electronics

MDPI

Article

Robust Chinese Named Entity Recognition Based on Fusion Graph Embedding

Xuhui Song [1], Hongtao Yu [2,*], Shaomei Li [2] and Huansha Wang [2]

[1] School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, China
[2] National Digital Switching System Engineering & Technological R&D Center, PLA Information Engineering University, Zhengzhou 450002, China
* Correspondence: yht_ndsc@139.com

Abstract: Named entity recognition is an important basic task in the field of natural language processing. The current mainstream named entity recognition methods are mainly based on the deep neural network model. The vulnerability of the deep neural network itself leads to a significant decline in the accuracy of named entity recognition when there is adversarial text in the text. In order to improve the robustness of named entity recognition under adversarial conditions, this paper proposes a Chinese named entity recognition model based on fusion graph embedding. Firstly, the model encodes and represents the phonetic and glyph information of the input text through graph learning and integrates above-multimodal knowledge into the model, thus enhancing the robustness of the model. Secondly, we use the Bi-LSTM to further obtain the context information of the text. Finally, conditional random field is used to decode and label entities. The experimental results on OntoNotes4.0, MSRA, Weibo, and Resume datasets show that the F1 values of this model increased by 3.76%, 3.93%, 4.16%, and 6.49%, respectively, in the presence of adversarial text, which verifies the effectiveness of this model.

Keywords: natural language processing; Chinese named entity recognition; graph embedding; adversarial text; model robustness

check for updates

Citation: Song, X.; Yu, H.; Li, S.; Wang, H. Robust Chinese Named Entity Recognition Based on Fusion Graph Embedding. *Electronics* **2023**, *12*, 569. https://doi.org/10.3390/electronics12030569

Academic Editor: Arkaitz Zubiaga

Received: 9 December 2022
Revised: 18 January 2023
Accepted: 19 January 2023
Published: 22 January 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

Named entity recognition (NER) task refers to analyzing natural language texts based on computer technology to identify entities with specific meanings, such as person, place, organization, proper nouns, etc. It is a basic task in the field of natural language processing and is of great significance to downstream tasks, such as relationship extraction [1], event extraction, syntactic analysis, and machine question answering [2].

Early named entity recognition models were mainly implemented based on rules, dictionaries, and traditional machine learning methods, but such models relied too much on feature engineering so that these had low training and verification efficiency. With the development of computer computing power and deep learning technology, most of the current mainstream named entity recognition methods are all based on a deep neural network model. However, previous studies have shown that models based on deep neural networks are vulnerable to adversarial text attacks [3]. With the large-scale emergence of public online forums and social software, users often use homophones, words with similar glyphs, and other methods to replace text for the purpose of avoiding detection. These processed texts are called adversarial texts. For example, use "崴心" to replace the text to express the meaning of "微信". Under adversarial conditions, the traditional named entity recognition model based on deep learning is difficult to accurately identify entities in the texts due to its own vulnerability, which affects the prediction effect of the model. Therefore, this paper mainly studies how to improve the robustness of the deep learning model to improve the recognition accuracy of named entities containing adversarial text.

