applied sciences

MDPI

Article

A Multi-Granularity Word Fusion Method for Chinese NER

Tong Liu [1], Jian Gao [1], Weijian Ni [1,2,*] and Qingtian Zeng [1]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2] Shandong Key Laboratory of Wisdom Mine Information Technology, Qingdao 266590, China
* Correspondence: niweijian@sdust.edu.cn

Abstract: Named entity recognition (NER) plays a crucial role in many downstream natural language processing (NLP) tasks. It is challenging for Chinese NER because of certain features of Chinese. Recently, large-scaled pre-training language models have been used in Chinese NER. However, since some of the pre-training language models do not use word information or just employ word information of single granularity, the semantic information in sentences could not be fully captured, which affects these models' performance. To fully take advantage of word information and obtain richer semantic information, we propose a multi-granularity word fusion method for Chinese NER. We introduce multi-granularity word information into our model. To make full use of the information, we classify the information into three kinds: strong information, moderate information, and weak information. These kinds of information are encoded by encoders and then integrated with each other through the strong-weak feedback attention mechanism. Specifically, we apply two separate attention networks to word embeddings and N-grams embeddings. Then, the outputs are fused into another attention. In these three attentions, character embeddings are used to be the query of attentions. We call the results the multi-granularity word information. To combine character information and multi-granularity word information, we introduce two fusion strategies for better performance. The process makes our model obtain rich semantic information and reduces word segmentation errors and noise in an explicit way. We design experiments to get our model's best performance by comparing some components. Ablation study is used to verify the effectiveness of each module. The final experiments are conducted on four Chinese NER benchmark datasets and the F1 scores are 81.51% for Ontonotes4.0, 95.47% for MSRA, 95.87% for Resume, and 69.41% for Weibo. The best improvement achieved by the proposed method is 1.37%. Experimental results show that our method outperforms most baselines and achieves the state-of-the-art method in performance.

Keywords: Chinese NER; character word fusion; N-grams; BERT-based model; attention mechanism

check for updates

Citation: Liu, T.; Gao, J.; Ni, W.; Zeng, Q. A Multi-Granularity Word Fusion Method for Chinese NER. *Appl. Sci.* **2023**, *13*, 2789. https://doi.org/10.3390/app13052789

Academic Editor: Dimitris Mourtzis

Received: 21 January 2023
Revised: 8 February 2023
Accepted: 18 February 2023
Published: 21 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Large-scaled pre-training language models, such as BERT [1] and RoBERTa [2], have become a fundamental backbone for various natural language processing (NLP) tasks. Due to the outstanding performance, a lot of work has applied these models to named entity recognition (NER). From the view of obtaining contextual semantics, for example, ERNIE [3] is designed to learn language representation that is strengthened by knowledge-masking techniques, including phrase- and entity-level masking. In order to direct word embedding learning, ERNIE learned intuitively about knowledge and longer semantic dependencies, such as the relationship between entities, the property of an entity, and the type of an event, as opposed to explicitly adding the knowledge embedding. As a result, the model is more generalizable and flexible.

However, there are differences between Chinese and English, such as the way of splitting words, resulting in the NER model not being able to transform easily from English to Chinese. There has been a large amount of work focusing on Chinese NER. For example, by changing the word splitting method, BERT can be applied to Chinese NER. It is possible

to achieve a good result with careful execution. However, this approach suffers from the problem that BERT blocks part of the word-piece segmentation before training. To address this issue, Cui et al. [4] proposed BERT-wwm. Specifically, BERT-wwm introduces the whole word masking strategy for Chinese BERT. It masked the whole word with Chinese text, instead of masking the Chinese character. By doing this, the model shifts its emphasis from focusing solely on the characters to focusing more on the role that words play in a phrase when learning.

We have a deeper perspective on properties of the Chinese language based on these works. We think there are two problems that hinder the Chinese NER from using the English NER model. The first problem is that multi-granularity word information is underutilized. In the Chinese language, the basic elements are characters and words, and they can form phrases. Characters, words, and variously sized phrases make up multi-granularity word information. Character and word information are usually used in Chinese NER models, but phrase information is often ignored. The length of phrases is not fixed, and might carry richer semantic information than the words and phrases. For example, the phrase "Zhu Jian Bu" is the "Ministry of Housing and Urban-Rural Development", which is an organization. "Zhu Jian Bu" consists of three characters, where "Zhu" means "housing", "Jian" means "development", and "Bu" means "ministry". Even though it's an abbreviation, we can easily understand what it means when we see it in the news. However, from the basic information of above three characters, it is difficult for models to recognize this as an organization, because the information they contain is insufficient. Humans associate frequently a number of words when they see characters and words, but the model does not. Based on this intuitive idea, we believe that the model will need to associate additional words, just like people. Therefore, in order to make the model obtain richer semantic information, we hope to introduce multi-granularity information into our model. The second problem is that word information could lead to word segmentation errors and noise. Using "Nan Jing Shi Chang Jiang Da Qiao"("Nanjing Yangtze River Bridge" in English) as an example, the correct segmentation of it is "Nan Jing Shi / Chang Jiang / Da Qiao". However, it is wrong to segment the phrase into "Nan Jing / Shi Zhang / Jiang Da Qiao", and the word segmentation could lead to the error propagation. Furthermore, for the character "Shi", "Nan Jing Shi" ("Nanjing City" in English) and "Shi Zhang" ("Mayor" in English) both contain it, but their semantics are extremely different, resulting in noise. Word segmentation errors and noise could reduce the performance of models. Some models introduce multi-granularity word information [3,5,6], and yet they reduce word segmentation errors and noise through model training, which is implicit. We hope to deal with them in an explicit way.

