# Improving Named Entity Recognition with Attentive Ensemble of Syntactic Information

**Yuyang Nie**$^{\diamond*}$, **Yuanhe Tian**$^{\heartsuit*}$, **Yan Song**$^{\spadesuit\heartsuit\dagger}$, **Xiang Ao**$^{\triangle\square}$, **Xiang Wan**$^{\heartsuit}$

$^{\diamond}$University of Electronic Science and Technology of China
$^{\heartsuit}$University of Washington   $^{\spadesuit}$The Chinese University of Hong Kong (Shenzhen)
$^{\heartsuit}$Shenzhen Research Institute of Big Data
$^{\triangle}$Institute of Computing Technology Chinese Academy of Sciences
$^{\square}$University of Chinese Academy of Sciences
$^{\diamond}$nyy207@gmail.com $^{\heartsuit}$yhtian@uw.edu
$^{\spadesuit}$songyan@cuhk.edu.cn $^{\triangle}$aoxiang@ict.ac.cn $^{\heartsuit}$wanxiang@sribd.cn

## Abstract

Named entity recognition (NER) is highly sensitive to sentential syntactic and semantic properties where entities may be extracted according to how they are used and placed in the running text. To model such properties, one could rely on existing resources to providing helpful knowledge to the NER task; some existing studies proved the effectiveness of doing so, and yet are limited in appropriately leveraging the knowledge such as distinguishing the important ones for particular context. In this paper, we improve NER by leveraging different types of syntactic information through attentive ensemble, which functionalizes by the proposed key-value memory networks, syntax attention, and the gate mechanism for encoding, weighting and aggregating such syntactic information, respectively. Experimental results on six English and Chinese benchmark datasets suggest the effectiveness of the proposed model and show that it outperforms previous studies on all experiment datasets.[1]

## 1 Introduction

Named entity recognition (NER) is one of the most important and fundamental tasks in natural language processing (NLP), which identifies named entities (NEs), such as locations, organizations, person names, etc., in running texts, and thus plays an important role in downstream NLP applications including question answering (Pang et al., 2019), semantic parsing (Dong and Lapata, 2018) and entity linking (Martins et al., 2019), etc.

The main methodology for NER is conventionally regarded as a sequence labeling task with models such as hidden Markov model (HMM) (Bikel et al., 1997) and conditional random field (CRF) (McCallum and Li, 2003) applied to it in previous studies. Recently, neural models play a dominate role in this task and illustrated promising results (Collobert et al., 2011; Huang et al., 2015; Lample et al., 2016; Strubell et al., 2017; Yadav and Bethard, 2018; Chen et al., 2019; Jie and Lu, 2019; Liu et al., 2019d; Baevski et al., 2019), because they are powerful in encoding contextual information and thus drive NER systems to better understand the text and recognize NEs in the input text. Although it is straightforward and effective to use neural models to help NER, it is expected to incorporate more useful features into an NER system. Among all such features, syntactic ones, such as part-of-speech (POS) labels, syntactic constituents, dependency relations, are of high importance to NER because they are effective in identifying the inherited structure in a piece of text and thus guide the system to find appropriate NEs accordingly, which is proved in a large body of previous studies (McCallum, 2003; Li et al., 2017; Luo et al., 2018; Dang et al., 2018; Jie and Lu, 2019). Although promising results are obtained, existing models are limited in regarding extra features as gold references and directly concatenate them with word embeddings. Therefore, such features are not distinguished and separately treated when they are used in those NER models, where the noise in the extra features (e.g., inaccurate POS tagging results) may hurt model performance. As a result, it is still a challenge to find an appropriate way to incorporate external information into neural models for NER. Moreover, in most cases, one would like to incorporate more than one types of extra features. Consequently, it is essential to design an effective mechanism to combine and weight those features so as to restrict the influence of noisy information.

---

*Equal contribution.
†Corresponding author.
[1]The code and the best performing models are available at https://github.com/cuhksz-nlp/AESINER
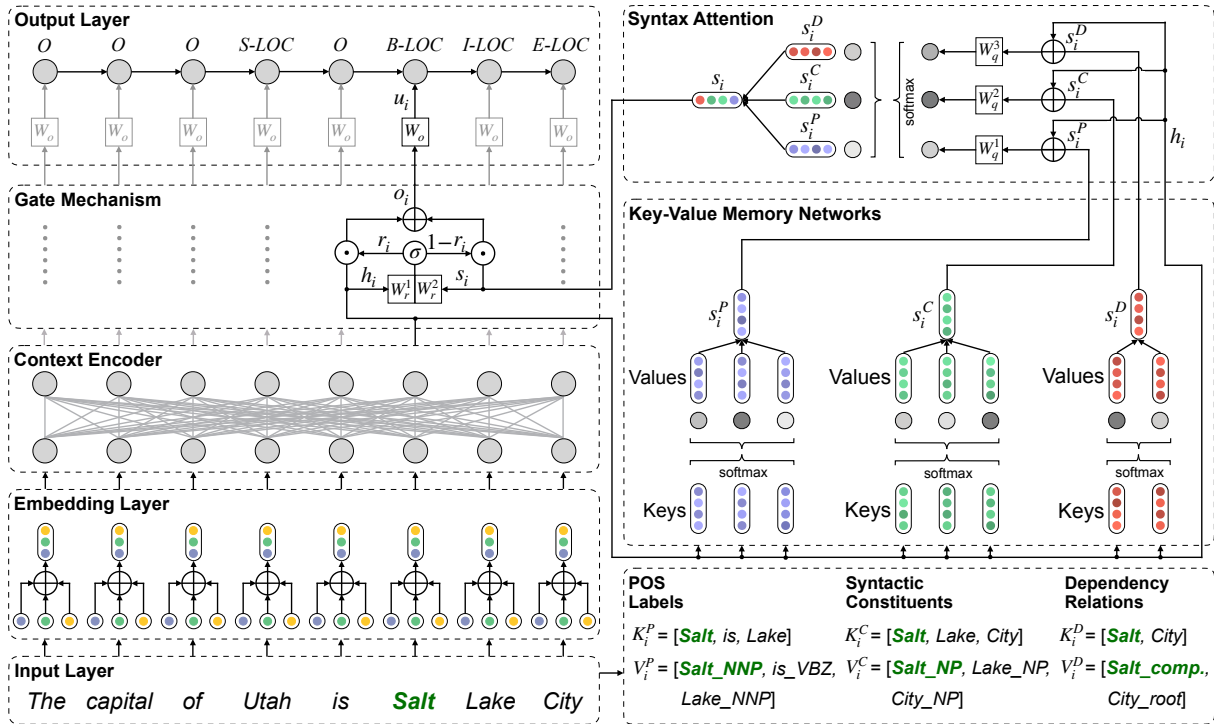
Figure 1: The overall architecture of the proposed NER model integrated with attentive ensemble of different syntactic information. An example input sentence and its output entity labels are given and the syntactic information for the word "*Salt*" are illustrated with their processing through KVMN, syntax attention and the gate mechanism.