In order to solve the robustness problem of named entity recognition models based on deep learning, the mainstream named entity recognition models try to rely on the idea of text adversarial defenses, that is, to improve robustness through spelling correction and model enhancement. Among them, the spelling correction method is used to identify the difference between the adversarial examples and the original examples by comparing them, and to deal with the spelling errors in the text through specific spelling error check to achieve the purpose of defense. For example, Gong et al. [4] propose a spelling error correction method based on word embedding, which first detects the spelling errors in the text and constructs a set of valid candidate words similar to the misspelled words, and then selects the optimal candidate words for spelling correction according to the context information. Alshmeli et al. [5] propose a spelling correction method combining editing distance, frequency counting, and contextual similarity techniques. Liu et al. [6] propose a Chinese spelling correction method using pre-trained models. Xu et al. [7] propose spelling correction using multimodal information of Chinese characters. Liu et al. [8] propose a Chinese spelling correction model on multi-typo texts, which first generates a noisy context for each training instance, and then forces the correction model to produce similar outputs based on the original and noisy contexts. The spelling correction method requires the error text in the text to be restored to the correct text, but errors in the model error correction process can cause errors in the model's predictions. The model enhancement method includes two methods: adversarial training and model structure improvement. The adversarial training method mixes the adversarial examples and the original examples for re-training to improve the robustness of the model. For example, Wang et al. [9] retrain the machine understanding model using adversarial samples generated by adversarial attacks, and the results show that the adversarial training method is effective in improving the robustness of the model. Liu et al. [10] proposed that the joint use of character embedding and adversarial stability training framework can effectively defend against character-level adversarial samples. Dong et al. [11] replace the original word vectors with weighted combinations of word vectors of synonyms, and then optimize the weights by gradient optimization to create virtual adversarial examples in the embedding space for adversarial training. Ou et al. [12] proposes a Chinese adversarial examples generation approach with a multi-strategy based on semantic, which combines five strategies, including synonymous words, similar words of form, similar words of sound, pinyin rewriting, and phrase disassembly, to replace important words in sentences to improve the quality of Chinese adversarial samples. However, the defense method of adversarial training needs to know the attacker's attack method during the retraining process; otherwise, the generated adversarial text is difficult to fight against unknown adversarial forms. At the same time, the quality of adversarial samples has a great influence on the effect of the model, because Chinese has more than 20,000 Chinese characters, and each Chinese character has a variety of deformation relationships, such as phonetic and glyph proximity, so that it is difficult for adversarial samples to enumerate all the deformation cases so that the model can be fully trained. Model structure improvement refers to directly modifying the backbone structure of the model, such as Jones et al. [13], to improve the robustness of the model by building a vocabulary list of the most frequent words in a text, clustering words in the dictionary, and sharing the same coding method for similar words. Yang et al. [14] propose a robust adversarial training method called fast triplet metric learning, which attempts to cluster similar embeddings and does not establish a relationship with nonsynonym clustering, ultimately minimizing the distance between synonyms and maximizing the distance between nonsynonyms to increase robustness. However, this method of synonym sharing embedding has certain limitations as a defense method for English scenarios, and Chinese adversarial text attacks are usually replaced based on Chinese characters, so it is difficult to directly migrate to the Chinese named entity recognition model.

To sum up, in order to further improve the robustness of the Chinese named entity recognition model, inspired by the characteristics that the generation of adversarial text will destroy the original characters' phonetic and glyph, this paper attempts to model the

knowledge of character phonetic glyph confrontation of Chinese characters by constructing an undirected graph and proposes a Chinese named entity recognition model that incorporates graph embedding. The model first uses the homophone and glyph relationships to build an undirected graph, then uses the node2vec algorithm [15] to obtain the graph embedding representation of words, and integrates it with semantic embedding, so as to improve the robustness of the named entity recognition model. The model is evaluated on four public datasets (OntoNotes4.0, MSRA, Resume, and Weibo). The experimental results show that the model can improve the robustness of the named entity recognition model and without a sacrifice of performance on normal text.

The main contributions of this paper are as follows: The model uses the phonetic glyph relationship of Chinese characters to construct a phonetic glyph undirected graph. In addition, this paper proposed to integrate the graph embedding representation of Chinese characters in the semantic embedding layer, so as to integrate the knowledge of the character phonetic glyph confrontation of Chinese characters into the embedding representation, so as to improve the robustness of the Chinese named entity recognition model.

2. Robust Chinese Named Entity Recognition Based on Fusion Graph Embedding

In order to improve the robustness of named entity recognition model, this paper proposes a Chinese named entity recognition model based on fusion graph embedding. The model is mainly composed of semantic embedding layer, context embedding layer, and decoding layer. The specific model structure is shown in Figure 1. Firstly, the model constructs an undirected graph of the relationship between homophone and glyph and then obtains the graph embedding representation of characters through node2vec random walk algorithm, which is fused with semantic embedding as the final encoded representation of characters. Secondly, feature extraction is carried out through the Bi-LSTM model to obtain the context information of the text. Finally, the conditional random field is used to output the label sequence with the largest probability value to complete the labeling of the entity.