To address the above problems, we propose a Multi-granularity Word fusion method for Chinese NER (short for MW-NER). We integrate character information and multi-granularity word information into our model, which is to enrich the semantics of sentences. To explicitly deal with word segmentation errors and noise, we design the strong–weak feedback attention mechanism. Specifically, according to semantic information, we firstly classify information into three kinds–strong information, moderate information, and weak information. Since the character information has clear semantics and a fixed size, it is strong. The word information has some errors and less clear semantic information, so it is moderate. The phrase information is weak since it contains rich semantics and various size, and also contains many potential errors. In our model, we use N-grams to represent phrase information. These embeddings of the three kinds of information are generated by the language model. Next, strong information is injected in the strong–weak feedback attention mechanism. The strong–weak feedback attention mechanism integrates strong information into moderate and weak information, and then feeds the moderate and weak information back to strong information. The rich semantic interaction could reduce word segmentation errors and noise in an explicit way. Experimental results show that our method outperforms most baselines and achieves the state-of-the-art methods in performance. (the

method's code is accessible on https://github.com/gaojianchina/mwner (accessed on 8 February 2023))).

Our contributions of this paper can be concluded as follows:

- We introduce multi-granularity word information into the model, and use the strong–weak feedback attention mechanism to integrate this information, which could obtain rich semantic information and reduce word segmentation errors and noise. To make it clear, we classify the multi-granularity word information to strong, moderate and weak information.
- We use the language model to generate character embeddings, word embeddings, and N-grams embeddings, which follows the closure property of the embedding space.
- We conducted numerous experiments, including comparing the effects of various tokenizers, the effects of two fusion strategies, and an ablation study. Finally, we present the experimental results on four Chinese NER benchmark datasets. Experiments show our proposed method outperforms other methods and achieves the performance of the state-of-the-art methods over Chinese NER benchmark datasets.

2. Related Work

2.1. Named Entity Recognition

Collobert [7] presented a CNN-CRF structure that Santos and Guimaraes [8] enhanced with character embeddings. Lample et al. [9] investigated neural structures for NER that combine bidirectional LSTMs with CRFs with features based on character-based word representations and unsupervised word representations. Character CNN was used by Ma and Hovy [10] and Chiu and Nichols [11] to extract features from characters. Recently, large-scaled language model pre-training methods, such as BERT [1] and ELMo [12], have improved NER performance to state-of-the-art levels.

2.2. Pre-Training Language Models in Chinese NER

Large-scaled pre-training in NER has been the subject of a lot of research in recent years. BERT [1], built on the top of the Transformer architecture [13], is pre-trained on a large-scaled unlabeled text corpus in the manner of Masked Language Model and Next Sentence Prediction. Inspired by the masking strategy of BERT [1], ERNIE is designed to learn language representation enhanced by knowledge masking strategies, which includes entity-level masking and phrase-level masking. To mitigate the drawbacks of masking only a part of the whole word, MACBERT could adopt the whole word masking in Chinese to mask the word instead of individual Chinese characters, which is easier for the model to predict. Moreover, Lex-BERT [14] incorporates the lexicon information into Chinese BERT for NER. LEBERT [15] integrates external lexicon knowledge into BERT layers directly by a Lexicon Adapter layer.

2.3. Models with Multi-Granularity Word Information

The goal of using multi-granularity information is to improve character-based models. Lattice LSTM [16] applied a lattice structure for Chinese NER, showing good performance. FLAT [17] improved the results of Chinese NER with a shallow Flat-Lattice Transformer to handle the character-word graph. A-NER [18] is BiGRU-CRF model that combines a self-attention mechanism with multi-embeddings technology. It can extract richer linguistic information of characters from different granularities (e.g., radical, character, word) and find the correlations between characters in the sequence. CWPC_BiAtt [19] put forward a new comprehensive-embedding, considering three aspects, namely character-embedding, word-embedding, and pos-embedding and thus acquired their dependencies, based on which we propose a new Character–Word–Position Combined BiLSTM-Attention for the Chinese NER task. ZEN [5] enhanced Chinese BERT with a multi-layered N-gram encoder but is limited by the small size of the N-gram vocabulary.

Recent works try to integrate the multi-granularity word information and pre-training language models by utilizing their respective strengths. PLTE [20] enhances self-attention

to incorporate lattice structure and introduces a porous mechanism to augment localness modeling while retaining the strength of capturing the rich long-term dependencies. ERNIE [3] exploites entity-level and word-level masking to integrate knowledge into BERT in an implicit way. Moreover, Character and N-gram features are concatenated for joint training. WMSEG [6] uses memory networks to incorporate wordhood information with several popular encoder-decoder combinations for CWS. Multi-metadata Embedding-based Cross-Transformer (MECT) [21] improves the performance of Chinese NER by fusing the structural information of Chinese characters, and then uses multi-metadata embedding in a two-stream Transformer to integrate Chinese character features with radical-level embedding.

Our method is in line with the above approaches. We try to combine multi-granularity information and BERT. The most notable difference between our method and others is in the fusion approaches. Unlike many models that focus on characters (e.g., A-NER and CWPC_BiAtt), we pay more attention to word information and the fusion of word information with character information. Instead of concatenating various information, we design a precise strong–weak feedback attention to accomplish the fusion. It ensures that the output embeddings are of the same length as the original character embeddings, thus avoiding computational pressure.

