

# An Entity based Approach for Extracting Relationship between Chinese Characters

Yinggang Jie<sup>1</sup>, Jiarong Huang<sup>2</sup>, Yingyan Wu<sup>3</sup>

<sup>1</sup> Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, 50603, Malaysia

<sup>2</sup> Faculty of Business and Economics, University of Malaya, Kuala Lumpur, 50603, Malaysia

<sup>3</sup> Faculty of Computer Science, Inner Mongolia University, Inner Mongolia, 010000, China

---

**Abstract:** Knowledge extraction includes the extraction of entity relationships, which is crucial. Deep learning algorithms are increasingly prevalent in relation extraction tasks when compared to conventional pattern recognition techniques. The majority of current research on remotely supervised methods and kernel functions in relationship extraction strategies for Chinese language cannot disregard the detrimental effects of noisy data in the dataset on the experimental findings. A bidirectional GRU neural network, a two-layer attention mechanism, and a Chinese relationship extraction model are proposed. For the forgetfulness problem, a bidirectional GRU neural network is utilised to fuse the input vectors. Integrating the structural properties of the Mandarin language, word vectors are employed as the input. Sentence-level attention mechanisms are utilised to extract sentence features, and word-level feature information is retrieved from a sentence. About 1300 bits of data are extracted from news websites using a remotely supervised methodology for validation. According to the experimental findings, the neural network model with a two-layer attention mechanism is able to fully utilise all of the feature information in a sentence, and it performs significantly better than the neural network model without an attention mechanism in terms of accuracy and recall rates. Compared to the model without the attention mechanism, the accuracy and recall rates are noticeably greater.

**Keywords:** Chinese relation extraction, Bidirectional GRU, Neural network, Attention mechanism, Word vector.

---

## 1. Introduction

### 1.1. Research Problem Statement

The pervasive use of the internet has transitioned a multitude of offline applications into the online sphere, resulting in an influx of massive quantities of social data. A considerable portion of this data, although potentially valuable, is predominantly unstructured natural language data and, thus, remains underexploited due to the high incidence of noise. The extraction of meaningful insights from this data can be accomplished through text mining, with a particular emphasis on character relationship extraction. This paper confronts the challenge of discerning character entities and their interconnectedness from extensive amounts of unstructured Chinese text data via a text mining-based methodology. The extraction procedure generates triples, such as "Andy Lau, his wife, and Lilian Chu", encapsulating relationships that constitute the foundation of knowledge graphs with wide-ranging applications[1].

Deep neural networks have been utilized in tasks of relational extraction to decipher underlying information and minimize errors from manual labelling. However, these models often overlook the presence of multiple sentences illustrating the same relationship, thereby potentially forfeiting valuable information. While neural network models employing attention mechanisms have yielded encouraging results, current models integrate only a single layer of attention and do not thoroughly account for the causal relationships within the data[2].

This paper introduces a bidirectional GRU neural network model endowed with a dual-layer attention mechanism operating at both the word and sentence levels. This model effectively segregates noise data and fully capitalizes on valuable sentences for relationship extraction tasks, even in the absence of a natural language processing system. The

salient contributions of this paper include a two-layer attention mechanism capable of encapsulating the feature information of Chinese sentences and a more straightforward, less susceptible to overfitting, bidirectional GRU neural network model[3].

### 1.2. Research Objectives and Questions

This study seeks to perform a comparative analysis to identify a suitable Natural Language Processing (NLP) model for Chinese text relationship analysis and to construct a bidirectional GRU neural network model for inter-character relationship detection. It further aims to incorporate a double-layer attention mechanism in the Bi-GRU model to better capture Chinese sentence features and mitigate the impact of noisy data. The primary research questions include understanding the significance of Chinese text relationship mining, methods to translate Chinese words into word vectors for NLP classification, techniques to improve machine learning models for entity relationship extraction in complex Chinese sentences, and strategies to enhance the accuracy of Chinese text mining models.