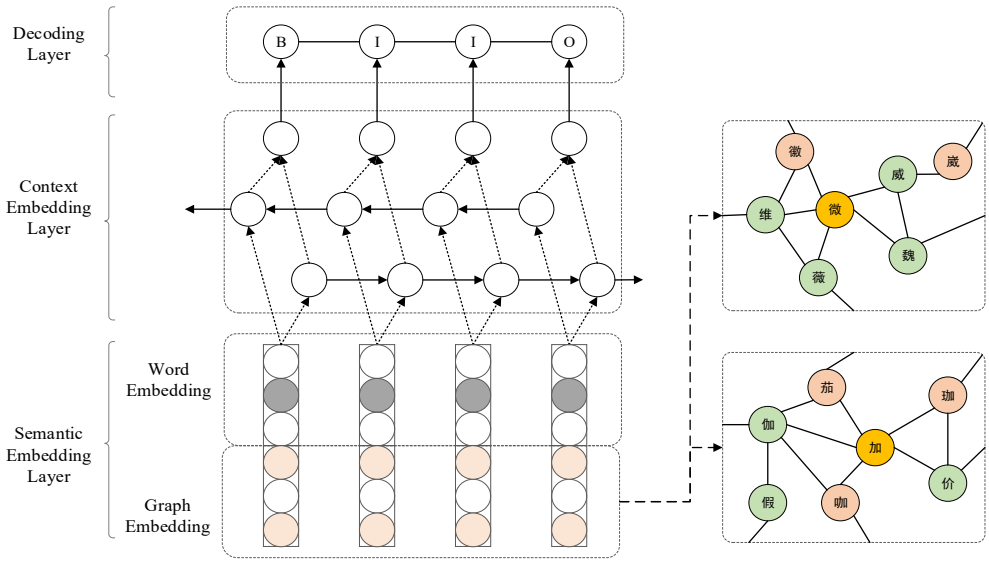


Figure 1. Robust named entity recognition model structure. The Chinese in the figure aims to express the relationship between Chinese characters.

2.1. Semantic Embedding Layer

The semantic embedding layer of the model is responsible for encoding and representing each character in the sentence. The traditional semantic embedding layer usually uses word2vec algorithm [16] to obtain the semantic embedding representation of characters. However, this method does not make full use of the information, such as the phonetic and

glyph of characters. When there is an adversarial text in the text, the recognition effect of the model will be affected. Considering that the adversarial text is usually replaced based on the phonetic and glyph similarity of Chinese characters, the model obtains the graph embedding representation of characters by constructing an undirected graph and using the graph learning method and integrates the adversarial relationship of characters, such as characters' phonetic and glyph, into the embedding representation, thereby improving the model's ability to recognize adversarial text. Firstly, an undirected graph of the relationship between phonetics and glyphs is constructed by using the similar relationship between phonetic and glyph. Secondly, the node2vec algorithm is used to represent the adversarial relationship between characters as the graph embedding representation of characters, and the skip-gram model is used to learn the semantic embedding representation of characters. Node2vec algorithm is an algorithm for graph node representation learning that combines two random walking modes: Depth-first search (DFS) and breadth-first search (BFS), through which the phonetic and glyph relationship in the graph is learned. Finally, the graph embedding and semantic embedding are fused as the final embedding representation and input into the context embedding layer.

2.1.1. Construction of the Undirected Graph of Characters' Phonetic and Glyph

Considering that adversarial texts are usually disturbed based on the relationship between characters' phonetic and glyph, the model first encodes the adversarial relationship by constructing an undirected graph of the relationship between characters' phonetic and glyph, which is recorded as $G = (V, E)$, where V represents the nodes set in the graph, including all characters; E represents the edges set, representing the relationship between characters. If the character i and the character j have a phonetic or glyph relationship, assign the connecting edge of node i and node j in the figure as 1, and the final undirected figure is shown in Figure 2.

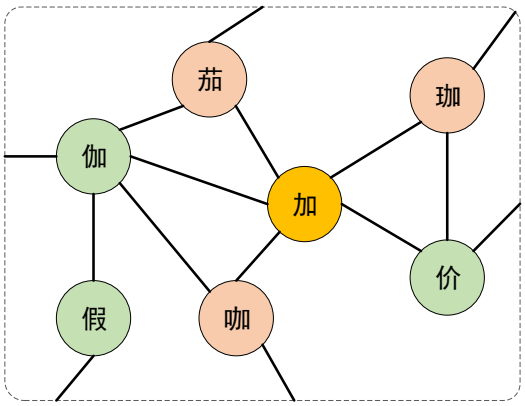


Figure 2. The undirected graph of characters' phonetic and glyph. The Chinese in the figure aims to express the relationship between Chinese characters.

In order to obtain the homophone relationship between nodes in the undirected graph, the model first converts Chinese characters into their corresponding pinyin forms through Python's pinyin processing library. Secondly, the pinyin Chinese characters with the same pinyin are stored in the dictionary as key value pairs. Finally, all the Chinese characters with the same pinyin form a connecting edge to complete the construction of the homophone relationship in the undirected graph.

For glyph similarity relationship, the model first converts it into the corresponding gray image through Python's image processing library cv2 and extracts the image features through a convolution neural network [17]. Secondly, the similarity between characters is calculated to measure the shape similarity between different characters, and the glyph similarity between the top ten characters and the character is established. Finally, a line is

formed between all the characters with glyph relationship to complete the construction of the glyph relationship in the undirected graph.