3. Method

We propose a multi-granularity word fusion method for Chinese NER. In this section, we are presenting the details of the model. The architecture of the model is illustrated in Figure 1.

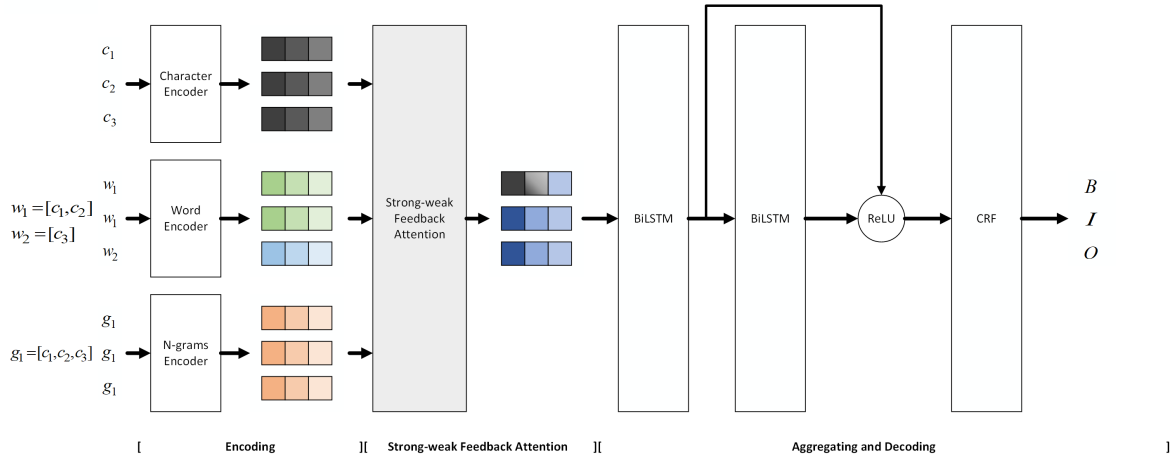


Figure 1. The architecture of MW-NER.

3.1. Encoding

3.1.1. Character Encoder

Characters are the most significant information, and we consider it as strong information since our model is built on characters. The character encoder is used to obtain strong information. We use BERT-wwm [4], a variant of BERT [1], as the model's character encoder. We give a brief explanation here since the basis of BERT has been well explained in the previous studies [1,22].

The pre-training model RoBERTa-wwm-ext-large of Chinese-BERT-wwm (Pre-training models could be obtained from Chinese-BERT-wwm: https://github.com/ymcui/Chinese-BERT-wwm (accessed on 17 April 2022)) is used to obtain an embedding representation of the input information. RoBERTa-wwm-ext-large contains 24-layer Transformers, and each Transformer layer contains 16-head self-attention and 1024 hidden cells. The [CLS] tag is placed at the start of the input sentence s, and the [SEP] tag is placed at the end. For brevity,

we ignore these two labels. The character encoder generates the character embeddings of an input sentence, which we denote as $E^c = [e_1^c, e_2^c, \cdots, e_n^c]$.

3.1.2. Word Encoder

We do not use the embeddings of word2vec [23] or Glove [24], since the number of pre-trained word embeddings (word2vec, Glove, etc.) is limited, and they could not fully cover the tokenized words. Moreover, The embedding space in which the character embeddings are placed differs from the embedding space in which the word embeddings are placed, which affects the model's performance. Instead, our model's character encoder employs the pre-training language model, which can generate embeddings dynamically based on contextual semantics. Especially, we use WoBERT [25]'s method to change Bert's input.

The most notable difference between WoBERT and BERT is that WoBERT's inputs are words instead of characters. The idea is simple but effective. Initially, each sentence is segmented with a tokenizer to obtain $s = [w_1, w_2, \cdots, w_m]$. To keep the shape of word embeddings consistent with that of character embeddings, we use the word embedding to pad the positions of characters in the word. For example, if the word $w_1 = [c_1, c_2]$, we use the embedding of w_1 to pad the position of c_1 and c_2. In this way, the words of the sentence become $s' = [w_1, w_1, w_2, \cdots, w_m]$, and $|s'| = n$. Then, the average of these character embeddings in the word is used as the word embeddings. We should also insert [CLS] and [SEP] labels into the input and yet we ignore these two labels in the subsequent representation. So, the word encoder's output is the embeddings of each word in the sentence, which we denote as $E^w = [e_1^w, \cdots, e_n^w]$.

3.1.3. N-Grams Encoder

Tokenizers might produce word segmentation errors in the word encoder, which would reduce the model's performance. Therefore, we introduce multi-granularity word information to alleviate the impact of this problem. With the addition of multi-granularity word information, the model could learn more word information when it encounters characters.

The sentence is first divided into N-grams of various lengths, and then the dictionary is used to filter out any N-grams that are not words. We denote the filtered N-grams as G. Then we construct the embeddings of the N-grams. For any N-gram: $g_p = [c_i, \cdots, c_j]$, we also use the average of these character embeddings in the N-grams to obtain the embeddings of N-grams in G:

$$e_p^g = \frac{1}{j-i} \sum_{k=i}^{j} e_k^c \tag{1}$$

E^g is used to represent all embeddings in G. We also use an N-gram's embedding to pad the positions of characters in the N-gram. So, the $|e^g| = n$.

So far, we have obtained all three kinds of information.

3.2. Strong–Weak Feedback Attention

The strong–weak feedback attention uses the strong information to guide the moderate information and weak information, and then integrate the moderate information and weak information back to the strong information. The output of it is called multi-granularity word information. The details are shown in Figure 2.