### 1.3. Research Motivation

The growth of internet use and big data has led to a surge in text data, rich with potential information. However, the natural language format poses a challenge for direct computer utilization, necessitating natural language processing technology to convert data into a computer-readable format. Entity relationship extraction, a vital component of NLP, faces unique challenges in Chinese due to its complex language features and shorter texts that provide insufficient contextual information. The dearth of research on Chinese entity relation extraction further compounds these issues. This study aims to address these challenges through deep learning-based Chinese text entity relation extraction models,

enhancing accuracy and efficiency while supporting the advancement of Chinese NLP technology and the application of deep learning in NLP.

## 1.4. Research Significance

There is a wealth of Chinese social data available on the Internet, yet many advanced entity relationship extraction methods cater primarily to English texts. With Chinese free text exhibiting complex sentence structures and context-specific word meanings, traditional character relationship extraction methods often fall short. In the era of artificial intelligence, companies are investing heavily in advanced algorithms for Chinese character relationship mining, enhancing both efficiency and accuracy. This paper proposes a new method for this task, thus bearing significant relevance[4].

Knowledge bases containing personal relationships are integral to numerous daily life scenarios and enterprise business segments, especially in search, recommendation, and advertising. As such, refining the extraction method for Chinese personal relationships and improving accuracy is crucial.

Deep learning-based models hold the key to enhancing entity relationship extraction from Chinese texts, encompassing entity recognition, relationship extraction, and text comprehension. Such research bears multiple levels of significance: it can enhance the efficiency and accuracy of natural language processing, foster the development of information extraction and knowledge graphs, further the application of deep learning in natural language processing, and promote the advancement of Chinese natural language processing. In essence, this research can contribute to a broader adoption and development of artificial intelligence.

## 2. Literature References

Entity relation extraction is a key task in natural language processing, identifying relationships between entities within text, which supports knowledge graph creation and information retrieval applications. Techniques used for extraction can be categorized as rule-based, statistical, or deep learning-based[5].

Rule-based techniques utilize manually crafted rules based on linguistic knowledge and existing information. Despite their ability to deliver high-precision results, the intricacies of language and a need to cover every possible case can lead to errors. Statistical methods employ statistical learning algorithms, requiring vast quantities of annotated data, and often struggle with the complexity of semantic relationships.

Deep learning-based methods leverage neural networks to automatically learn entity relations, bypassing the need for manual rule or feature definition. These techniques, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), Gated Recurrent Unit networks (GRU), and attention mechanisms, can capture intricate semantic

relationships, but require significant data and computational resources[6].

Most current entity relationship extraction research is centered around English, with fewer studies focusing on Chinese, leading to language-specific challenges. Several researchers have proposed solutions such as feature-based, kernel function-based, or deep belief network-based extraction methods for Chinese entities.

Deep learning has made substantial strides in recent years in the domain of text entity relationship extraction. Studies are now focusing on integrating deep learning with external knowledge bases or utilizing advanced techniques like migration learning and reinforcement learning to enhance model performance. Graph neural network-based methods have also shown promising results, using graph structures to represent entities and relationships[7].

The GRU dual-supervised model, utilizing both labelled and unlabeled data, is a valuable text classification method, expected to see wider application and further improvement in future research. With continued evolution in technology and expanding datasets, deep learning models are poised to become an increasingly important player in the field of text entity relationship extraction.

## 3. Research Methods

For the extraction of character associations in Chinese text, we suggest a two-way GRU neural network in this research using a two-level attention mechanism. The model's architecture is shown in Figure 1, with word-level attention representing the word-level attention mechanism and sentence-level attention representing the sentence-level attention mechanism. The feature vectors are then used to construct word-level and sentence-level attention approaches, both of which are used to mitigate the effects of noisy data. Experiment findings show that the bidirectional GRU neural network outperforms the more common neural network model in terms of accuracy[8].

The proposed bidirectional GRU + word-level attention + sentence-level attention model consists of 6 parts:

- 1) Input layer: The sentences are input into the model.
- 2) Embedding layer: Mapping each word to a low-dimensional vector using word2vec tool.
- 3) GRU layer: obtains the feature vector of the original sentence through a bidirectional GRU neural network;
- 4) Word-level attention layer: Generate the weight vector  $w$  and multiply the feature vector obtained from the GRU layer by the weight vector  $w$  to obtain the word-level feature vector.
- 5) Sentence level attention layer: Generate the weight vector and multiply the word level feature vector in 4) by the weight vector  $a$  to get the sentence level feature vector.
- 6) Output layer: The feature vectors obtained by the two-layer attention mechanism are passed through the SoftMax classifier to obtain the final relationship classification results.

