2)$$

where $\mathcal{N}$ is the NE label set. Please note that we add an extra label of "$\varnothing$" in the label set to indicate not an entity.

4.2. The Neural Model Architecture

This subsection mainly introduces the details of the model calculating span scores in two stages, which is shown in the lower part of the Figure 3.

Inputs. For a given sentence, we use character embedding $\mathbf{e}_i$ to represent the i-th character c_i.

$$\mathbf{e}_i = \mathbf{emb}(c_i) \tag{3}$$

Encoder. We apply two types of encoders, i.e., 1) a three-layer BiLSTM or 2) BERT, to encode the sentence and obtain context-aware representations $\mathbf{h}_i$ of c_i, where $\mathbf{h}_i$ is the top-layer outputs of BiLSTM or BERT.

Boundary representation. We use two separate MLPs to acquire the left and right boundary representation vectors of c_i.

$$\mathbf{r}_i^L; \mathbf{r}_i^R = \mathbf{MLP}^L(\mathbf{h}_i); \mathbf{MLP}^R(\mathbf{h}_i) \tag{4}$$

where $\mathbf{r}_i^L/\mathbf{r}_i^R$ is the boundary representation in the situation that c_i is the left/right boundary of a span.

Biaffine Scorer. The score $s(i,j,t)$ of each MWS span $c_i...c_j$ with a POS tag t in the first stage is computed via a biaffine operation (Dozat and Manning, 2017) over $\mathbf{r}_i^L$ and $\mathbf{r}_j^R$.

$$s(i,j,t) = \begin{bmatrix} \mathbf{r}_i^L \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}_t \begin{bmatrix} \mathbf{r}_j^R \\ 1 \end{bmatrix} \tag{5}$$

where $\mathbf{W}_t \in \mathbb{R}^{501\times501}$ is the biaffine parameter.

Similarly, for the score $s(i,j,l)$ of each entity span in the second stage, two extra MLPs are used to obtain boundary representations, and an extra biaffine operation is performed to calculate $s(i,j,l)$.

4.3. Training Loss

For the first stage, we apply a label aware TreeCRF loss $\mathcal{L}^{1st}$ to maximize the conditional probability $p(\boldsymbol{y}^*|\boldsymbol{x})$ of the gold MWS-POS tree $\boldsymbol{y}^*$.

$$\begin{aligned} \mathcal{L}^{1st}(\boldsymbol{x}, \boldsymbol{y}^*) &= -\log p(\boldsymbol{y}^*|\boldsymbol{x}) \\ p(\boldsymbol{y}^*|\boldsymbol{x}) &= \frac{e^{s(\boldsymbol{x},\boldsymbol{y}^*)}}{\boldsymbol{Z}(\boldsymbol{x}) \equiv \sum_{\boldsymbol{y}' \in \mathcal{T}(\boldsymbol{x})} e^{s(\boldsymbol{x},\boldsymbol{y}')}} \end{aligned} \tag{6}$$

where $\boldsymbol{Z}(\boldsymbol{x})$ is the normalization term and $\mathcal{T}(\boldsymbol{x})$ is the set of all possible trees.

For the second stage, we compute a cross-entropy loss for each span (i,j,l) in $\boldsymbol{z}^*$, where $\boldsymbol{z}^*$ is the gold-standard tree with NE labels (including "$\varnothing$") , and accumulate them as the loss $\mathcal{L}^{2nd}$ of the second stage.

$$\mathcal{L}^{2nd}(\boldsymbol{x}, \boldsymbol{z}^*) = \sum_{(i,j,l)\in \boldsymbol{z}^*} -\log \frac{e^{s(i,j,l)}}{\sum_{l'} e^{s(i,j,l')}} \tag{7}$$

Finally, the training loss of the two stages are added as the total loss of our joint model.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^*, \boldsymbol{z}^*) = \mathcal{L}^{1st}(\boldsymbol{x}, \boldsymbol{y}^*) + \mathcal{L}^{2nd}(\boldsymbol{x}, \boldsymbol{z}^*) \tag{8}$$

Datasets	Type	Train	Dev	Test
OntoNotes4	#Sent.	15,724	4,301	4,346
	#Entity	13,372	6,950	7,684
OntoNotes5	#Sent.	36,487	6,083	4,472
	#Entity	62,543	9,104	7,494

Table 1: Numbers of sentences and entities in OntoNotes4 and OntoNotes5 datasets.