In this paper, we propose a sequence labeling based neural model to enhance NER by incorporating different types of syntactic information, which is conducted by attentive ensemble with key-value memory networks (KVMN) (Miller et al., 2016), syntax attention and the gate mechanism. Particularly, the KVMN is applied to encode the context features and their syntax information from different types, e.g., POS labels, syntactic constituents, or dependency relations; syntax attention is proposed to weight different types of such syntactic information, and the gate mechanism controls the contribution of the results from the context encoding and the syntax attention to the NER process. Through the attentive ensemble, important syntactic information is highlighted and emphasized during labeling NEs. In addition, to further improve NER performance, we also try different types of pre-trained word embeddings, which is demonstrated to be effective in previous studies (Akbik et al., 2018; Jie and Lu, 2019; Liu et al., 2019b; Yan et al., 2019). We experiment our approach on six widely used benchmark datasets from the general domain, where half of them are in English and the other half are in Chinese. Experimental results on all datasets suggest the effectiveness of our approach to enhance NER through syntactic information, where state-of-the-art results are achieved on all datasets.

## 2 The Proposed Model

NER is conventionally regarded as a typical sequence labeling task, where an input sequence $\mathcal{X} = x_1, x_2, \cdots, x_i, \cdots, x_n$ with $n$ tokens is annotated with its corresponding NE labels $\widehat{\mathcal{Y}} = \widehat{y}_1, \widehat{y}_2, \cdots, \widehat{y}_i, \cdots, \widehat{y}_n$ in the same length. Following this paradigm, we propose a neural NER model depicted in Figure 1 with attentive ensemble to incorporate different types of syntactic information, where it can be conceptually formalized by

$$\widehat{\mathcal{Y}} = f(\mathcal{GM}(\mathcal{X}, \mathcal{SA}([\mathcal{M}^c(\mathcal{K}^c, \mathcal{V}^c)]_{c \in \mathcal{C}}))) \quad (1)$$

where $\mathcal{C}$ denotes the set of all syntactic information types and $c$ is one of them; $\mathcal{M}^c$ is the KVMN for encoding syntactic information of type $c$ with $\mathcal{K}^c$ and $\mathcal{V}^c$ referring to the keys and values in it, respectively; $\mathcal{SA}$ denotes the syntax attention to weight different types of syntactic information obtained through $\mathcal{M}^c$; $\mathcal{GM}$ refers to the gate mechanism to control how to use the encodings from context encoder and that from $\mathcal{SA}$. In the following text, we firstly introduce how we extract different types of syntactic information, then illustrate the attentive ensemble of syntactic information through KVMN, syntax attention, and gate mechanism, finally elaborate the encoding and decoding of the input text for NER as shown in the left part of Figure 1.
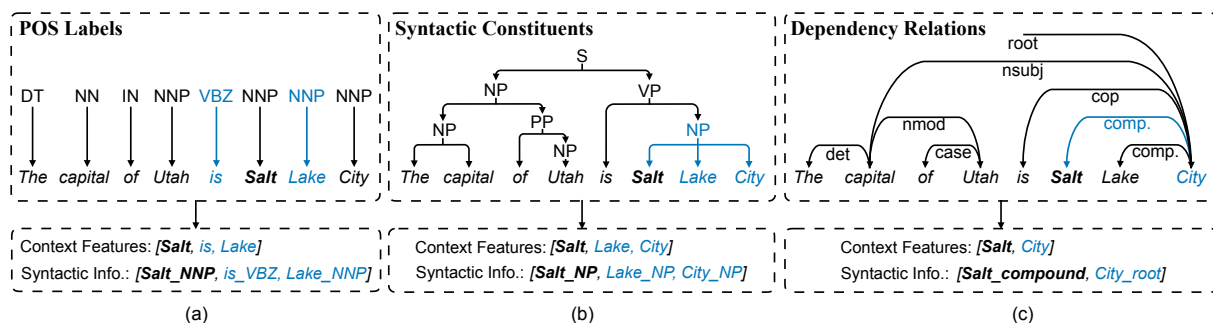
Figure 2: The extracted syntactic information in POS labels (a), syntactic constituents (b), and dependency relations (c) for *"Salt"* in the example sentence, where associated contextual features and the corresponding instances of syntactic information are highlighted in blue.

## 2.1 Syntactic Information Extraction

A good representation of the input text is the key to obtain good model performance for many NLP tasks (Song et al., 2017; Sileo et al., 2019). Normally, a straightforward way to improve model performance is to enhance text representation by embeddings of extra features, which is demonstrated to be useful across tasks (Marcheggiani and Titov, 2017; Song et al., 2018a; Zhang et al., 2019; Huang and Carley, 2019; Tian et al., 2020c), including NER (Zhang and Yang, 2018; Seyler et al., 2018; Sui et al., 2019; Gui et al., 2019b,a; Liu et al., 2019b). Among different types of extra features, the syntactic one has been proved to be helpful in previous studies for NER, where the effectiveness of POS labels, syntactic constituents, and dependency relations, are demonstrated by McCallum (2003), Li et al. (2017), and Cetoli et al. (2018), respectively. In this paper, we also focus on these three types of syntactic information. In doing so, we obtain the POS labels, the syntax tree and the dependency parsing results from an off-the-shelf NLP toolkit (e.g., Stanford Parser) for each input sequence $\mathcal{X}$. Then, for each token $x_i$ in $\mathcal{X}$, we extract its context features and related syntactic information according to the following procedures.

For **POS labels**, we treat every $x_i$ as the central word and employ a window of $\pm 1$ word to extract its context words and their corresponding POS labels. For example, in the example in Figure 2(a), for *"Salt"*, the $\pm 1$ word window covers its left and right words, so that the resulting context features are *"Salt"*, *"is"*, and *"Lake"*, and we use the combination of such words and their POS labels as the POS information (i.e., *"Salt_NNP"*, *"is_BVZ"*, and *"Lake_NNP"*) for the NER task.

For **syntactic constituents**, we start with $x_i$ at the leaf of $\mathcal{X}$'s syntax tree, then search up through

the tree to find the first acceptable syntactic node[2], and select all tokens under that node as the context features and the combination of tokens and their syntactic nodes as the constituent information. For example, in Figure 2(b), we start from *"Salt"* and extract its first accepted node *"NP"*, then collect the tokens under *"NP"* as the context features (i.e., *"Salt"*, *"Lake"*, and *"City"*) and combine them with *"NP"* to get the constituent information (i.e., *"Salt_NP"*, *"Lake_NP"*, and *"City_NP"*).

For **dependency relations**, we find all context features for each $x_i$ by collecting all its dependents and governor from $\mathcal{X}$'s dependency parse, and then regard the combination of the context features and their in-bound dependency types as the corresponding dependency information. For example, as illustrated in Figure 2(c), for *"Salt"*, its context features are *"Salt"* and *"City"* (the governor of *"Salt"*), and their corresponding dependency information are *"Salt_compound"* and *"City_root"*.[3]

As a result, for each type of syntactic inforamtion, we obtain a list of context features and a list of syntactic information instances, which are modeled by a KVMN module to enhance input text representation and thus improve model performance.