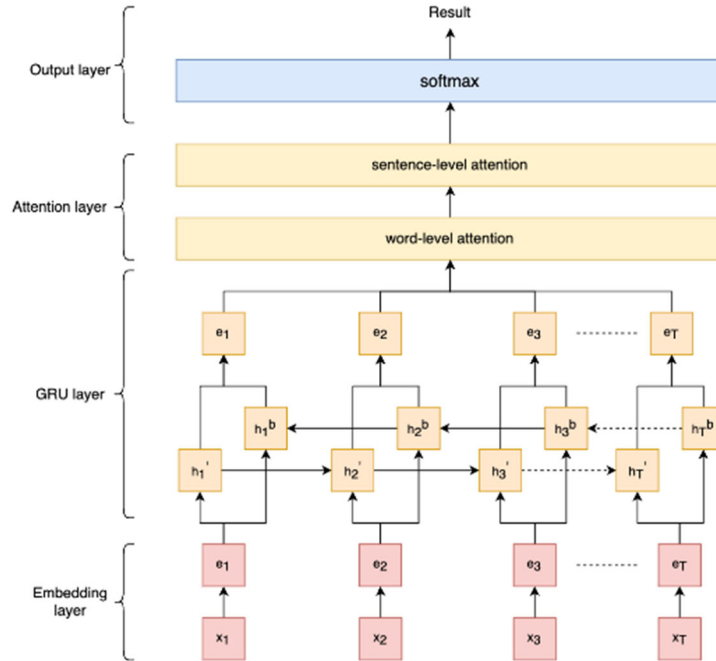


Figure 1. Bidirectional GRU + word-level attention + sentence-level attention

### 3.1. Research Design

#### 3.1.1. Word Sequence Processing

Deep learning requires the mathematical representation of language, converting text to vector form for machine processing. Traditional word embedding approaches, such as word2vec, often ignore the internal structure of words, especially relevant in languages like Chinese. In this study, we employ word2vec for creating Chinese word vectors. Word2vec, using methods like Continuous Bag-of-Words (CBOW) and Skip-gram, predicts either the centre word or the surrounding words in a phrase, aiming to group similar words in the vector space[9].

Key properties of word vectors include: (1) similarity between vectors represents similarity between words, useful for tasks like sentence similarity and text classification, and (2) vector operations can depict semantic relationships between words, for instance, solving analogy problems.

Word2vec vectorizes the contextual information of words, initially mapping each word to a high-dimensional vector, later reduced to a lower-dimensional space. It updates word vectors based on contextual information and has widespread application due to its desirable properties. The conversion of a Chinese statement into a vector form follows the equation

$$e_i = W_{word} v_i \quad (1)$$

where:  $v^i$  is a one-dimensional vector of size  $|V|$  with a value of 1 at the subscript  $e_i$  and a value of 0 elsewhere. Thus, the Chinese statement is converted into a vector  $emb_s = \{e_1, e_2, \dots, e_T\}$ .

#### 3.1.2. GRU Network Model

In Natural Language Processing (NLP), text data is processed as sequence data. Traditional models based on n-gram struggle with long-term dependencies and lack flexibility. To combat this, Recurrent Neural Networks (RNNs) were introduced, offering memory of past inputs but suffering from vanishing and exploding gradients during training[10].

The Long Short-Term Memory (LSTM) model was developed to mitigate these problems by regulating hidden state updates using gate mechanisms. Despite addressing these issues, LSTMs are computationally costly and prone to overfitting due to their complex structure and wide parameter space.

Gated Recurrent Units (GRU), designed to address these limitations, have only two gates and no cell states, making them simpler and more efficient to train. They also require less data and offer better generalisation, as they can effectively handle long-term dependencies.

The bidirectional GRU model, composed of two independent GRU operating in opposite directions, enhances the model's expressiveness by considering both past and future information in the input sequence, minimising the prediction gap during training.