5. Experiments

Data. We conduct experiments on two NER datasets, i.e., OntoNotes4 and OntoNotes5. We choose these two datasets becaouse they are two widely-used datasets that contain CNER and parallel annotations of CWS and POS tagging. Table 1 shows the data statistics. OntoNotes4 is mainly from newswire domain and has 4 entity types. OntoNotes5 also contains texts mainly from newswire domain, but is in larger scale compared with OntoNotes4 and has 18 entity types. We use the same data split as Zhang and Yang (2018) for OntoNotes4, and the same data split as Jie and Lu (2019) for OntoNotes5.

In order to enhance CNER performance by joint learning with MWS and POS tagging, for the training data, we automatically obtain extra MWS and POS tags for each sentence to form a unified MWS-POS-NER tree for model training, by leveraging two heterogeneous datasets (i.e., PPD and MSR), as illustrated in Section 3.2.

Evaluation metrics. We employ the standard precision (P), recall (R), and F1 score for evaluating NER performance.

Model details. In order to perform CKY decoding, we adopt left binarization and transform the original tree into those of Chomsky normal form (CNF) via the NLTK[4] tool. The hyper-parameter settings is same to the constituency parser of Zhang et al. (2020). The character embeddings are pre-trained on Chinese Giga-Word (Li et al., 2019), whose dimension is 100. The training process will be stopped if the peak performance on dev data does not increase in 100 consecutive epochs on BiLSTM models. For experiments with BERT, we adopt "BERT-wwm (Cui et al., 2021)" to fine-tune and the training process continues 25 epochs. We run each model for three times with different random seeds and report the average result.

5.1. Results on Dev Data

To understand the influence of integrating word information into CNER via joint modeling with MWS and POS tagging, we compare the development

[4]https://www.nltk.org

Model	OntoNotes4-Dev			OntoNotes5-Dev			Sent/s
	P	R	F1	P	R	F1	
Char-based	72.77	64.64	$68.42_{\pm 0.25}$	72.90	69.53	$71.17_{\pm 0.11}$	393
Joint model	75.86	66.21	$\mathbf{70.70}_{\pm 0.29}$	77.50	71.22	$\mathbf{74.22}_{\pm 0.03}$	349
Char-based w/ lexicon	74.63	72.72	$73.65_{\pm 0.19}$	74.25	74.39	$74.32_{\pm 0.20}$	136
Joint model w/ lexicon	76.07	72.37	$\mathbf{74.18}_{\pm 0.12}$	78.83	73.59	$\mathbf{76.12}_{\pm 0.12}$	131
Char-based w/ BERT	78.96	80.18	$79.55_{\pm 0.11}$	75.86	78.19	$77.01_{\pm 0.07}$	204
Joint model w/ BERT	80.39	80.44	$\mathbf{80.41}_{\pm 0.21}$	78.83	77.41	$\mathbf{78.09}_{\pm 0.16}$	179

Table 2: Development results on OntoNotes4/5.

results under various model configurations. Table 2 shows the results.

The first major row compares the CNER performance of character-based BiLSTM-CRF sequence labeling CNER model (denoted as "Char-based") and the joint MWS-POS-NER tree parsing model without using lexicon information or BERT (denoted as "Joint model"). We can see that the proposed "Joint model" greatly outperforms the "Char-based" by more than 2 in F1 on both datasets. It indicates that joint learning CNER with MWS and POS tagging can bring advantage to CNER performance by sharing the rich word boundary information and POS information.