## 2.2 KVMN for Syntactic Information

Since the syntactic information is obtained from off-the-shelf toolkits, it is possible that there is noise in the extracted syntactic information, which may hurt model performance if it is not leveraged appropriately. Inspired by the studies that

---

[2]There are 10 accepted constituent nodes, including *NP*, *VP*, *PP*, *ADVP*, *SBAR*, *ADJP*, *PRT*, *INTJ*, *CONJP* and *LST*, which are selected from the types used in the CoNLL-2003 shared task (Sang and Meulder, 2003).

[3]Note that, in this case, we do not have context features selected from the dependents since *"Salt"* do not have any dependents according to the dependency parse result.

use KVMN and its variants to weight and leverage extra features to enhance model performance in many NLP tasks (Miller et al., 2016; Mino et al., 2017; Xu et al., 2019b; Tian et al., 2020d), for each type of the syntactic information (denoted as $c$), we propose a KVMN module ($\mathcal{M}^c$) to model the pair-wisely organized context features and the syntactic information instances. Specifically, for each $x_i$ in the input, we firstly map its context features and the syntactic information to keys and values in the KVMN, which are denoted by $\mathcal{K}_i^c = [k_{i,1}^c, \ldots, k_{i,j}^c, \ldots, k_{i,m_i}^c]$ and $\mathcal{V}_i^c = [v_{i,1}^c, \ldots, v_{i,j}^c, \ldots, v_{i,m_i}^c]$, respectively, with $m_i$ the number of context features for $x_i$. Next, we use two matrices to map them to their embeddings, with $\mathbf{e}_{i,j}^{k_c}$ referring to the embedding of $k_{i,j}^c$ and $\mathbf{e}_{i,j}^{v_c}$ for $v_{i,j}^c$, respectively. Then, for each token $x_i$ and its associated context features $\mathcal{K}_i^c$ and syntactic information $\mathcal{V}_i^c$, the weight assigned to the syntactic information $v_{i,j}^c$ is computed by

$$p_{i,j}^c = \frac{exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^{k_c})}{\sum_{j=1}^{m_i} exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^{k_c})} \tag{2}$$

where $\mathbf{h}_i$ is the hidden vector for $x_i$ obtained from the context encoder. Afterwards, we apply the weights $p_{i,j}^c$ to their corresponding syntactic information $v_{i,j}^c$ by

$$\mathbf{s}_i^c = \sum_{j=1}^{m_i} p_{i,j}^c \mathbf{e}_{i,j}^{v_c} \tag{3}$$

where $\mathbf{s}_i^c$ is the output of $\mathcal{M}^c$, containing the weighted syntactic information in type $c$. Therefore, KVMN ensures that the syntactic information are weighted according to their corresponding context features, so that important information could be distinguished and leveraged accordingly.

## 2.3 The Syntax Attention

Upon encoding each type of syntactic information by KVMN, one can assemble different types of them with an overall representation. The most straightforward way of doing so is to concatenate the encoding from each type by

$$\mathbf{s}_i = \bigoplus_{c \in \mathcal{C}} \mathbf{s}_i^c \tag{4}$$

where $\mathbf{s}_i$ is the aggregated results of $\mathbf{s}_i^c$, the embedding for each syntactic type from $\mathcal{M}^c$. However, given the fact that different syntactic information may conflict to each other, it is expected to have a more effective way to combine them.

Motivated by studies that selectively leverage different features by assigning different weights to them (Kumar et al., 2018; Higashiyama et al., 2019; Tian et al., 2020a,b), we propose a syntax attention for the syntactic information ensemble. Particularly, for each syntactic type $c$, we firstly concatenate $\mathbf{s}_i^c$ with $\mathbf{h}_i$ and use the resulting vector to compute the weight $q_i^c$ for $\mathbf{s}_i^c$:

$$q_i^c = \sigma(\mathbf{W}_q^c \cdot (\mathbf{h}_i \oplus \mathbf{s}_i^c) + b_q^c) \tag{5}$$

where $\mathbf{W}_q^c$ and $b_q^c$ are trainable vector and variable, respectively, and $\sigma$ is the *sigmoid* function. Then, a *softmax* function is applied over all types of syntactic information to compute their corresponding attentions $a_i^c$, which is formalized by

$$a_i^c = \frac{exp(q_i^c)}{\sum_{c \in \mathcal{C}} exp(q_i^c)} \tag{6}$$

In the last, we apply the weights to their corresponding encoded syntactic information vectors by

$$\mathbf{s}_i = \sum_{c \in \mathcal{C}} a_i^c \mathbf{s}_i^c \tag{7}$$

where $\mathbf{s}_i$ is the output of the syntax attention of different syntactic information types.

## 2.4 The Gate Mechanism

To enhance NER with the syntactic information encoded by KVMN and combined by syntax attention, we propose a gate mechanism ($\mathcal{GM}$) to incorporate it to the backbone NER model, where we expect such mechanism could dynamically weight and decide how to leverage the syntactic information in labeling NEs. In detail, we propose a *reset* function $\mathbf{r}_i$ to evaluate the encodings from the context encoder and the syntax attention by

$$\mathbf{r}_i = \sigma(\mathbf{W}_{r_1} \cdot \mathbf{h}_i + \mathbf{W}_{r_2} \cdot \mathbf{s_i} + \mathbf{b}_r) \tag{8}$$

where $\mathbf{W}_{r_1}, \mathbf{W}_{r_2}$ are trainable matrices and $\mathbf{b}_r$ the bias term, and use

$$\mathbf{o}_i = [\mathbf{r}_i \circ \mathbf{h}_i] \oplus [(\mathbf{1} - \mathbf{r}_i) \circ \mathbf{s}_i] \tag{9}$$

to control the contribution of them, where $\mathbf{o}_i$ is the output of the gate mechanism corresponding to input $x_i$, $\mathbf{1}$ is a 1-vector with its dimension matching $\mathbf{h}_i$ and $\circ$ the element-wise multiplication operation.