2.1.2. Graph Embedding Method Based on node2vec Algorithm

Aiming at the undirected graph of the relationship between characters' phonetic and glyph built above, the model uses the node2vec algorithm to learn the adversarial relationship between characters in the undirected graph, so as to integrate the adversarial knowledge of characters' phonetic and glyph into the representation and serve as the final graph embedding representation of characters, so as to improve the recognition effect of the model when the input contains adversarial text.

Node2vec algorithm is a graph embedding method that comprehensively considers the depth-first search (DFS) neighborhood and the breadth-first search (BFS) neighborhood. The principle of the algorithm is to first obtain a random walk sequence through a fixed-point sequence sampling strategy and then train the random walk sequence generated by all nodes through a skip-gram model to map the nodes in the sequence into a vector representation. First, obtain the nearest neighbor sequence of vertices by biased random walk. For a given vertex v, the probability of accessing the next vertex x is [15]:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & if (v, x) \in E \\ 0, & otherwise \end{cases} \quad (1)$$

where π_{vx} represents the transition probability between vertex v and vertex x, and Z represents the normalization constant.

The simple random walk method takes the edge weight w_{vx} as the transition probability, that is, $\pi_{vx} = w_{vx}$. Node2vec algorithm controls the strategy of random walk by defining two parameters p and q, which consider both depth-first search and breadth-first search. Suppose that the current random walk just passes through the edge (t, v) and reaches the vertex v, calculate the transition probability π_{vx} of the connected edge (v, x) corresponding to the vertex v to determine the next walk vertex. Let $\pi_{vx} = \alpha_{pq}(t, x) \cdot \omega_{vx}$, where ω_{vx} is the weight of the edge between the vertex v and the vertex x. $\alpha_{pq}(t, x)$ is calculated as [15]:

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p}, & if\ d_{tx} = 0 \\ 1, & if\ d_{tx} = 1 \\ \frac{1}{q}, & if\ d_{tx} = 2 \end{cases} \quad (2)$$

where d_{tx} represents the shortest path distance between vertex t and vertex x.

The model applies the node2vec algorithm to the generation process of graph embedding; the BFS can model the directly connected phonetic and glyph relationship of each vertex in the undirected graph generated in Section 2.1.1, and DFS can establish an adversarial relationship for Chinese characters indirectly connected by vertices and can model more complex adversarial relationships. After the random walk sequence of text input is obtained by the random walk method above, the model is trained by the skip-gram model to obtain the embedding representation of the node in the graph as the graph embedding representation of the Chinese characters contained in the node, which is denoted as e_g.

2.1.3. Semantic Embedding

Word2Vec is a vector representation method commonly used in natural language processing tasks. This method learns from a large number of text predictions in an unsupervised way, so as to obtain the word vector representation representing the semantic information of words. The Word2Vec model includes the CBOW model and the skip-gram model. This model uses the skip-gram model to obtain the semantic embedding representation of characters, which is denoted as e_s.

After obtaining the graph embedding representation and semantic embedding, the model fuses the graph embedding representation (e_g) of the fusion adversarial relation-

ship with the semantic embedding representation (e_s) as the semantic embedding representation containing the knowledge of the adversarial relationship:

$$e_i = [e_{gi},\ e_{si}] \tag{3}$$

where e_{gi} is the graph embedding representation of characters, e_{si} is the semantic embedding representation of characters, and e_i is the final semantic representation of fusion adversarial relations.

2.2. Context Embedding Layer

The context encoding layer aims to extract the context features of text. Recurrent neural network is widely used in natural language processing tasks because it can model text sequences with context semantics well. However, if the sequence length is too long, the gradient may vanish or explode. Long short-term memory (LSTM) alleviates the problem of gradient vanishing to some extent by introducing the gating mechanism. In order to further enhance the effect of feature extraction, this paper applies a bi-directional long short-term memory network (Bi-LSTM) [18] to further feature extraction of the semantic embedding obtained above, so as to obtain the coded representation containing context semantic relations. The specific calculation process of LSTM is as follows [18]:

$$\begin{gathered} f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \\ o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \\ \widetilde{C}_t = \tan h(W_c * [h_{t-1}, x_t] + b_c) \\ C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \\ h_t = o_t * \tan \mathrm{h}(C_t) \end{gathered} \tag{4}$$

where f_t represents the state of forget gate, i_t represents the state of input gate, which is used to determine the new information stored in the cell state, o_t represents the state of output gate, h_{t-1} represents the state of the last hidden layer, c_t represents the updated cell state, and h_t represents the hidden layer output state of the current cell.