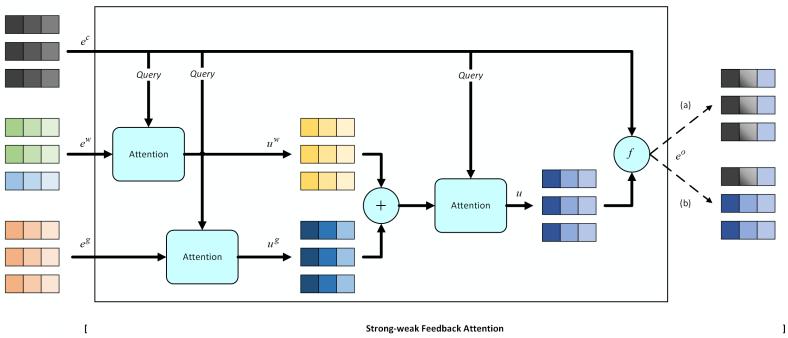


Figure 2. Strong–weak Feedback Attention.

Specifically, we employ two separate attention networks [13]. One is used to combine character and word information, while the other is used to combine character and N-grams information. We denote the output as u^w and u^g, respectively. Character embeddings of one sentence are used to be the query of attentions. It is formulated as follows:

$$\alpha_p^w = \frac{\exp(e^c \cdot e_p^w)}{\sum_{k=i}^{j} \exp(q^w \cdot e_k^c)}, \; u_k^w = \sum_{p=i}^{j} \alpha_j^w e_j^c \tag{2}$$

$$\alpha_p^g = \frac{\exp(e^c \cdot e_p^g)}{\sum_{k=i}^{j} \exp(q^g \cdot e_k^c)}, \; u_k^g = \sum_{p=i}^{j} \alpha_j^g e_j^g. \tag{3}$$

Next, we combine word information and N-grams information, and feed them back to character information, where u_k represents a word embedding ($k \in [i, j]$).

$$\beta_p = \frac{\exp(e^c \cdot (u_p^w + u_p^g))}{\sum_{q=i}^{j} \exp((u_q^w + u_q^g) \cdot e_q^c)}, \; u_k = \sum_{p=i}^{j} \beta_p e_p^G \tag{4}$$

The information-rich embeddings are obtained after the above process, which we denote as $U = \{u_1, u_2, \ldots, u_n\}$.

The steps above generate embedding representations with the multi-granularity information. However, they do not incorporate the representations of these words and N-grams into the characters.

Therefore, to combine character and multi-granularity word information, we introduce two fusion strategies, as shown in the rightmost of Figure 2.

The first fusion strategy, as in Figure 2a, is to link a word's associated word embedding e_k^w to the word's first character e_i^c for a word $w_k = [c_i, \cdots, c_j]$:

$$e_i^o = u_k + e_i^c. \tag{5}$$

And the second, as in Figure 2b, is to add w_k to the character embedding for each character $e_i^c, \cdots, e^j$ in the word:

$$\begin{aligned} e_i^o &= u_k + e_i^c \\ &\cdots \\ e_j^o &= u_k + e_j^c. \end{aligned} \tag{6}$$

We will compare the performance of the two strategies in the comparison of word information fusion strategies experiments.

3.3. Aggregating and Decoding

We design the character–word information aggregator to extract deep semantic information, which is composed of 2 layers of Bi-LSTM. The character–word information aggregator integrates character information with multi-granularity word information. The character's context representation $E^o = [e_1^o, \cdots, e_n^c]$ is fed into the first layer of the Bi-LSTM,

$$\begin{bmatrix} i_t \\ f_t \\ O_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} * \left(WE^o + b\right) \tag{7}$$
$$c_t = \tilde{c}_t \odot f_t,$$
$$h_t^c = O_t \odot tanh(c_t)$$

and we get the hidden state $H^c = [h_1^c, \cdots, h_n^c]$.

Then, we put H^c into the second layer of the Bi-LSTM:

$$\begin{bmatrix} i_t \\ f_t \\ O_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} * \left(WH^c + b\right) \tag{8}$$
$$c_t = \tilde{c}_t \odot f_t,$$
$$h_t^o = O_t \odot tanh(c_t)$$

and obtains the hidden state $H^o = [h_1^o, \cdots, h_n^o]$.

We apply the residual mechanism to process the character hidden state H^c and fusion information H^o to generate the final hidden layer representation H, which alleviates the model's gradient disappearance problem.

$$H = ReLU(ReLU(H^o - H^c) + H^c) \tag{9}$$

To take advantage of dependency between different tags, the Conditional Random field (CRF) was used in all of our models. Given a sequence $s = [c_1, c_2, \cdots, c_n]$, the corresponding golden label sequence is $y = [y_1, y_2, \cdots, y_n]$, and $Y(s)$ represents all valid label sequences. The probability of y is calculated by the following equation:

$$P(y|s) = \frac{\sum_{t=1}^{n} e^{f(y_{t-1}, y_t, s)}}{\sum_{y'}^{Y(s)} \sum_{t=1}^{n} e^{f(y'_{t-1}, y'_t, s)}}, \tag{10}$$

where $f(y_{t-1}, y_t, s)$ computes the transition score from y_{t-1} to y_t and the score for y_t. The optimization target is to maximize $P(y|s)$. When decoding, the Viterbi Algorithm is used to find the path that achieves the maximum probability.

3.4. Pseudo Code

In this section, we give the pseudo code of the process of MW-NER by pseudo code.