## 2.5 Encoding and Decoding for NER

To provide $\mathbf{h}_i$ to KVMN, we adopt Adapted-Transformer[4] (Yan et al., 2019) as the context en-

---

[4]The Adapted-Transformer additionally models direction and distance information of the input, which are demonstrated

| TYPE | ENGLISH | | | | | | CHINESE | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ON5E | | WN16 | | WN17 | | ON4C | | RE | | WE | |
| | # T. = 18 | | # T. = 10 | | # T. = 6 | | # T. = 4 | | # T. = 8 | | # T. = 4 | |
| | # S. | # E. | # S. | # E. | # S. | # E. | # S. | # E. | # S. | # E. | # S. | # E. |
| TRAIN | 59.9K | 81.8K | 2.4K | 1.5K | 3.4K | 2.0K | 15.7K | 13.4K | 3.8K | 13.4K | 1.4K | 1.9K |
| DEV | 8.5K | 11.1K | 1.0K | 0.7K | 1.0K | 0.8K | 4.3K | 7.0K | 0.5K | 1.5K | 0.3K | 0.4K |
| TEST | 8.3K | 11.3K | 3.9K | 3.5K | 1.3K | 1.1K | 4.3K | 7.7K | 0.5K | 1.6K | 0.3K | 0.4K |

Table 1: Statistics of all datasets with respect to the number of NE types (T.), sentences (S.), and total NEs (E.).

coder in this work. So that the encoding of the input text can be formalized as

$$\mathbf{H} = \text{Adapted-Transformer}(\mathbf{E}) \qquad (10)$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_i, \cdots, \mathbf{h}_n]$ and $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_i, \cdots, \mathbf{e}_n]$ are lists of hidden vectors and embeddings of $\mathcal{X}$, respectively. Note that, since pre-trained embeddings contain context information learned from large-scale corpora, and different types of them may carry heterogeneous context information learned from different algorithms and corpora, we incorporate multiple pre-trained embeddings by direct concatenating them in the input:

$$\mathbf{e}_i = \underset{z \in \mathcal{Z}}{\oplus} \mathbf{e}_i^z \qquad (11)$$

where $\mathbf{e}_i$ is the final word representation to feed the context encoder; $\mathbf{e}_i^z$ represents the word embedding of $x_i$ in embedding type $z$ and $\mathcal{Z}$ the set of all embedding types.

For the output, upon the receiving of $\mathbf{o}_i$, a trainable matrix $\mathbf{W}_o$ is used to align its dimension to the output space by $\mathbf{u}_i = \mathbf{W}_o \cdot \mathbf{o}_i$. Finally, we apply a conditional random field (CRF) decoder to predict the labels $\hat{y}_i \in \mathcal{T}$ (where $\mathcal{T}$ is the set with all NE labels) in the output sequence $\hat{\mathcal{Y}}$ by

$$\hat{y}_i = \underset{y_i \in \mathcal{T}}{\arg\max} \frac{exp(\mathbf{W}_c \cdot \mathbf{u}_i + \mathbf{b}_c)}{\sum_{y_{i-1} y_i} exp(\mathbf{W}_c \cdot \mathbf{u}_i + \mathbf{b}_c)} \qquad (12)$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are trainable parameters to model the transition for $y_{i-1}$ to $y_i$.

## 3 Experimental Settings

### 3.1 Datasets

In our experiments, we use three English benchmark datasets, i.e., OntoNotes 5.0 (ON5e) (Pradhan et al., 2013) , WNUT-16 (WN16), WNUT-17 (WN17) (Derczynski et al., 2017), and three Chinese datasets, i.e., OntoNotes 4.0 (ON4c) (Weischedel et al., 2011), Resume (RE) (Zhang and Yang, 2018), Weibo (WE) (Peng and Dredze,

2015).[5] These datasets come from a wide range of sources so that we are able to comprehensively evaluate our approach with them. In detail, WN16 and WN17 are constructed from social media; ON5e consists of mixed sources, such as telephone conversation, newswire, etc.; ON4c is from news domain; RE and WE are extracted from Chinese online resources. For all datasets, we use their original splits and the statistics of them with respect to the number of entity types (# T.), sentences (# S.) and total entities (# E.) in the train/dev/test sets are reported in Table 1.

### 3.2 Implementation

To label NEs, we use the **BIOES** tagging scheme instead of the standard **BIO** scheme for the reason that previous studies have shown optimistic improvement with this scheme (Lample et al., 2016; Yan et al., 2019). For the text input, we use three types of embeddings for each language by default. Specifically, for English, we use Glove (100-dimension)[6] (Pennington et al., 2014), ELMo (Peters et al., 2018), and the BERT-cased large[7] (Devlin et al., 2019) (the derived embeddings for each word); for Chinese, we use pre-trained character and bi-gram embeddings[8] released by Zhang and Yang (2018) (denoted as Giga), Tencent Embedding[9] (Song et al., 2018b), and ZEN[10] (Diao et al., 2019). For both BERT and ZEN, we follow their

---

to be useful for NER comparing to the vanilla Transformer.

[5] Among these datasets, ON5e and ON4c are multi-lingual datasets. We follow Yan et al. (2019) by extracting the corresponding English and Chinese part from them.

[6] We download the Glove.6B embedding from https://nlp.stanford.edu/projects/glove/

[7] We obtain the pre-trained BERT from https://github.com/google-research/bert.

[8] We obtain the embeddings from https://github.com/jiesutd/LatticeLSTM.

[9] We use the official release from https://ai.tencent.com/ailab/nlp/embedding.html.

[10] We use the pre-trained ZEN-base downloaded from https://github.com/sinovation/ZEN. Note that we do not use the Chinese BERT since ZEN performs better across three Chinese datasets. For reference, we report the results of using BERT in Appendix A.

| SYNTACTIC INFORMATION | | | ON5E | WN16 | WN17 | ON4C | RE | WE |
|---|---|---|---|---|---|---|---|---|
| POS. | CON. | DEP. | | | | | | |
| | | | 89.32 | 53.81 | 48.96 | 79.04 | 95.84 | 67.79 |
| √ | | | 89.51 | 53.94 | 49.68 | 79.53 | 96.09 | **68.76** |
| | √ | | **89.64** | **54.59** | **49.82** | 79.76 | **96.11** | 68.11 |
| | | √ | 89.58 | 54.37 | 49.47 | **80.03** | 96.02 | 68.64 |

Table 2: $F1$ scores of the baseline model and ours enhanced with different types of syntactic information ("POS.", "CON." and "DEP." refer to POS labels, syntactic constituents and dependency relations, respectively).

| TYPE | SYNTACTIC INFORMATION | | | ON5E | WN16 | WN17 | ON4C | RE | WE |
|---|---|---|---|---|---|---|---|---|---|
| | POS. | CON. | DEP. | | | | | | |
| $\mathcal{DC}$ | √ | √ | | 89.61 | 54.11 | 49.61 | 79.61 | 95.72 | 68.27 |
| | √ | | √ | 89.56 | 54.03 | 49.74 | 79.83 | 96.11 | 68.51 |
| | | √ | √ | 89.60 | 54.26 | 49.58 | 79.89 | 96.08 | 68.36 |
| | √ | √ | √ | 89.62 | 54.41 | 49.63 | 79.81 | 95.31 | 68.49 |
| $\mathcal{SA}$ | √ | √ | | 89.68 | 54.68 | 49.81 | 79.92 | 96.19 | 68.94 |
| | √ | | √ | 89.76 | 54.61 | 49.89 | 80.29 | 96.23 | 69.01 |
| | | √ | √ | 89.78 | 54.56 | 49.96 | 80.41 | 96.31 | 68.76 |
| | √ | √ | √ | **89.86** | **54.79** | **50.21** | **80.65** | **96.43** | **69.37** |