In this paper, Bi-LSTM is used for feature coding; two feature representations are obtained through forward and backward calculation and are spliced as the final feature representation. The specific process is as follows:

$$\begin{gathered} \overrightarrow{h_t} = \overrightarrow{LSTM}(x_i, \overrightarrow{h_{t-1}}) \\ \overleftarrow{h_t} = \overleftarrow{LSTM}(x_i, \overleftarrow{h_{t-1}}) \\ h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \end{gathered} \tag{5}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent forward and backward feature representation, respectively.

2.3. Decoding Layer

The decoding layer aims to decode the output sequence of the encoding layer to obtain the labeled sequence. The Softmax classifier in the traditional model outputs the label with the highest corresponding probability of each word for the input sequence through a greedy method to obtain its corresponding label sequence. However, this method ignores the temporal dependency between adjacent labels and obtains only the locally optimal labeled sequence.

The Chinese named entity recognition model based on fusion graph embedding proposed in this paper uses the current mainstream conditional random field (CRF) [19] as the decoding layer. This method uses the Viterbi algorithm to capture the dependencies between consecutive tags to obtain the global optimal tag sequence. For the output sequence

$R = \{r_1, r_2, ..., r_n\}$ of the context encoding module, the matching score of the given input and output is:

$$score(R, y) = \sum_{i,yi} E_{i,yi} + \sum_{i} T_{yi,yi+1} \quad (6)$$

where R is the input sequence, y is the predicted label sequence, T is the label transition matrix, and E_i, y_i represent the probability that the i_{th} word in the sentence is labeled y_i.

If the real label corresponding to sequence R is $Y = \{y_1, y_2, ..., y_n\}$, then the probability of labeled sequence y is:

$$P(y|x) = \frac{e^{score(R,y)}}{\sum_{y'} e^{score(R,y')}} \quad (7)$$

During model training, L2 regularization is used to minimize the log likelihood loss to optimize the model. The loss function is defined as:

$$L = -\sum_{i=1}^{N} \log(P(y_i|s_i)) + \frac{y}{2}||\theta||^2 \quad (8)$$

where γ is the regularization parameter of L2, θ is the all-trainable parameters set [20].

3. Experiment

3.1. Datasets

The four datasets, MSRA [21], OntoNotes4.0 [22], Resume [23], and Weibo [24], are common Chinese datasets for Chinese named entity recognition tasks. Among them, MSRA and OntoNotes 4.0 are news datasets, and Resume and Weibo are resumes and social media datasets, respectively. These four commonly used Chinese datasets were selected for model training and validation, which made the scale of the datasets and the corpus source more comprehensive and the evaluation results more objective. The details of each dataset are shown in Table 1.

Table 1. Detail statistics of datasets.

Dataset	Data Type	Train Set	Validation Set	Test Set
MSRA	Sentence	46.4k	-	4.4k
	Character	2169.9k	-	172.6k
OntoNotes4.0	Sentence	15.7k	4.3k	4.3k
	Character	491.95k	200.5k	208.1k
Resume	Sentence	3.8k	0.46k	0.48k
	Character	124.1k	13.9k	15.1k
Weibo	Sentence	1.4k	0.27k	0.27k
	Character	73.8k	14.5k	14.8k

In order to verify the effectiveness of the proposed method, the four public datasets are processed in a replacement way to simulate the named entity recognition scenario with adversarial text. Specifically, adversarial text generation is generated on a sentence-by-sentence basis for text in a dataset, and the attack rate is $\gamma = 0.15$ to replace the words in a sentence. Previous studies have shown that attacking information words in a sentence is more effective than attacking random words [25]. Therefore, based on the importance of words in sentences, this paper gives priority to the named entity part of a sentence for adversarial attack. If the labels corresponding to words in a sentence are non-entity labels, then the text in the sentence is randomly selected. For the selected text, replace it with its homophone or glyph-similar character to generate an adversarial text. In this paper, experiments were carried out on the original dataset and the processed dataset, respectively.

3.2. Marking Specifications and Evaluation Indicators

This experiment adopts the BIOS annotation method, where B represents the start position of the entity, I represents the middle position and end position of the entity, S represents the single-word entity, and O represents the rest of the non-entity parts.