The second major row shows the results of "Char-based" and "Joint model" enhanced with lexicon. Compared with the models in the first major row, we further integrate additional lexicon words into the models by matching the sentence with an automatically-obtained lexicon and concatenate the lexicon representations with the original input character embeddings following WC-LSTM (Liu et al., 2019). After incorporating lexicon words, our "Joint model w/ lexicon" can still achieve better performance than "Char-based w/ lexicon" on both two datasets, demonstrating that the shared morphology knowledge in our joint model is complementary with that in lexicon, and both make contributions to CNER model.

The third major row reports the results with BERT encoder, which are dramatically improved compared with the results in the first two major rows. Based on BERT encoder, our "Joint model w/ BERT" outperforms "Char-based w/ BERT" by about 1 in F1 on both datasets. It shows that our method of leveraging word information via joint modeling with MWS and POS tagging can complement the contextualized information contained in BERT, further verifying the effectiveness of our method.

We also compare the parsing speed of different models to investigate whether our joint models achieve better performance at the cost of very low efficiency. In the last colum of Table 2, we report the average number of sentences parsed by different models in OntoNotes4. For fair comparison, we run each model with a single Nvidia GTX 1080Ti GPU on the same machine. We can see that the efficiency of our joint models are only slightly inferior than that of char-based models under all the three settings [5]. The reason is that we follow Zhang et al. (2020) to employ the batchified inside algorithm to perform parallel operation and thus can fully utilize the power of GPUs to gain efficiency.

Overall, we can conclude that incorporating word boundary information and POS information via joint modeling can help improve the performance of CNER consistently, without much hurt in efficiency. Considering that the model based on BERT encoder performs the best on dev data, our subsequent experiments and analyses are conducted on the model based on BERT encoder.

5.2. Comparison with Previous Works

Table 3 compares the results of our model with previous works on OntoNotes4 and OntoNotes5 test data. In Table 3, for the OntoNotes5 results of WC-LSTM, FLAT, SoftLexicon, LEBERT and W^{2}NER, we re-run the codes released by corresponding works. After comparison, we can see that our "Joint model†" achieves better performance compared with most of the previous approaches on OntoNotes4 and OntoNotes5, showing that our MWS-POS-NER joint model is effective in improving CNER performance by making full use of word information.

The result of our joint model on OntoNotes4 is inferior to that of ATSSA, ACT-S, and W ^{2}NER. The reason is that ACT-S introduces additional bilingual information, and thus its result is much higher than other models. ATSSA uses Adaptive Threshold Selective Self Attention to replace the Self-Attention module in FLAT model, which makes the model can focus on more critical keys and obtain more effective information. W^{2}NER uses two sets of

[5]The speed of models under the setting of "w/ lexicon" are inferior than that of other two settings due to the computational cost of handling additional lexicon information.

Model	F1
OntoNotes4	
Lattice LSTM (Zhang and Yang, 2018)	73.88
LR-CNN (Gui et al., 2019)	74.45
WC-LSTM (Liu et al., 2019)	74.43
PLTE† (Xue et al., 2020)	80.60
FLAT† (Li et al., 2020)	81.82
SoftLexicon† (Ma et al., 2020)	82.81
LEBERT† (Liu et al., 2021)	82.08
MECT† (Wu et al., 2021)	82.57
ATSSA† (Hu et al., 2022a)	83.31
ACT-S† (Ning et al., 2022)	**83.91**
W²NER† (Li et al., 2022)	83.08
Joint model†	82.82
OntoNotes5	
WC-LSTM (Liu et al., 2019)	75.95
DGLSTM-CRF (Jie and Lu, 2019)	77.40
FLAT† (Li et al., 2020)	77.87
SoftLexicon† (Ma et al., 2020)	79.71
LEBERT† (Liu et al., 2021)	78.30
W²NER† (Li et al., 2022)	79.04
Joint model†	**79.87**

Table 3: Comparison with previous works on OntoNotes4/5 test data. '†' indicates that the model uses BERT.

relationships to unified model named entities with three different representations: flat, nested, and discontinuous. These methods are orthogonal to our work, and we can also attempt to use these methods to further improve the performance of our joint model.