Table 3: $F1$ scores of our models with different combinations of syntactic information. "TYPE" indicates how they are combined, where "$\mathcal{DC}$" and "$\mathcal{SA}$" refer to direct concatenation and syntax attention, respectively.

default settings, i.e., 24 layers of self-attention with 1024 dimensional embeddings for BERT-large and 12 layers of self-attention with 768 dimensional embeddings for ZEN-base. For syntactic information, we use the Stanford CoreNLP Toolkit[11] (Manning et al., 2014) to produce the aforementioned three types of syntactic information, i.e. POS labels, syntactic constituents, and dependency relations, for each input text. In the context encoding layer, we use a two-layer Adapted-Transformer encoder[12] with 128 hidden units and 12 heads and set the dropout rate to 0.2. For the memory module, all key and value embeddings are initialized randomly. During the training process, we fix all pretrained embeddings and use Adam (Kingma and Ba, 2015) to optimize negative log-likelihood loss function with the learning rate set to $\eta = 0.0001$, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. In all experiments, we run a maximum of 100 epochs with the batch size of 32 and tune the hyper-parameters on the development set.[13] The model that achieves the highest

performance on the development set is evaluated on the test set with respect to the $F1$ scores obtained from the official *conlleval* toolkits[14].

## 4 Experimental Results

### 4.1 Effect of Key-Value Memory Networks

To explore how different syntactic information helps NER, we run the baselines without syntactic information and the ones with each type of syntactic information through KVMN.[15] Experimental results ($F1$) are reported in Table 2 for all datasets.

It is observed from the results that the models with syntactic information outperform the baseline in all cases, which demonstrates the effectiveness of using KVMN in our model. In addition, it is also noticed that the best performed model is not exactly the same one across different datasets, which indicates the contributions of different syntactic information vary in those datasets. For example, in most datasets, models using syntactic constituents achieve the best results, which can be explained by that syntactic constituents provide important cues of NE chunks. As a comparison, POS labels are

---

[11]We use its 3.9.2 version downloaded from `https://stanfordnlp.github.io/CoreNLP/`.

[12]We also try other encoders (i.e., Bi-LSTM and Transformer) and report their results in Appendix B for reference.

[13]We report the hyper-parameter settings of different models as well as the best one in Appendix C.

[14]`https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt`.

[15]Syntax attention and the gate mechanism are not applied.

| $\mathcal{GM}$ | Syntactic Information POS. | CON. | DEP. | ON5E | WN16 | WN17 | ON4C | RE | WE |
|---|---|---|---|---|---|---|---|---|---|
| | √ | √ | | 89.68 | 54.68 | 49.81 | 79.92 | 96.19 | 68.94 |
| √ | √ | √ | | 90.09 | 54.92 | 50.28 | 80.31 | 96.51 | 69.31 |
| | √ | | √ | 89.76 | 54.61 | 49.89 | 80.29 | 96.23 | 69.01 |
| √ | √ | | √ | 90.08 | 54.78 | 50.16 | 80.64 | 96.47 | 69.47 |
| | | √ | √ | 89.78 | 54.56 | 49.96 | 80.41 | 96.31 | 68.76 |
| √ | | √ | √ | 90.11 | 54.96 | 50.36 | 80.87 | 96.51 | 69.24 |
| | √ | √ | √ | 89.86 | 54.79 | 50.21 | 80.65 | 96.43 | 69.37 |
| √ | √ | √ | √ | **90.32** | **55.14** | **50.68** | **81.18** | **96.62** | **69.78** |

Table 4: $F1$ scores of our models with and without applying the gate mechanism ("$\mathcal{GM}$") when different syntactic information are combined by syntactic attention.

| ENGLISH | | | | CHINESE | | | |
|---|---|---|---|---|---|---|---|
| MODEL | ON5E | WN16 | WN17 | MODEL | ON4C | RE | WE |
| CHIU AND NICHOLS (2016) | 86.12 | - | - | ZHANG AND YANG (2018) | 73.88 | - | 58.79 |
| †LUO ET AL. (2018) | 88.79 | 51.26 | 48.63 | YAN ET AL. (2019) | 72.43 | 95.00 | 58.17 |
| †DANG ET AL. (2018) | 88.91 | 51.84 | 48.12 | GUI ET AL. (2019B) | 74.89 | 95.37 | 60.21 |
| AKBIK ET AL. (2018) | 89.30 | - | - | ZHU AND WANG (2019) | 73.64 | 94.94 | 59.31 |
| JIE AND LU (2019) | 89.88 | - | - | GUI ET AL. (2019A) | 74.45 | 95.11 | 59.92 |
| YAN ET AL. (2019) | 89.78 | 54.06 | 48.98 | LIU ET AL. (2019C) | 74.43 | 95.21 | 59.84 |
| *DEVLIN ET AL. (2019) | 89.16 | 54.36 | 49.52 | SUI ET AL. (2019) | 74.79 | - | 63.09 |
| ZHOU ET AL. (2019) | - | 53.43 | 42.83 | DING ET AL. (2019) | 76.00 | - | 59.50 |
| AKBIK ET AL. (2019) | - | - | 49.59 | *MENG ET AL. (2019) | 80.62 | 96.54 | 67.60 |
| DAI ET AL. (2019) | 89.83 | - | - | XU ET AL. (2019A) | - | - | 68.93 |
| LIU ET AL. (2019B) | 89.94 | - | - | MA ET AL. (2020) | 75.54 | 95.59 | 61.24 |
| *LUO ET AL. | 90.30 | - | - | *HU AND WEI (2020) | 80.20 | 95.80 | 64.00 |
| OURS | 90.32 | 55.14 | 50.68 | OURS | 81.18 | 96.62 | 69.78 |

Table 5: Comparison of $F1$ scores of our best performing model (i.e. the full model with attentive ensemble of all syntactic information) with that reported in previous studies on all English and Chinese benchmark datasets. "*" indicates the studies using BERT as the text encoder; "†" means the results are our runs of their models.

the most effective syntactic information for WE dataset, which can be attributed to the natural of the dataset that most sentences in social media are not formally written, so that their parsing results could be inaccurate and mislead the NER process.

## 4.2 Effect of Syntax Attention

To examine the effectiveness of syntax attention ($\mathcal{SA}$), we compare it with another strategy through direct concatenation ($\mathcal{DC}$) of the KVMN output to model ouput. The results are reported in Table 3 with applying all combinations of different syntactic information by $\mathcal{DC}$ and $\mathcal{SA}$.

There are several observations. First, interestingly, compared to the results in Table 2, direct concatenation of different syntactic information hurts NER performance in most cases. For example, on the RE dataset, the ensemble of all types of syntactic information through $\mathcal{DC}$ obtains the worst results compared to all other results with integrating less information under the same setting. The reason behind this phenomenon could be that differ-

ent syntactic information may provide conflict cues to NE tags and thus result in inferior performance. Second, on the contrary, $\mathcal{SA}$ is able to improve NER with integrating multiple types of syntactic information, where consistent improvements are observed among all datasets when more types of syntactic information are incorporated. As a result, the best results are achieved by the model using all types of syntactic information. This observation suggests that the syntax attention is able to weight different syntactic information and distinguish important ones from others, thus alleviate possible conflicts of them when labeling entities.