The overall pseudo code of the MW-NER is shown in Algorithm 1. First we call the variant of BERT to obtain the embeddings of each character in the sentence. Then, we use the multi-granularity information fusion function to generate embeddings that include multi-granularity word information, with the number of words aligned with the number of characters. (This function will be discussed below.) The multi-granularity embeddings are next fed into the two-layer Bi-LSTM to obtain the word embeddings with temporal information. Since words and characters have one-to-one correspondence in embedding positions, we finally use the residual mechanism to fuse the character information with the word information and obtain the labels via CRF.

Algorithm 1 Multi-granularity information fusion

Input: $s = \{c_1, ..., c_n\}$ is a sequence of characters;
Output: the labels of characters in a sequence t;

$e^c \leftarrow$ BERT(s)
$e^o \leftarrow$ Multi-granularity information fusion(s, e^c)
$h^c \leftarrow$ BiLSTM(e^o);
$h^o \leftarrow$ BiLSTM(h^c);
$h \leftarrow$ ReLU(ReLU(h^o - h^c) + h^c)
$t \leftarrow$ CRF(h)
return t

Algorithm 1 clearly shows where each module is called in the model. In the following, we are going to focus on the execution of the multi-granularity information fusion function. The process of it is shown in Algorithm 2. We briefly describe Algorithm 2. The first step is feeding the embeddings of each character into WoBERT and obtaining the embeddings of words. Then, the sentence is segmented with different granularity to obtain a variety of N-grams. Next, we obtain the representation of words with different granularity by adding the embeddings of each character in the word and take the average value. If a character has many N-grams, these N-grams need to be averaged. The next step is to calculate the weights of words and N-grams on the same character by Attention and the word embeddings after fusion by weight. Finally, the algorithm applies the words or N-grams representation to each character in the word or N-grams respectively and returns the character representation with multi-granularity word information.

Algorithm 2 Multi-granularity information fusion

Input: the sequence of characters $s = \{c_1, ..., c_n\}$; the character embeddings e^c;
Output: the character embeddings of a sequence with multi-granularity word information E^o;

$w \leftarrow$ tokenizer(s)
$g \leftarrow$ N-grams(s)
$e^w \leftarrow$ WoBERT(s)
for c_i in s **do**
    if c_i in w_j **then**
        $e_i^W \leftarrow e_i^w$
    $v^g \leftarrow \frac{1}{j-i} \sum_j^{k=i} e_k^c$
    if c_i in g_j **then**
        $e_i^G \leftarrow v^g$
$\alpha \leftarrow$ Attention(e^c, e^W)
$u^w \leftarrow \sum^n \alpha\, c_i$
$\beta \leftarrow$ Attention(e^c, e^G)
$u^g \leftarrow \sum^n \beta\, c_i$
$u \leftarrow$ Attention(e^c, $u^w + u^g$)
$e^o \leftarrow$ fusion strategy(e^c, u)
return e^o

4. Experiments

4.1. Datasets

In our experiments, we use four Chinese NER benchmark datasets, including Ontonotes4.0 [26], MSRA [27], Resume [16], and Weibo [28,29]. The four datasets are widely used and recognized in Chinese NER. They could ensure the fairness and comparability of experimental results.

- Ontonotes4.0. OntoNotes Release 4.0 was developed as part of the OntoNotes project. It annotates a large corpus comprising various genres of text (news, conversational tele-

phone speech, weblogs, usenet newsgroups, broadcast, talk shows) in three languages (English, Chinese, and Arabic) with structural information and shallow semantics.

- MSRA. The MSRA NER dataset contains 46,364 samples in the training set and 4365 samples in the test set. The original workshop/paper for the dataset is by Levow [27]. The corpus of the Chinese NER dataset MSRA came from the news domain.
- Resume. The Resume dataset is collected from Sina finance (https://finance.sina.com.cn/stock/ (accessed on 17 April 2022)), which consists of resumes of senior executives from listed companies in the Chinese stock market.
- Weibo. The Weibo dataset contains messages selected from Weibo (http://www.weibo.com (accessed on 17 April 2022)). The corpus contains 1890 messages sampled from Weibo between November 2013 and December 2014.

The statistics of these datasets are shown in Table 1.

Table 1. Statistics of datasets.

Datasets	Type	Train	Dev	Test
Ontonotes	Sentence	15.7 k	4.3 k	4.3 k
	Char	491.9 k	200.5 k	208.1 k
MSRA	Sentence	46.4 k	-	4.4 k
	Char	2169.9 k	-	172.6 k
Resume	Sentence	3.8 k	0.46 k	0.48 k
	Char	124.1 k	13.9 k	15.1 k
Weibo	Sentence	1.4 k	0.27 k	0.27 k
	Char	73.8 k	14.5 k	14.8 k

4.2. Model Implementing and Experimental Settings

We use fastNLP (https://github.com/fastnlp/fastNLP (accessed on 17 April 2022)) to implement our model. FastNLP is an NLP framework developed by PyTorch. The implementation of the character encoder depends on Bert in huggingface (https://github.com/huggingface/transformers (accessed on 17 April 2022)), and the BERT-wwm pre-training language model (https://github.com/ymcui/Chinese-BERT-wwm (accessed on 17 April 2022)) is employed. The implementation of the word encoder refers to WoBERT [25]. The dropout rate is 0.5. Adam [30] is used for optimization, with an initial learning rate of 1.5×10^{-5}.

4.3. Metrics

The entity-level metrics, Precision, Recall, and F1 score, are used for evaluation.