5.3. Analysis

We conduct detailed analysis to better understand the NER improvements introduced by our MWS-POS-NER joint model. In Table 4, we present the test results on the settings of using pre-trained BERT encoder.

Pipeline vs. Joint framework in using WS. We compare the results of integrating WS information under the pipeline and the joint framework. The "Word-based (orig.)" row shows the result of the pipeline framework, which first obtains segmented words[6] according to the original SWS annotations in OntoNotes4/5 and then take the words as the input of NER model. In the "+SWS (orig.)" row, we learn NER and WS simultaneously under the joint framework by parsing on SWS-NER structure, which is a simplified version of the MWS-POS-NER

[6]gold words for train, and automatic predicted words for dev/test, the same for "+SWS (orig.)"

Model	OntoNotes4	OntoNotes5
NER as sequence labeling		
Char-based	$81.70_{\pm 0.28}$	$78.30_{\pm 0.16}$
Word-based (orig.)	$79.28_{\pm 0.17}$	$78.14_{\pm 0.11}$
Joint NER w/ WS as tree parsing		
+SWS (orig.)	$81.82_{\pm 0.17}$	$79.34_{\pm 0.39}$
+SWS (fine)	$81.96_{\pm 0.32}$	$79.29_{\pm 0.29}$
+SWS (coarse)	$82.04_{\pm 0.23}$	$79.50_{\pm 0.05}$
+MWS	$\mathbf{82.11}_{\pm 0.16}$	$\mathbf{79.58}_{\pm 0.20}$
Joint NER w/ WS&POS as tree parsing		
+SWS (orig.)&POS	$82.20_{\pm 0.05}$	$79.69_{\pm 0.14}$
+SWS (fine)&POS	$81.97_{\pm 0.19}$	$79.64_{\pm 0.25}$
+SWS (coarse)&POS	$82.43_{\pm 0.24}$	$79.84_{\pm 0.41}$
+MWS&POS	$\mathbf{82.82}_{\pm 0.07}$	$\mathbf{79.87}_{\pm 0.20}$
w/o PROPN constraint	$82.55_{\pm 0.06}$	$79.82_{\pm 0.12}$
merge POS&NE label	$81.91_{\pm 0.58}$	$79.52_{\pm 0.29}$

Table 4: Ablation studies on models with BERT on OntoNotes4/5 test data.

tree in Figure 1 by reserving only single-grained word spans obtained from the original SWS annotations in OntoNotes4/5 and omitting POS tags. By comparing "Word-based (orig.)" and "+SWS (orig.)" results, we observe that integrating WS with pipeline framework is distinctly inferior to that using joint framework, even inferior to the result of "Char-based". It indicates that the pipeline framework severely suffer from the error propagation issue, while the joint framework can alleviate the issue.

Impact of word granularities. To analyze the effectiveness of introducing words of multiple granularities, we compare the results of joint learning NER with SWS and MWS on the settings of without POS tags (the second major row in Table 4) or with POS tags (the first four lines of the third major row in Table 4). The results show that the joint models integrated with MWS under both settings, i.e., "+MWS" and "+MWS&POS" in Table 4, achieve consistently better performance than that integrated with SWS, demonstrating that words of multiple granularities can provide more word information for CNER than single-grained words.

Moreover, to further analyze which granularity of the multi-grained words contributes to NER performance best, we look into the rows of "+SWS (fine)" and "+SWS (coarse)", which are results of only reserving the finest-grained words or the coarsest-grained words in the unified tree. We observe that integrating either of the two granularities via joint modeling can help improve the char-based NER, and coarse-grained words bring more improvements to NER performance possibly be-

cause coarse-grained words usually have the same boundaries as named entities, and thus can provide richer useful boundary information.