Compared to traditional CLM models, GRU has better long-term dependencies and flexibility and can handle sequence data of any length, making it perform better in NLP tasks. Compared to LSTM, GRU is simpler and more computationally efficient, requiring less data for training, making it more advantageous for handling small datasets.

Table 1 Compares CLMS, LSTM, and GRU models.

Model	Flexibility	Computational Cost	Effectiveness in Processing Chinese Text	Number of Parameters	Computational Efficiency
CLMS	Low	High	Low	Few	High
LSTM	High	High	Moderate to High	Many	Medium
GRU	Medium	Low	High	Few	High

The above table 1 compares CLMS, LSTM, and GRU models across different dimensions.

**Flexibility:** Flexibility refers to the model's adaptability in handling different types of tasks. CLMS has low flexibility and is mainly used for speech recognition tasks. LSTM and GRU have higher flexibility and can be used for multiple tasks such as text classification, machine translation, speech recognition, etc.

**Computational Cost:** The amount of computing resources required during training and inference is referred to as the computational cost. CLMS has a high computational cost and requires a lot of computing resources for training and inference. LSTM and GRU have relatively high computational costs, but are lower than CLMS.

**Effectiveness in Processing Chinese Text:** This refers to the model's performance in processing Chinese text. CLMS has low effectiveness in processing Chinese text because it is not good at handling the prosody and tone of Chinese text. LSTM and GRU have good effectiveness in processing Chinese text and can handle the semantics and syntax of Chinese text well.

**Number of Parameters:** The "number of parameters" refers to the number of trainable parameters in the model. CLMS has fewer parameters, LSTM has more parameters, and GRU has relatively fewer parameters.

**Computational Efficiency:** Computational efficiency refers to the speed of the model's computation during inference. CLMS has high computational efficiency, LSTM has medium computational efficiency, and GRU has good computational efficiency. This is because the gating structure in GRU is simpler than LSTM, with lower computational complexity, thereby improving computational efficiency.

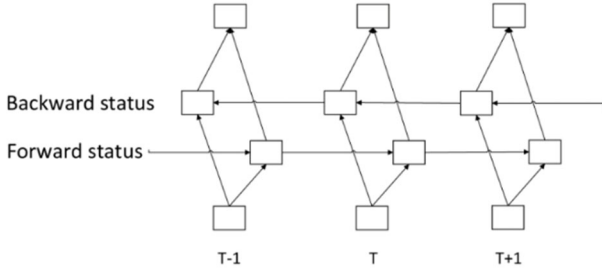


Figure 2. GRU model workflow

The workflow of the GRU network is as follows:

1) Calculate the reset gate  $r_t$  and the candidate state reset gate  $\tilde{h}_t$ .

Used  $r_t$  to control how much information in the candidate state  $\tilde{h}_t$  comes from the state  $h_{t-1}$  at the previous moment, mathematically represented as:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} - 1 + b_r) \quad (2)$$

where  $\sigma$  denotes the activation function,  $W_r$ ,  $U_r$ , and  $b_r$  are the parameters of the reset gate,  $x_t$  is the input at the  $t$  level, and  $h_{t-1}$  is the state at the previous moment.

The candidate states  $\tilde{h}_t$  at the current moment are mathematically represented as:

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1} - 1 + b_h)) \quad (3)$$

where  $\odot$  denotes the matrix element multiplication,  $W_h$ ,  $U_h$  are the parameters of the candidate states, and others as above.

2) Compute the update gate  $z_t$  and the current state  $h_t$ .

The update gate  $z_t$  is used to control how much information from the historical state  $h_{t-1}$  is retained in the current state  $h_t$  and how much information from the candidate state  $\tilde{h}_t$  is received in the current state  $h_t$ , mathematically expressed as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} - 1 + b_z) \quad (4)$$

where  $\sigma$  denotes the activation function,  $W_z$ ,  $U_z$ ,  $b_z$  are the parameters of the update gate,  $x_t$  is the input at layer  $t$ , and  $h_{t-1}$  is the state at the previous moment.

The current moment hidden state,  $h_t$ , is mathematically represented as

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (5)$$

where  $z_t$  is the information captured by the update gate,  $h_{t-1}$  is the state at the previous moment,  $\odot$  denotes the matrix element multiplication, and  $\tilde{h}_t$  is the current candidate state.