Precision is to measure how precise the model is. It is the ratio between true positives and all identified positives (including true positives (TP) and false positives (FP)). The precision metric reveals how many of the predicted entities are correctly labeled.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Recall is to measure the model's ability to predict actual positive classes. It is the ratio between the predicted true positives and what was actually tagged (including true positives (TP) and false negatives (FN)). The recall metric reveals how many of the predicted entities are correct.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

F1 score is a function of Precision and Recall. It is used for a balance between Precision and Recall.

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

4.4. Baselines

To investigate the impact of our suggested method in Chinese NER, we compare it to other representative methods, such as including BiLSTM-CRF [31], Lattice-LSTM [16], FLAT [17], ZEN [17], BERT [1], and RoBERTa [2]. Meanwhile, the PLTE [20] model is chosen for comparison because it is a novel model with currently excellent results. A brief introduction of these models is given as follows.

- BiLSTM-CRF [31]: a classic method, which uses bi-directional LSTM (BiLSTM) to model the language, obtain the left-to-right sentence encoding and right-to-left encoding then use crf to capture the dependencies between tags.
- Lattice LSTM [16]: a lattice-structured LSTM model for Chinese NER, which encodes a sequence of input characters as well as all potential words that match a lexicon.
- FLAT [17]: Flat-LAttice Transformer for Chinese NER converts the lattice structure into a flat structure consisting of spans. Each span corresponds to a character or latent word and its position in the original lattice.
- BERT [1]: BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks.
- RoBERTa [2]: a replication study of BERT pre-training that carefully measures the impact of many key hyper-parameters and training data size.
- ZEN [5]: a BERT-based Chinese text encoder enhanced by N-gram representations. During training, different character combinations are examined, thus potential word or phrase boundaries are explicitly pre-trained and fine-tuned with the character encoder.
- PLTE [20]: PLTE augments self-attention with positional relation representations to incorporate lattice structure, and introduces a porous mechanism to augment localness modeling and maintain the strength of capturing the rich long-term dependencies.
- MECT [21]: MECT uses radical-level embeddings to combine the structural information into Chinese characters, and then uses multi-metadata embeddings in a two-stream Transformer to combine the features of Chinese characters with the radical-level embedding.

4.5. Results

4.5.1. Comparison of Different Tokenizers

In this section, we compare the effects of the model's three tokenizers: jieba (https://github.com/fxsjy/jieba (accessed on 17 April 2022)), jiagu (https://github.com/ownthink/Jiagu (accessed on 17 April 2022)), and fastHan [32], in order to find the one that best fits our model and performs the best in them. The experiments are carried out on the Resume dataset, and the model settings remain consistent with the exception of the tokenizer.

According to the experimental results on 10 epochs, the best result of fastHan is 95.87%, which is higher than 95.71% of jieba and 95.34% of jiagu. Figure 3 depicts the complete results for the ten epochs. We then compute the average values over these ten epochs, which are 94.57% for fastHan, 94.53% for jieba, and 93.97% for jiagu, with fastHan being the highest. As a result, we think fastHan fits our model better than jieba and jiagu, so it will be employed as the tokenizer in future experiments.

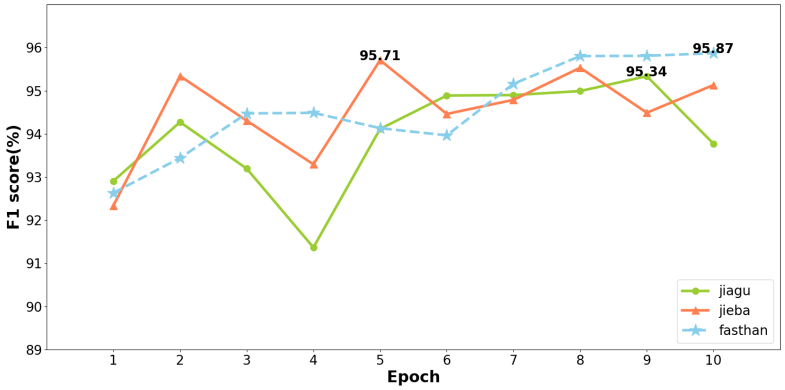


Figure 3. F1 scores of Different Tokenizers on Resume (%).

4.5.2. Comparison of Character and Multi-Granularity Word Information Fusion Strategies

We introduced two strategies of word fusion: (a) integrate word information into the first character of a word, or (b) integrate word information into all characters in a word, which are shown in the rightmost of Figure 2. We devise an experiment to compare the two strategies.

The experimental results are shown in Figure 4. The data show that the first strategy outperforms the second. we suppose the first fusion strategy emphasizes the importance of the initial character of words or N-grams. As a result, when labeling the entity's first character, the model could detect a clear start position, making the entity easier to identify. So, we will employ the first fusion strategy in future experiments.

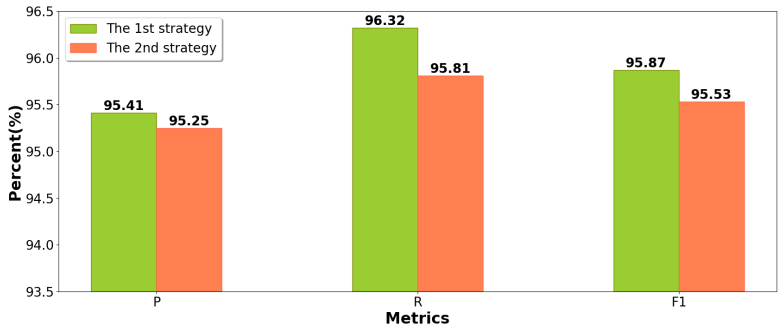


Figure 4. F1 scores of Different Fusion Strategies on Resume (%).