Precision, Recall, and F1 values are selected as evaluation indicators in the experiment. The specific formula is as follows:

$$\begin{aligned} P &= \frac{T_p}{T_p+F_p} \times 100\% \\ R &= \frac{T_p}{T_p+F_n} \times 100\% \\ F_1 &= \frac{2PR}{P+R} \times 100\% \end{aligned} \tag{9}$$

where T_p is the number of entities marked correctly by the model, F_p is the number of entities marked incorrectly, and F_n is the number of entities not marked.

3.3. Experimental Parameter Setting

In the experiment, the paddlepaddle deep learning framework is used for model training and verification. The dimensions of word vector and graph embedding vector are 128, and the hidden layer dimension of Bi-LSTM is 128. Adam algorithm [26] is used as the optimizer to optimize parameters. The super parameter values set in the experiment are shown in Table 2.

Table 2. Experimental hyperparameter settings.

Experimental Environment	Setting
Word vector dimension	128
Hidden layer dimension of Bi-LSTM	128
Initial learning rate	0.001
Training epochs	200

3.4. Experimental Results and Analysis

In order to verify the effectiveness of incorporating graph embedding representation into the semantic embedding layer of the model for improving the robustness of Chinese named entity recognition, this paper first carried out experiments on the processed datasets containing adversarial text and compared the results with the original model and the spelling correction model to verify the effectiveness of the proposed method in improving the robustness of the model. Secondly, we conducted comparative experiments on four original public datasets to verify that the method in this paper will not have a negative impact on normal text. In the following, the adversarial scenario and the non-adversarial scenario are analyzed, respectively.

3.4.1. Analysis of Model Effectiveness in Adversarial Scenario

In order to verify the effectiveness of the model proposed in this paper in the adversarial scenario, this paper conducted comparative experiments on the four processed datasets containing the adversarial text before and after the improvement. The experimental results are shown in Table 3. Compared with the original model, the Precision, Recall, and F1 value of the fusion graph embedding model in the adversarial scenario have significantly improved in the four datasets, which increased by 3.76%, 3.93%, 4.16%, and 6.49%, respectively. This indicates that the adversarial knowledge in the text can be learned by constructing an undirected graph of the phonetic glyph relationship and representing the nodes in the graph-by-graph embedding, which verifies the effectiveness of the proposed method in improving the robustness of the Chinese named entity recognition model in the adversarial scenario. The possible reason for the improvement of model robustness is that in the adversarial scenario, adversarial text is usually replaced according to a phonetic and glyph similarity relationship, and the model encodes and represents the Chinese

characters in the text through the semantic embedding layer, which integrates the phonetic glyph adversarial knowledge of Chinese characters. The Chinese characters with phonetic or glyph similarity relationships have similar embedding representations, which improves the performance and robustness of the named entity recognition model. At the same time, on the two relatively non-standard datasets of Resume and Weibo, the improvement of the F1 value of the model is more obvious, mainly because the integration of the graph embedding representation is equivalent to enhancing the semantic coding of the model, and the enhancement effect is more obvious on the small datasets.

Table 3. Compare experimental results in adversarial scenario. The best results are marked in bold.

Dataset	Ontonotes4.0			MSRA		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BiLSTM + CRF	45.10	45.46	45.28	63.15	63.47	63.31
BiLSTM + CRF + SC	49.47	46.92	48.16	63.36	67.41	65.32
BiLSTM + CRF + Graph embedding	**50.12**	**48.01**	**49.04**	**65.28**	**69.32**	**67.24**
Dataset	**Resume**			**Weibo**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BiLSTM + CRF	82.81	83.93	83.37	38.12	40.67	39.35
BiLSTM + CRF + SC	83.24	87.26	85.14	45.97	36.84	40.90
BiLSTM + CRF + Graph embedding	**85.41**	**89.75**	**87.53**	**51.24**	**41.47**	**45.84**