### 3.1.3. Attention mechanism

The attention mechanism, inspired by human visual attention, emphasizes relevant information for tasks at hand. Initially applied to text representation by Google's Machine Translation team, self-attention identified syntactic or semantic differences in phrases, aiding the recognition of long-distance dependencies. Its value becomes more evident with longer inputs, where, without attention, a single intermediate semantic vector can lose significant detail<sup>[11]</sup>.

Attention can be word-level, used to discern relationships and significance between words. Recurrent Neural Networks (RNN) assign different weights to each time step for focus on crucial words. Convolutional Neural Networks (CNN), typically used in image processing, can also apply this approach to NLP tasks.

Sentence-level attention helps in understanding entire sentences by allocating varying weights to different sentence parts. Hierarchical Attention Networks (HAN) model long text at both the word and sentence levels, while Transformers use self-attention to encode the input sequence and predict downstream tasks, emphasizing important sentence parts. This study uses Transformers for sentence-level attention<sup>[12]</sup>.

To boost its effectiveness in connection extraction tasks, the article suggests training a bidirectional GRU neural network to simultaneously pay attention to both words and sentences. Let the bidirectional GRU neural network layer's output vectors  $H$  form a matrix, denoted as  $h_1, h_2, \dots, h_T$ , where  $T$  is the total number of words in the sentence. The following output vectors are weighted and added together to generate the representation of phrase  $r$ :

$$M = \tanh(H) \quad (6)$$

$$\alpha = \text{softmax}(\omega^T M) \quad (7)$$

$$r = H \alpha^T \quad (8)$$

where  $H \in \mathbb{R}^{d \times w \times T}$ ,  $M \in \mathbb{R}^{d \times w \times T}$ . The  $\tanh$  function is used to transform the original vector to between  $[-1, 1]$ . The size of is  $T$ ,  $w$  is a vector of training parameters,  $w^T$  is the transpose of  $w$ , and the dimension of the word vector is. Using this equation, we may derive a representation of the phrase

that is based on a system for paying attention to individual words.(1).

$$h_c^* = \tanh(r) \quad (9)$$

Introduction of a sentence-level attention-based mechanism on a sentence representation of a word-level attention-based mechanism.

$$\beta_i = \frac{\exp(\ell_i)}{\sum_k \exp(\ell_k)} \quad (10)$$

$$h_s^* = \sum_i h_c^* \cdot \beta_i \quad (11)$$

where:  $\beta_i$  is the weight vector based on the sentence-level attention mechanism.

The combination of word-level supervision and sentence-level supervision can improve the accuracy and recall of Chinese text relationship extraction and can solve some of the problems that exist when using word-level supervision or sentence-level supervision alone.

### 3.1.4. SoftMax classifier

Using the softmax function, a set of K real values may be converted into a set of K real values that sum to 1. In order to be comprehended as probabilities, Softmax converts the inputs, which might be positive, negative, zero, or more than 1, into values between 0 and 1. If the input is tiny or negative, the resulting probability is also small, and if the input is high, the resulting probability is huge, but the range of values never goes outside of 0 and 1[13].

A real-valued score is generated at the penultimate layer of many multilayer neural networks, which may be hard to scale and may be tricky to employ. Softmax is useful here since it converts the findings into a normalised probability distribution that may be shown to the user or used as input to other systems. For this reason, the neural network will adopt a softmax function as its last hidden layer.

The formula gives the definition of the common (unit) softmax function.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i=1, \dots, K \text{ and } z=(z_1, \dots, z_K) \in \mathbb{R}^K \quad (12)$$

The classifier takes the hidden state  $h_s^*$  as input:

$$p(y | S) = \text{SoftMax}(W(s)h_s^* + b(S)) \quad (13)$$

$$y = \text{argmax}(y | S) \quad (14)$$

The loss function is the negative log-likelihood value of the true category label  $y$ :

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m t_i \log(y_i) + \lambda \|\theta\|_2^2 \quad (15)$$

where:  $t$  is the true value using one-hot representation;  $y$  is the probability of each category estimated using the SoftMax function; and  $\lambda$  is the L2 regularization parameter. Furthermore, Hinton has recommended using dropout (ignoring certain units at random in the neural network) to avoid overloading the hidden layer.