4.5.3. Comparison with baselines

Table 2 shows the experimental results (F1 scores) of the baselines and our method on the four datasets.

Table 2. The performance of methods on four datasets (%).

Models	Ontonotes4.0	MSRA	Resume	Weibo
BiLSTM-CRF	71.81	91.87	94.41	56.75
Lattice LSTM	73.88	93.18	94.46	58.79
FLAT	76.45	94.35	95.45	63.42
MECT	76.92	94.32	95.89	63.30
BERT	80.14	94.95	94.87	68.20
RoBERTa	81.39	95.38	95.31	68.35
ZEN	-	95.25	-	-
PLTE	80.60	94.53	96.45	69.23
MW-NER	81.51	95.47	95.87	69.41

Our model outperforms both BiLSTM-CRF and Lattice LSTM in terms of performance. On the Resume, Ontonotes, and MSRA datasets, our model outperforms FLAT by 0.42%, 5.06%, and 1.12%, respectively. The improvement on the Weibo dataset is even greater, reaching 5.99%. And when compared to MECT, our model trails by 0.02 percent on the Resume dataset while leading on the other datasets. This could be due to the benefits of multi-layer Transformers and the evolution of multi-granularity word information fusion. In comparison with Bert and RoBERTa, our model improves on each dataset, with the biggest gap reaching 1.37%. This demonstrates that combining moderate and weak information could indeed enhance the performance of our model.

Compared with PLTE, a novel model with currently excellent results, our model's F1 scores on the Ontonotes, MSRA, and Weibo datasets are higher. However, there is still room for improvement in the Resume dataset, as our model's F1 score lags behind that of PLTE. In the Resume dataset, we suspect our model is inferior to PLTE for the following reasons. PLTE introduces a porous mechanism to augment localness modeling and maintain the strength of capturing the rich long-term dependencies. Our model introduces richer word information, and uses Bi-LSTM to model long-term dependencies. However, the effect of Bi-LSTM may be inferior to that of the porous mechanism. Furthermore, due to the small size of the Resume dataset, our model could not be fully trained.

According to experiments on Chinese NER benchmark datasets, our method outperforms most baselines and achieves the state-of-the-art performance. We consider that there are two main reasons for the excellent results of our model. On the one hand, our model is inherently based on BERT, which makes the character embeddings richer and more dynamic to learn appropriate semantics. On the other hand, our model focuses on word information, and uses multi-granularity word information back to guide character information. In other words, the strong–weak feedback attention mechanism plays a crucial role. Due to powerful character information and richer word information, our model presents better performance. However, some modules of our model might not take full advantage of the parallel power of the GPU. In the strong–weak feedback attention mechanism, the advantage of parallel computing may not be fully utilized because one character in a sentence might be fused with many words or phrases in corresponding positions. So, the computational speed of the model is impaired. We are aware of this issue, and will study it in our future work.

4.6. Ablation Study

We devise ablation experiments to evaluate the effectiveness of the MW-NER model's two modules: N-grams aggregator and word encoder. The following are the settings for ablation experiments:

- MW-NER. MW-NER is the proposed method, as depicted in Figure 1.
- MW-NER without N-grams aggregator (w/o NA). The N-grams aggregator of MW-NER is removed. There is only character and word information in the model, but no multi-granularity N-grams information.
- MW-NER without Word encoder (w/o WE). The word encoder is further removed. In this case, the model has degenerated into a character model.

The N-grams aggregator module requires some hyper-parameters to be adjusted, i.e., utilizing different combinations of N-grams, among the experiments. The Resume dataset is still chosen as the experimental dataset. The other settings remain the same as in the previous experiments.

In Figure 5, the numbers in parentheses indicate different combinations of N-gram in the N-grams aggregator module. For example, "MW-NER(2,3)" indicates that MW-NER uses the 2-gram and 3-gram in the N-grams aggregator module. The experimental results could be divided into three groups, and the first group has multiple bars (filled with green) in the figure due to different hyper-parameters. By removing the N-grams aggregator, the model's best result falls from 95.87% to 95.41%, which demonstrates that the N-grams aggregator could improve the model's comprehension of phrases and sentences

by enriching semantics. When the word encoder is removed from the model, performance suffers even more, illustrating the importance of incorporating word embeddings into the model.

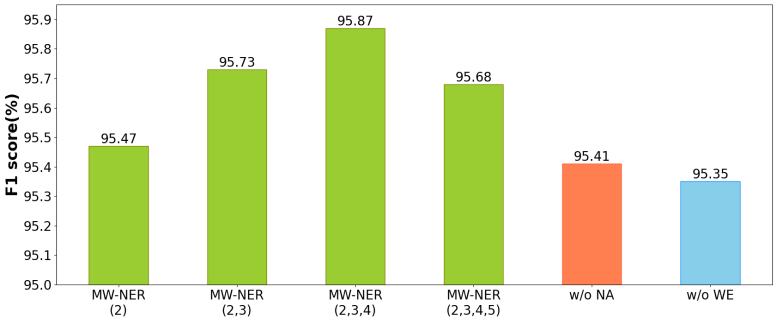


Figure 5. F1 scores of different models on Resume (%).

We further analyze the parameters used by the N-grams aggregator. When comparing the models using different N-grams information, as shown in the first four bars in Figure 5, we discover that as the number of N-grams increases, the model's performance does not improve. On the contrary, the model based on 2,3,4-grams outperforms the model based on 2,3,4,5-grams. This could be explained by the fact that the positive gain introduced by 5-grams is less than the negative gain introduced by it. Some official data supports this explanation. According to statistics, there are around 120,000 words in Chinese dictionaries, with over 80,000 two-character words, 20,000 three-character words, 20,000 four-character words, and only about 2500 five-character words. When we employ 5-grams, we run the risk of introducing a large number of errors, lowering the model's performance.