In order to fully verify the effectiveness of the Chinese named entity recognition method based on fusion graph embedding, this paper introduces the Chinese named entity recognition defense method based on spelling correction [27] for comparative experiments. The experimental results are shown in Table 3. It can be seen from the results that both the Chinese named entity recognition method based on fusion graph embedding and the named entity recognition method based on spelling correction proposed in this paper have improved the model recognition performance. The F1 values of this method on OntoNotes4.0, MSRA, Resume, and Weibo datasets are 0.88%, 1.92%, 2.39%, and 4.94% higher than those based on spelling correction, respectively. The main reason is that the model structure based on the spelling correction method is a pipeline model that is first corrected and then recognized. The accuracy of corrected text will directly affect the model performance, and there is an error propagation problem. In this paper, the robustness of the model is improved by fusing the adversarial knowledge into the graph embedding representation, and there is no error propagation problem. Therefore, the experimental results are better than the spelling correction method, which further proves the effectiveness of the model in this paper. At the same time, on the Ontonotes4.0 and MSRA datasets, the spelling correction-based method improved significantly compared with the original model, while on the Resume and Weibo datasets, the F1 value of the spelling correction-based method model improved lower than that of the original model. The main reason is that the Ontonotes4.0 and MSRA datasets are the news domain datasets, and the text expression is more standardized, so F1 value of spelling correction is higher. However, Resume and Weibo expressions are more colloquial, which makes it difficult to use spelling correction, and the performance improvement of the model is relatively limited. This shows that the fusion graph embedding method is more widely used than the spelling correction method, and the improvement of model robustness is more obvious in the scenario where the expression is not standardized.

3.4.2. Analysis of Model Effectiveness in Non-Adversarial Scenario

To verify whether the model proposed in this paper will have a negative impact on normal text, this paper conducts comparative experiments on OntoNotes4.0, MSRA, Resume, and Weibo datasets, respectively, to verify the effectiveness of the model in non-adversarial scenario.

As shown in Table 4, on the OntoNotes4.0, MSRA, Resume, and Weibo datasets, the F1 value of the fusion graph embedding model has improved to a certain extent compared with the original model and has increased by 0.11%, 0.30%, 0.09%, and 0.23%, respectively on the four types of datasets. This proves that the method in this paper will not have a negative impact on the normal text and thus reduce the accuracy of the model. On the contrary, the F1 value of the model is slightly improved after the fusion of graph embedding. The main reason is that the graph embedding representation introduced by the semantic embedding layer of the model contains the phonetic and font information of the text. The embedding representation of the model is enhanced to improve the recognition effect of named entities. In summary, in the non-adversarial scenario that does not contain adversarial text, fusing the graph embedding representation in the semantic embedding layer of the model does not reduce the overall performance of the model.

Table 4. Compare experimental results in a non-adversarial scenario. The best results are marked in bold.

Dataset	Ontonotes4.0			MSRA		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BiLSTM + CRF	51.30	54.29	52.75	74.18	72.19	73.17
BiLSTM + CRF + Graph embedding	**53.09**	52.64	**52.86**	**74.43**	**72.54**	**73.47**
Dataset	**Resume**			**Weibo**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BiLSTM + CRF	88.66	90.61	89.62	47.00	44.98	45.97
BiLSTM + CRF + Graph embedding	88.65	**90.80**	**89.71**	45.76	**46.65**	**46.20**

Comparing the experimental results in Table 4 with the experimental results in Table 3, the results show that the accuracy, recall rate, and F1 values of the proposed model and the original model decrease to varying degrees in the scenario containing adversarial text, indicating that the accuracy of the named entity recognition model will be reduced in the case of adversarial text attacks, and the influence of adversarial text on model performance is verified. Compared with the original model, the decrease in F1 values on the four datasets after fused graph embedding was increased by 3.65%, 3.63%, 4.07%, and 6.26%, respectively, which indicates that the robustness of the model has been improved to a certain extent after the context embedding layer of the model is integrated into the graph embedding representation, and the performance degradation of the model is greatly reduced in the face of adversarial text attacks, which further verifies the effectiveness of the fused graph coding representation in improving the robustness of the Chinese named entity recognition model.

4. Conclusions

In order to solve the problem that the accuracy of named entity recognition model decreases in the presence of adversarial text scenario, this paper proposes a Chinese named entity recognition model based on fusion graph embedding. This model represents the adversarial relationship of text by constructing the phonetic and glyph similarity relations of Chinese characters. Through graph learning, the graph embedding representation of nodes in the graph is learned to be integrated into the semantic embedding representation, which effectively uses the pinyin and glyph information of Chinese characters and enhances the robustness of the Chinese named entity recognition model. Through experiments on four open datasets, the experimental results show that the accuracy, recall rate, and F1 value of the proposed method in the adversarial scenario are improved to a certain extent compared with the method of unfused graph embedding, which proves that constructing the adversarial relationship graph of character phonetic glyphs and learning the adversarial relationship between Chinese characters through graph coding representation can effectively enhance the robustness of the Chinese named entity recognition model. In

the follow-up work, we will focus on how to further improve the robustness of the model in the case of other forms of adversarial text, such as insertion, deletion, Chinese character splitting, etc. At the same time, this method of constructing an undirected graph containing an adversarial relationship and learning this adversarial relationship through graph learning can be applied to the English named entity recognition task to enhance the robustness of the English named entity recognition containing adversarial text.