Overall, it can be concluded that words of all the different granularities in MWS can provide complementary contributions to CNER, and among which coarse-grained words make more contributions.

Impact of using POS tags. The third major row in Table 4 shows the results of further integrating POS tags into the joint model. First, comparing the first four lines of the third major row and the second major row, we can see that the integration of POS tags leads to consistent improvements for the performance of NER, since entities usually have the POS tags of "PROPN" and thus POS information can help better recognize named entities.

Second, in the "w/o PROPN constraint" row, we remove the constraint of only recognizing NE labels on the predicted "PROPN" words (i.e., recognize NE labels on all the predicted words) in the second stage of our model. The decreased performance is because after removing the PROPN constraint, the average precision of the two datasets decreased by about 1.3%, while the average recall increased by about 0.3%. The reason is that after removing the constraint that only predicting NE on the words with the predicted POS tag of PROPN, the joint model can predict NE label on all POS tags, resulting in an increased number of predicted NEs. Among the increased number of predicted NEs, although some of these increased NEs are predicted correctly, more additional predicted NEs are incorrectly predicted without PROPN constraint. The experimental results also verified the effectiveness of our strategy of PROPN constraint.

Finally, in order to measure the effectiveness of our method in distinguishing POS label space from NE label space via two-stage framework, we remove the second stage of our joint model, and merge POS and NE labels in the same label space to predict them simultaneously in one stage. The decline in "merge POS&NE label" performance and stability verifies the importance of a clear distinction for POS and NE label sets, showing the advantage of our method in avoiding the incompatibility and unbalance issues of the two label sets.

6. Conclusions

In this paper, we propose to jointly model CNER with MWS and POS tagging to promote the performance of CNER. We first propose a unified MWS-POS-NER tree-structure representation to represent each sentence. Based on the tree representation, we then naturally employ a two-stage parsing framework for joint modeling. Experiments and analysis on two widely-used NER datasets, i.e., OntoNotes4 and OntoNotes5, show that the proposed joint model can effectively improve CNER with the shared and interactive rich word information in multi-grained words and POS tags, achieving better or comparable performance compared with the previous approaches.

Acknowledgements

The authors would like to thank the anonymous reviewers for the helpful comments. We are very grateful to Houquan Zhou to discuss with us and provide valuable suggestions. This work is supported by the National Natural Science Foundation of China (Grant No. 62306202, 62176173 and 62176174), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No.23KJB520034), and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Bibliographical References

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Thanh Hai Dang, Hoang-Quynh Le, Trang M Nguyen, and Sinh T Vu. 2018. D3NER: biomedical named entity recognition using CRF-BiLSTM improved with fine-tuned embeddings of various linguistic information. *Bioinformatics*, 34(20):3539–3546.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Chen Gong, Zhenghua Li, and Min Zhang. 2022. Neural coupled sequence labeling for heterogeneous annotation conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1624–1636.

Chen Gong, Zhenghua Li, Bowei Zou, and Min Zhang. 2020. Multi-grained Chinese word segmentation with weakly labeled data. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2026–2036.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019. CNN-Based Chinese NER with lexicon rethinking. In *Proceedings of the Twenty-Eighth International*

Joint Conference on Artificial Intelligence, pages 4982–4988.

Hangfeng He and Xu Sun. 2017. F-score driven max margin neural network for named entity recognition in Chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 713–718.

Biao Hu, Zhen Huang, Minghao Hu, Ziwen Zhang, and Yong Dou. 2022a. Adaptive threshold selective self-attention for Chinese NER. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1823–1833.

Jinpeng Hu, Yaling Shen, Yang Liu, Xiang Wan, and Tsung-Hui Chang. 2022b. Hero-Gang neural model for named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1924–1936.

Xiyang Hu, Xinchi Chen, Peng Qi, Deguang Kong, Kunlun Liu, William Yang Wang, and Zhiheng Huang. 2023. Language agnostic multilingual information retrieval with contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9133–9146.