5. Conclusions and Future Work

We propose a multi-granularity word fusion method for Chinese NER. To make full use of character and word information, we classify the information into strong, moderate, and weak information. The strong-weak feedback attention mechanism integrates these kinds of information with each other and obtains rich semantic information and reduces word segmentation errors and noise. Moreover, we carefully construct two strategies to integrate the character information and multi-granularity word information. Experiments show that our method outperforms most baselines and achieves the state-of-the-art performance. However, the model might encounter parallelism trouble when characters need to be combined with many words. For example, since splitting the sentence to get the start and end positions of the words, the model might not take full advantage of parallel computing, which in turn affects the later steps. We will optimize the shortcomings of the model in future work.

Author Contributions: Methodology, W.N.; Writing—original draft, J.G.; Writing—review & editing, T.L. and Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grants 71704096, in part by Shandong Provincial Natural Science Foundation under Grant ZR2022MF319, in part by the Qingdao Philosophy and Social Sciences Planning Project under Grant QDSKL1801122 and QDSKL2001117, and by the Talented Young Teachers Training Program of Shandong University of Science and Technology under Grant BJ20211110.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Four Chinese NER benchmark datasets are used in experiments, including Ontonotes4.0 (https://catalog.ldc.upenn.edu/LDC2011T03 (accessed on 17 April 2022)), MSRA (https://www.microsoft.com/en-us/download/details.aspx?id=52531 (accessed on 17 April 2022)), Resume (https://github.com/jiesutd/LatticeLSTM/tree/master/ResumeNER (accessed on 17 April 2022)),

and Weibo (https://github.com/OYE93/Chinese-NLP-Corpus/tree/master/NER/Weibo (accessed on 17 April 2022)). They are open resources.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
2. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
3. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv* **2019**, arXiv:1904.09223.
4. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3504–3514. [CrossRef]
5. Diao, S.; Bai, J.; Song, Y.; Zhang, T.; Wang, Y. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Stroudsburg, PA, USA, 16–20 November 2020; pp. 4729–4740. [CrossRef]
6. Tian, Y.; Song, Y.; Xia, F.; Zhang, T.; Wang, Y. Improving Chinese Word Segmentation with Wordhood Memory Networks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8274–8285. [CrossRef]
7. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
8. Santos, C.N.d.; Guimaraes, V. Boosting named entity recognition with neural character embeddings. *arXiv* **2015**, arXiv:1505.05008.
9. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 260–270. [CrossRef]
10. Ma, X.; Hovy, E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1064–1074. [CrossRef]
11. Chiu, J.P.; Nichols, E. Named Entity Recognition with Bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
12. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237. [CrossRef]
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
14. Zhu, W.; Cheung, D. Lex-BERT: Enhancing BERT based NER with lexicons. *arXiv* **2021**, arXiv:2101.00396.
15. Liu, W.; Fu, X.; Zhang, Y.; Xiao, W. 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), Online, 1–6 August 2021; pp. 5847–5858. [CrossRef]
16. Zhang, Y.; Yang, J. Chinese NER Using Lattice LSTM. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1554–1564. [CrossRef]
17. Li, X.; Yan, H.; Qiu, X.; Huang, X. FLAT: Chinese NER Using Flat-Lattice Transformer. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6836–6842. [CrossRef]
18. Song, C.; Xiong, Y.; Huang, W.; Ma, L. Joint Self-Attention and Multi-Embeddings for Chinese Named Entity Recognition. In Proceedings of the 2020 6th International Conference on Big Data Computing and Communications (BIGCOM), Deqing, China, 24–25 July 2020; pp. 76–80. [CrossRef]
19. Johnson, S.; Shen, S.; Liu, Y. CWPC_BiAtt: Character–Word–Position Combined BiLSTM-Attention for Chinese Named Entity Recognition. *Information* **2020**, *11*, 45. [CrossRef]
20. Mengge, X.; Yu, B.; Liu, T.; Zhang, Y.; Meng, E.; Wang, B. Porous Lattice Transformer Encoder for Chinese NER. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 3831–3841. [CrossRef]
21. Wu, S.; Song, X.; Feng, Z. 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, Online, 1–6 August 2021; pp. 1529–1539. [CrossRef]

22. Yu, J.; Jiang, J. Adapting BERT for Target-Oriented Multimodal Sentiment Classification. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 5408–5414. [CrossRef]
23. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013.
24. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
25. Su, J. WoBERT: Word-Based Chinese BERT Model—ZhuiyiAI. Technical Report. 2020. Available online: https://zhuiyi.ai/ (accessed on 17 April 2022).
26. Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; et al. *OntoNotes Release 4.0*; Linguistic Data Consortium: Philadelphia, PA, USA, 2011. [CrossRef]
27. Levow, G.A. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Sydney, Australia, 22–23 July 2006; pp. 108–117.
28. Peng, N.; Dredze, M. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 548–554. [CrossRef]
29. He, H.; Sun, X. 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, Valencia, Spain, 3–7 April 2017; pp. 713–718. [CrossRef]
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
32. Geng, Z.; Yan, H.; Qiu, X.; Huang, X. fastHan: A BERT-based Multi-Task Toolkit for Chinese NLP. *arXiv* **2020**, arXiv:2009.08633.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.