## 4.3 Effect of the Gate Mechanism

We experiment our model under its best setting (i.e., $\mathcal{SA}$ over all combinations of syntactic information with KVMN) with and without the gate mechanism to investigate its effectiveness of actively controlling the information flow from the context encoder and $\mathcal{SA}$. The results are presented in Table 4, where the ones without using the gate

| ENGLISH | | | | | | CHINESE | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EMBEDDINGS | | | ON5E | WN16 | WN17 | EMBEDDINGS | | | ON4C | RE | WE |
| GLOVE | ELMO | BERT | | | | GIGA | TENCENT | ZEN | | | |
| √ | | | 89.37 | 47.92 | 43.24 | √ | | | 72.11 | 94.99 | 61.94 |
| | √ | | 89.71 | 53.96 | 47.92 | | √ | | 73.54 | 95.21 | 63.06 |
| | | √ | 89.53 | 53.74 | 48.74 | | | √ | 80.06 | 95.98 | 68.84 |
| √ | √ | | 89.91 | 54.36 | 48.21 | √ | √ | | 74.86 | 95.46 | 63.96 |
| √ | | √ | 89.82 | 54.16 | 49.61 | √ | | √ | 80.49 | 96.24 | 68.94 |
| | √ | √ | 90.13 | 54.92 | 50.12 | | √ | √ | 80.81 | 96.41 | 69.42 |
| √ | √ | √ | **90.32** | **55.14** | **50.68** | √ | √ | √ | **81.18** | **96.62** | **69.78** |

Table 6: Experimental results ($F1$ scores) of our best performing model (i.e., the full model with attentive ensemble of all syntactic information) using different pre-trained embeddings and their combinations as input.

mechanism are obtained directly from Table 3.[16] It is clearly shown that in all cases, the model with gate mechanism achieves superior performance to the other one without it. These results suggest that the importance of the information from the context encoder and $\mathcal{SA}$ varies, so that the proposed gate mechanism is effective in balancing them.

### 4.4 Comparison with Previous Studies

To further illustrate the effectiveness of our models, we compare the best performing one, i.e., the last line in Table 4, with the results from previous studies. The results are shown in Table 5, where our approach outperforms previous models with BERT encoder (marked by "*") and achieves state-of-the-art results on all English and Chinese datasets. This observation indicates that incorporate different embeddings as input is more effective than directly using pre-trained models. In addition, compared to some previous studies (Luo et al., 2018; Dang et al., 2018)[17] that leverage multiple types of syntactic information by regarding the information as gold references and directly concatenating their embeddings with word embeddings, our approach has its superiority by using attentive ensemble through KVMN, syntax attention, and the gate mechanism to selectively learn from different syntactic information according to their contribution to NER, where such multi-phase strategy of attentive ensemble guarantees the appropriateness of learning them in a reasonable manner.

## 5 Analyses

### 5.1 Effect of Different Word Embeddings

Neural models are sensitive to input embeddings, which is also true for our approach. Consider that different types of embeddings carry contextual information learned from various corpora and algorithms, we explore the effect of those embeddings when they are used separately or combined as the input. The experiment is performed on our best model (i.e., KVMN+$\mathcal{SA}$+$\mathcal{GM}$ on all syntactic information), with the results reported in Table 6. It is clearly observed that for all English and Chinese datasets, the model with all three embeddings achieves the best performance and its performance drops consistently when more types of embeddings are excluded. It is confirmed that different types of embedding do provide complement context information to enhance the understanding of input texts for NER. Particularly, although contextualized embeddings (i.e., ELMo, BERT, and ZEN) show significantly better performance than others (especially on Chinese), combining them with static embeddings still provide further improvement on the $F1$ score of NER systems.

### 5.2 Case Study

To better understand how attentive ensemble of syntactic information helps NER, we conduct a case study for the word "*Bill*" in an example sentence "*Mason was one of the drafters of the Bill of Rights*" from the ON5e dataset. Figure 3 visualizes the weights for different context features in KVMN, as well as the weights from $\mathcal{SA}$ and $\mathcal{GM}$, where darker colors refer to higher weights.[18]

Interestingly, in this case, both POS labels and dependency relations receiving highest weights

---

[16]The results of those models on the development sets of all datasets are reported in Appendix D.

[17]Luo et al. (2018) and Dang et al. (2018) do not report their results on all general domain benchmark datasets, because the focus of their studies is biomedical NER. Therefore, we report our runs of their method in Table 5 (marked by "†").

[18]For the weights, we visualize $p_{i,j}^c$ in Eq. (2) for KVMN, $a_i^c$ in Eq. (6) for $\mathcal{SA}$, and $||\mathbf{r}_i||_2$ for $\mathbf{r_i}$ in Eq. (8) for $\mathcal{GM}$.
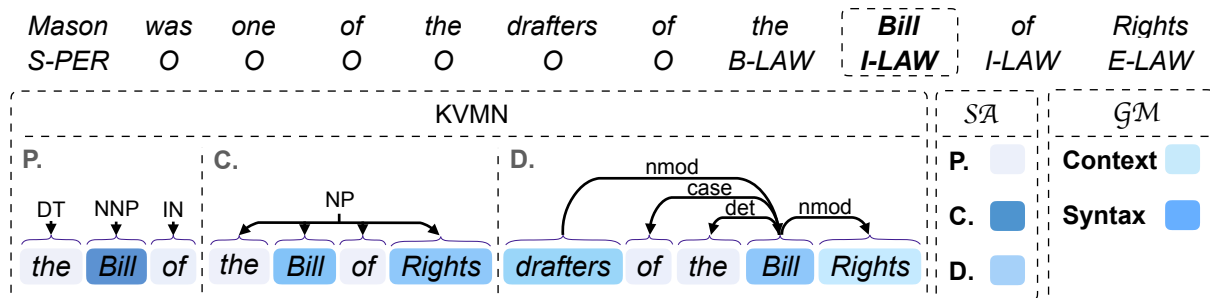
Figure 3: An illustration of how our model encodes syntactic information through KVMN, weights them by syntax attention ($\mathcal{SA}$) and learns from the gate mechanism ($\mathcal{GM}$), where the weights for different features and information types are visualized. The example sentence is shown at the top with the gold NE tags for each word marked below. The weights assigned to different syntactic information for "*Bill*" in KVMN, $\mathcal{SA}$, and $\mathcal{GM}$ are highlighted with colors, where the darker colors referring to higher values.

suggest a misleading "*PERSON*" label[19] because of their context features, so that an incorrect NER prediction is expected if treating the three types of syntactic information equally. However, the syntactic constituents give strong indication of the correct label through the word "*Rights*" for a "*LAW*" entity. Later, the syntax attention ensures that the constituent information should be emphasized and the gate mechanism also tends to use syntax for this input with higher weights. Therefore this case clearly illustrates the contribution of each component in our attentive ensemble of syntactic information.