Author Contributions: Conceptualization, X.S. and H.Y.; methodology, X.S.; software, X.S.; validation, X.S., H.Y., S.L. and H.W.; formal analysis, X.S.; investigation, X.S.; resources, X.S.; data curation, X.S.; writing—original draft preparation, X.S.; writing—review and editing, X.S., S.L. and H.W.; visualization, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the major science and technology special project of Songshan Laboratory, grant number 221100210700-3 and the National Natural Science Foundation of China, grant number 62002384.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, X.; Gao, T.; Lin, Y.; Peng, H.; Yang, Y.; Xiao, C.; Liu, Z.; Li, P.; Sun, M.; Zhou, J. More data, more relations, more context and more openness: A review and outlook for relation extraction. *arXiv* **2020**, arXiv:2004.03186.
2. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569. [CrossRef]
3. Du, X.; Wu, H.; Yi, Z.; Li, S.; Ma, J.; Yu, J. Adversarial Text Attack and Defense: A Review. *J. Chin. Inf. Technol.* **2021**, *35*, 1–15.
4. Gong, H.; Li, Y.; Bhat, S.; Viswanath, P. Context-sensitive malicious spelling error correction. In Proceedings of the WWW'19: The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2771–2777.
5. Alshemali, B.; Kalita, J. Toward mitigating adversarial texts. *Int. J. Comput. Appl.* **2019**, *178*, 1–7. [CrossRef]
6. Liu, S.; Yang, T.; Yue, T.; Zhang, F.; Wang, D. Plome: Pre-training with misspelled knowledge for chinese spelling correction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual, 1–6 August 2021; Volume 1, pp. 2991–3000.
7. Xu, H.-D.; Li, Z.; Zhou, Q.; Li, C.; Wang, Z.; Cao, Y.; Huang, H.; Mao, X. Read, listen, and see: Leveraging multimodal information helps chinese spell checking. *arXiv* **2021**, arXiv:2105.12306.
8. Liu, S.; Song, S.; Yue, T.; Yang, T.; Cai, H.; Yu, T.; Sun, S. CRASpell: A Contextual Typo Robust Approach to Improve Chinese Spelling Correction. In *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 3008–3018.
9. Wang, Y.; Bansal, M. Robust machine comprehension models via adversarial training. *arXiv* **2018**, arXiv:1804.06473.
10. Liu, H.; Zhang, Y.; Wang, Y.; Lin, Z.; Chen, Y. Joint character-level word embedding and adversarial stability training to defend adversarial text. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8384–8391. [CrossRef]
11. Dong, X.; Luu, A.T.; Ji, R.; Liu, H. Towards robustness against natural language word substitutions. *arXiv* **2021**, arXiv:2107.13541.
12. Ou, H.; Yu, L.; Tian, S.; Chen, X. Chinese adversarial examples generation approach with multi-strategy based on semantic. *Knowl. Inf. Syst.* **2022**, *64*, 1101–1119. [CrossRef]
13. Jones, E.; Jia, R.; Raghunathan, A.; Liang, P. Robust encodings: A framework for combating adversarial typos. *arXiv* **2020**, arXiv:2005.01229.
14. Yang, Y.; Wang, X.; He, K. Robust Textual Embedding against Word-level Adversarial Attacks. *arXiv* **2022**, arXiv:2202.13817.
15. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
16. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
17. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; pp. 1746–1751.
18. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
19. Lafferty, J.; Mccallum, A.; Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning, San Francisco, CA, USA, 28 June–1 July 2001; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2001; pp. 282–289.

20. Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; Liu, S. 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), Hong Kong, China, 3–7 November 2019; pp. 3830–3840.
21. Levow, G. 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.
22. 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. LDC2011T03*; Linguistic Data Consortium: Philadelphia, PA, USA, 2011.
23. Yue, Z.; Jie, Y. Chinese NER Using Lattice LSTM. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 1554–1564.
24. Peng, N.; Dredze, M. Named entity recognition for chinese social media with jointly trained embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 548–554.
25. Sun, L.; Hashimoto, K.; Yin, W.; Asai, A.; Li, J.; Yu, P.; Xiong, C. Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT. *arXiv* **2020**, arXiv:2003.04985.
26. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
27. Xu, M. Pycorrector: Text Error Correction Tool. Available online: https://github.com/shibing624/pycorrector (accessed on 1 October 2022).

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.