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

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuan-Jing Huang. 2020. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842.

Ying Li, Zhenghua Li, Min Zhang, Rui Wang, Sheng Li, and Luo Si. 2019. Self-attentive biaffine dependency parsing. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5067–5073.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. 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)*, pages 2379–2389.

Chao Lou, Songlin Yang, and Kewei Tu. 2022. Nested named entity recognition as latent lexicalized constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6183–6198.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuan-Jing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020. Improving named entity recognition with attentive ensemble of syntactic information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245.

Jinzhong Ning, Zhihao Yang, Zhizheng Wang, Yuanyuan Sun, Hongfei Lin, and Jian Wang. 2022. Two languages are better than one: Bilingual enhancement for Chinese named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2024–2033.

Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. 2022. Nested named entity recognition with span-level graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–903.

Rui Wang, Xin Xin, Wei Chang, Kun Ming, Biao Li, and Xin Fan. 2019. Chinese NER with height-limited constituent parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7160–7167.

Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2019. Neural Chinese named entity recognition via CNN-LSTM-CRF and joint training with word segmentation. In *The World Wide Web Conference*, pages 3342–3348.

Shuang Wu, Xiaoning Song, and Zhenhua Feng. 2021. MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1529–1539.

Zejian Wu, Zhengtao Yu, Jianyi Guo, Cunli Mao, and Youmin Zhang. 2012. Fusion of long distance dependency features for Chinese named entity recognition based on markov logic networks. In *Natural Language Processing and Chinese Computing*, pages 132–142.

Mengge Xue, Bowen Yu, Tingwen Liu, Yue Zhang, Erli Meng, and Bin Wang. 2020. Porous lattice transformer encoder for Chinese NER. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3831–3841.

Yibo Yan, Peng Zhu, Dawei Cheng, Fangzhou Yang, and Yifeng Luo. 2023. Adversarial multitask learning for efficient chinese named entity recognition. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(7):1–19.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023. FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1002–1017.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural CRF constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4046–4053.

Yue Zhang and Jie Yang. 2018. Chinese NER using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564.

Wenzheng Zhao, Yuanning Cui, and Wei Hu. 2023. Improving continual relation extraction by distinguishing analogous semantics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1162–1175.

Junhao Zheng, Zhanxian Liang, Haibin Chen, and Qianli Ma. 2022. Distilling causal effect from miscellaneous other-class for continual named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3615.

Renjie Zhou, Zhongyi Xie, Jian Wan, Jilin Zhang, Yong Liao, and Qiang Liu. 2022. Attention and edge-label guided graph convolutional networks for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6499–6510.

Enwei Zhu and Jinpeng Li. 2022. Boundary smoothing for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7096–7108.

Yuying Zhu and Guoxin Wang. 2019. CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition. 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)*, pages 3384–3393.

Language Resource References

Chang-Ning Huang, Yumei Li, and Xiaodan Zhu. 2006. Tokenization guidelines of Chinese text (V5. 0, in Chinese). *Microsoft Research Asia*.

Pradhan, Sameer and Moschitti, Alessandro and Xue, Nianwen and Ng, Hwee Tou and Björkelund, Anders and Uryupina, Olga and Zhang, Yuchen and Zhong, Zhi. 2013. *Towards Robust Linguistic Analysis using OntoNotes*. Association for Computational Linguistics, ISLRN 151-738-649-048-2.

Weischedel, Ralph and Pradhan, Sameer and Ramshaw, Lance and Palmer, Martha and Xue, Nianwen and Marcus, Mitchell and Taylor, Ann and Greenberg, Craig and Hovy, Eduard and Belvin, Robert and others. 2011. *OntoNotes release 4.0*. ISLRN 272-858-321-100-4.

Yu, Shiwen. 2003. *Specification for corpus processing at Peking University: Word segmentation, POS tagging and phonetic notation*.