## 6 Related Work

Recently, neural models play dominant roles in NER because of their effectiveness in capturing contextual information in the text without requiring to extract manually crafted features (Huang et al., 2015; Lample et al., 2016; Strubell et al., 2017; Zhang and Yang, 2018; Peters et al., 2018; Yadav and Bethard, 2018; Cetoli et al., 2018; Akbik et al., 2018, 2019; Chen et al., 2019; Devlin et al., 2019; Zhu and Wang, 2019; Liu et al., 2019b; Baevski et al., 2019; Yan et al., 2019; Xu et al., 2019a; Zhu et al., 2020; Luo et al.). However, to enhance NER, it is straightforward to incorporate more knowledge to it than only modeling from contexts. Therefore, additional resources such as knowledge base (Kazama and Torisawa, 2008; Tkachenko and Simanovsky, 2012; Seyler et al., 2018; Liu et al., 2019b,a; Gui et al., 2019b,a) and syntactic information (McCallum, 2003; Mohit and Hwa, 2005; Finkel and Manning, 2009; Li et al., 2017; Luo et al., 2018; Cetoli et al., 2018; Jie and Lu, 2019) are applied in previous studies. Particularly, Luo et al. (2018) and Dang et al. (2018)

exploited POS labels and syntactic constituents in their methods and found that the combination of them improves NER performance. Yet they are limited in regarding such syntactic information as gold references and directly concatenated them to the input embeddings, so that noises are expected to affect NER accordingly. Compared with them, our model provides an alternative option to leverage syntactic information with attentive ensemble to encode, weight and select them to help NER, which proves its effectiveness and has the potential to be applied in other similar tasks.

## 7 Conclusion

In this paper, we proposed a neural model following the sequence labeling paradigm to enhance NER through attentive ensemble of syntactic information. Particularly, the attentive ensemble consists of three components in a sequence: each type of syntactic information is encoded by key-value memory networks, different information types are then weighted in syntax attention, and the gate mechanism is finally applied to control the contribution of syntax attention outputs to NER for different contexts. In doing so, different syntactic information are comprehensively and selectively learned to enhance NER, where the experimental results on six benchmark datasets in English and Chinese confirm the validity and effectiveness of our model with state-of-the-art performance obtained.

### Acknowledgments

---

[19]"*Bill*" is part of the "*PERSON*" entity in most cases.

# References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled Contextualized Embeddings for Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven Pretraining of Self-attention Networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5359–5368.

Daniel M. Bikel, Scott Miller, Richard M. Schwartz, and Ralph M. Weischedel. 1997. Nymble: a High-Performance Learning Name-finder. In *5th Applied Natural Language Processing Conference, ANLP 1997, Marriott Hotel, Washington, USA, March 31 - April 3, 1997*, pages 194–201.

Alberto Cetoli, Stefano Bragaglia, Andrew D. O'Harney, and Marc Sloan. 2018. Graph Convolutional Networks for Named Entity Recognition. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories, TLT 2018, Prague, Czech Republic, January 23-24, 2018*, pages 37–45.

Hui Chen, Zijia Lin, Guiguang Ding, Jianguang Lou, Yusen Zhang, and Börje Karlsson. 2019. GRN: Gated Relation Network to Enhance Convolutional Neural Network for Named Entity Recognition. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6236–6243.

Jason P. C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *TACL*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Zeyu Dai, Hongliang Fei, and Ping Li. 2019. Coreference Aware Representation Learning for Neural Named Entity Recognition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4946–4953.

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.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. *arXiv*, abs/1911.00720.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A Neural Multi-digraph Model for Chinese NER with Gazetteers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1462–1467.

Li Dong and Mirella Lapata. 2018. Coarse-to-Fine Decoding for Neural Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 731–742.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint Parsing and Named Entity Recognition. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 326–334.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019a. CNN-Based Chinese NER with Lexicon Rethinking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4982–4988.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019b. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1040–1050.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 2019. Incorporating Word Attention into Character-Based Word Segmentation. 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 2699–2709, Minneapolis, Minnesota.

Dou Hu and Lingwei Wei. 2020. SLK-NER: Exploiting Second-order Lexicon Knowledge for Chinese NER (S). In *The 32nd International Conference on Software Engineering and Knowledge Engineering, SEKE 2020, KSIR Virtual Conference Center, USA, July 9-19, 2020*, pages 413–417.

Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. 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 5472–5480.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv*, abs/1508.01991.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3860–3870.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing Gazetteers for Named Entity Recognition by Large-Scale Clustering of Dependency Relations. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 407–415.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Abhishek Kumar, Daisuke Kawahara, and Sadao Kurohashi. 2018. Knowledge-Enriched Two-Layered Attention Network for Sentiment Analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 253–258, New Orleans, Louisiana.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun 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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2664–2669.

Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019a. Knowledge-Augmented Language Model and Its Application to Unsupervised Named-Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1142–1150.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019b. Towards Improving Neural Named Entity Recognition with Gazetteers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5301–5307.

Wei Liu, Tongge Xu, QingHua Xu, Jiayu Song, and Yueran Zu. 2019c. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2379–2389.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019d. GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2431–2441.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. 2018. An Attention-based BiLSTM-CRF Approach to Document-Level Chemical Named Entity Recognition. *Bioinformatics*, 34(8):1381–1388.

Ying Luo, Fengshun Xiao, and Hai Zhao. Hierarchical Contextualized Representation for Named Entity

Recognition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, pages 8441–8448.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5951–5960.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford Corenlp Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2019. Joint Learning of Named Entity Recognition and Entity Linking. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 190–196.

Andrew McCallum. 2003. Efficiently Inducing Features of Conditional Random Fields. In *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, Acapulco, Mexico, August 7-10 2003*, pages 403–410.

Andrew McCallum and Wei Li. 2003. Early Results for Named entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 188–191.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese Character Representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 2742–2753.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Hideya Mino, Masao Utiyama, Eiichiro Sumita, and Takenobu Tokunaga. 2017. Key-value Attention Mechanism for Neural Machine Translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 290–295, Taipei, Taiwan.

Behrang Mohit and Rebecca Hwa. 2005. Syntax-based Semi-Supervised Named Entity Tagging. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 57–60.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Lixin Su, and Xueqi Cheng. 2019. HAS-QA: Hierarchical Answer Spans Model for Open-Domain Question Answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6875–6882.

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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards Robust Linguistic Analysis using Ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 143–152.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the Conll-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147.

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A Study of the Importance of External Knowledge in the Named Entity Recognition Task. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 241–246.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining Discourse Markers for Unsupervised Sentence Representation Learning. 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 3477–3486, Minneapolis, Minnesota.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song, Shuming Shi, and Jing Li. 2018a. Joint Learning Embeddings for Chinese Words and their Components via Ladder Structured Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4375–4381.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018b. Directional Skip-Gram: Explicitly Distinguishing Left and Right Context for Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 175–180.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2670–2680.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3828–3838.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. 2020c. Improving Constituency Parsing with Span Attention. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020d. Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285.

Maksim Tkachenko and Andrey Simanovsky. 2012. Named Entity Recognition: Exploring Features. In *11th Conference on Natural Language Processing, KONVENS 2012, Empirical Methods in Natural Language Processing, Vienna, Austria, September 19-21, 2012*, pages 118–127.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, and Robert Belvin. 2011. OntoNotes Release 4.0.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019a. Exploiting Multiple Embeddings for Chinese Named Entity Recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2269–2272.

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019b. Enhancing Key-Value Memory Neural Networks for Knowledge Based Question Answering. 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 2937–2947, Minneapolis, Minnesota.

Vikas Yadav and Steven Bethard. 2018. A Survey on Recent Advances in Named Entity Recognition from Deep Learning Models. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2145–2158.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv*, abs/1911.04474.

Hongming Zhang, Yan Song, and Yangqiu Song. 2019. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. 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 872–881, Minneapolis, Minnesota.

Yue Zhang and Jie Yang. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1554–1564.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual Adversarial Neural Transfer for Low-Resource Named Entity Recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3461–3471.

Huiming Zhu, Chunhui He, Yang Fang, and Weidong Xiao. 2020. Fine Grained Named Entity Recognition via Seq2seq Framework. *IEEE Access*, 8:53953–53961.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3384–3393.

## Appendix A: Comparison Between BERT and ZEN on Chinese Datasets

| EMBEDDINGS | ON4C | RE | WE |
|---|---|---|---|
| BERT + GIGA + TENCENT | 80.91 | 96.56 | 69.61 |
| ZEN + GIGA + TENCENT | **81.18** | **96.62** | **69.78** |

Table 7: Experimental results ($F1$ scores) of our models (i.e. the full model with attentive ensemble of all syntactic information), where BERT or ZEN is used as one of the three types of embeddings (the others are Giga and Tencent Embedding) in the embedding layer.

In the main experiments, we use ZEN (Diao et al., 2019) rather than BERT (Devlin et al., 2019) as part of the embeddings of input texts on all Chinese datasets. The reason is that compared to BERT, ZEN achieves better performance across all Chinese datasets, which is demonstrated in Table 7. In this experiment, the results ($F1$ scores) show the performance of our best performing model (i.e. the full model with attentive ensemble of all syntactic

information) on the test set of all three Chinese datasets. Specifically, either BERT or ZEN is used as one of the three types of embeddings (the others are Giga and Tencent Embedding).

## Appendix B: Effect of Different Context Encoders

| Context Encoder | Syntactic Information | ON5e | WN16 | WN17 |
|---|---|---|---|---|
| Bi-LSTM | | 88.56 | 51.16 | 48.11 |
| | √ | 89.64 | 53.39 | 49.56 |
| Transformer | | 88.97 | 52.31 | 48.69 |
| | √ | 89.92 | 54.56 | 50.21 |
| Adapted-Transformer | | 89.32 | 53.81 | 48.96 |
| | √ | **90.32** | **55.14** | **50.68** |

(a) Performance on all English datasets.

| Context Encoder | Syntactic Information | ON4c | RE | WE |
|---|---|---|---|---|
| Bi-LSTM | | 77.32 | 94.81 | 65.72 |
| | √ | 80.03 | 96.08 | 68.11 |
| Transformer | | 78.18 | 95.26 | 67.16 |
| | √ | 80.46 | 96.31 | 69.24 |
| Adapted-Transformer | | 79.04 | 95.84 | 67.79 |
| | √ | **81.18** | **96.62** | **69.78** |

(b) Performance on all Chinese datasets.

Table 8: Experimental results ($F1$ scores) of our models with and without applying syntactic information (attentive ensemble of all syntactic information) using different types of context encoders.

In the main experiments, we use Adapted-Transformer (Yan et al., 2019) as the context encoder. In this experiment, we try other two popular context encoders, i.e., Bi-LSTM and Transformer (Vaswani et al., 2017), with attentive ensemble of all syntactic information). The results ($F1$) on the test set of all datasets are reported in Table 8, where the corresponding baselines without using syntactic information are also reported for reference. From the results, we find that all models with attentive ensemble to incorporate syntactic information consistently outperform their corresponding baselines across all English and Chinese datasets, which demonstrates the robustness of our approach to leverage syntactic information to improve NER.

## Appendix C: Hyper-parameter Settings

We try different values for hyper-parameters for our model, presented in Table 9. The best values for all hyper-parameters are also reported, which

|  | Values | Best |
|---|---|---|
| Dropout rate | 0, 0.1, 0.2, 0.3 | 0.2 |
| Learning rate | $e^{-5}, e^{-4}, e^{-3}$ | $e^{-4}$ |
| Batch size | 8, 16, 32 | 32 |
| Number of layers | 1, 2, 4 | 2 |
| Number of head | 4, 8, 12 | 12 |
| Hidden units | 64, 128, 256 | 128 |

Table 9: Values tested for different hyper-parameters and the best one used in our experiments.

are obtained by tuning our model with the given hyper-parameter values on the development set of each dataset.

## Appendix D: The Results of Our Models on the Development Set

| $\mathcal{GM}$ | Syn. | | | ON5E | WN16 | WN17 | ON4C | RE | WE |
|---|---|---|---|---|---|---|---|---|---|
| | P. | C. | D. | | | | | | |
| | ✓ | ✓ | | 86.21 | 55.54 | 49.48 | 77.06 | 96.04 | 67.86 |
| ✓ | ✓ | ✓ | | 86.78 | 56.26 | 49.76 | 77.65 | 96.34 | 68.35 |
| | ✓ | | ✓ | 86.41 | 56.31 | 49.69 | 77.31 | 96.12 | 67.63 |
| ✓ | ✓ | | ✓ | 86.84 | 56.84 | 50.02 | 77.56 | 96.31 | 67.86 |
| | | ✓ | ✓ | 86.58 | 56.56 | 49.75 | 77.42 | 96.12 | 67.52 |
| ✓ | | ✓ | ✓ | 86.92 | 57.26 | 50.18 | 77.74 | 96.39 | 68.14 |
| | ✓ | ✓ | ✓ | 86.71 | 56.41 | 50.04 | 77.61 | 96.41 | 68.16 |
| ✓ | ✓ | ✓ | ✓ | **87.03** | **57.38** | **50.51** | **78.05** | **96.46** | **68.92** |

Table 10: $F1$ scores of our models under different configurations on the development set of all datasets. "$\mathcal{GM}$" is the gate mechanism; "P.", "C." and "D." refer to POS labels, syntactic constituents and dependency relations, respectively.

In Table 10, we report the experimental results ($F1$) of our models (i.e., with all types of embeddings and KVMN) under different configurations (using syntax attention on different combinations of syntactic information and whether to use the gate mechanism) on development set of all datasets.