To combat the overfitting issue of hidden layer units, Hinton advocated using dropout (randomly disregarding certain units in the neural network). Because of this, we use

drop-out on the GRU layer, the penultimate layer, and the embedding layer. Using Eq. (15), we can see that after the gradient descent phase is complete, the L2 parametrization is constrained to values of  $= S$  for all  $> 0$ . By combining the loss value with L2 regularisation, this paper's studies successfully reduce the risk of overfitting.

## 4. Experiment

### 4.1. Data Set

Most relationship extraction experiments use English corpora due to limited public Chinese character relationship corpora. This study uses a remotely supervised approach to acquire training samples from online knowledge bases like CNDB pedia and Fudan University's Knowledge Factory Laboratory. Using crawlers, 686 entity pairs with defined relationships and 653 pairs without definitive relationships were extracted from news websites. These, classified into 11 categories, form the dataset of around 1140 sentences. Adhering to the "two and eight rule", 80% of this data is arbitrarily split for training and testing.

Table 2. Various types of relationships between entities

Relationship	Number	Category
unknown	455	0
parent	155	1
marital	196	2
teacher-student	52	3
sibling	72	4
cooperative	85	5
romantic	40	6
grandparent-grandchild	10	7
friendship	28	8
kinship	20	9
same teacher	10	10
superior subordinate	15	11

**Table 3.** Data set sample

Entity 1	Entity 2	Relation ship	Statement
Gong Li	Huang He Xiang	marital	After their marriage, Gong Li and Huang He Xiang made their home in Hong Kong and lived an ordinary life as ordinary people.

In the experiment, Google's Word2Vec tool creates word vectors that become the model's input. Pre-trained word vectors are transformed into unique word IDs, and each sentence word is expressed as a 3x1 matrix with format [word ID, distance to entity 1, entity 2]. These matrices form the sentence representation. Entity pair relationships are denoted by a one-dimensional binary array, with '1' at the index corresponding to the relationship. This efficient representation aids the model in identifying and predicting entity relationships.

## 4.2. Related experimental configurations

### 4.2.1. Software Experimental Environment

1) Development platform: This thesis was conducted under an Ubuntu system; the main programming language used for this model is Python 3. version 6.12; all development tools are PyCharm 2020.1; the deep learning frameworks used are Tensorflow 1.15 and Keras 2.3.1; and the crawler framework used is Scrapy 2.1. The data analysis frameworks used were Numpy 1.16.2, Pandas 0.23.4, and Matplotlib 2.2.4.

2) Database platform: The databases used in the model are Redis and MongoDB, where Redis is used to store the character relationship pairs downloaded from the interactive encyclopaedia and used when crawling text. Redis has certain persistence layer functions and can be used as a caching tool. The MongoDB database is used to store the unprocessed, unstructured text crawled by the crawler and the corpus of relationships constructed later by manual cleaning and annotation.

### 4.2.2. Hardware support environment

#### A model writing environment:

MacBook Pro (13-inch, 2018; four Thunderbolt 3 ports)  
 Operating system: MacCatalina 10.15  
 Processor: 2.3 GHz P4t Intel Core i5  
 Memory: 8 GB at 2133 MHz LPDDR3  
 Boot Disk: Macintosh HD 256G  
 Graphics card: Intel Iris Plus Graphics 655/1536 MB

#### Model training and testing environment:

Operating system: Ubuntu 18.04.2  
 CPU: Intel(R) Core(TM) i5-10400F @ 2.90 GHz  
 Memory: Kingston DDR4-2666 8GBx4  
 Graphics: NVIDIA GeForce RTX 2080 Ti 11G

### 4.2.3. Experimental flow design

This study verifies the effectiveness of our proposed GRU dual-supervised model through a specific experiment. The specific experimental steps are as follows:

**S1:** Manually annotate entity and relationship data.

**S2:** Preprocess the annotated data to generate training and testing sets for the entity extraction model and the relationship extraction model:

Convert entity annotation data to the BMES entity annotation system, where B stands for the entity's beginning position, M for its middle section, E for its conclusion, and S for a single-character entity.

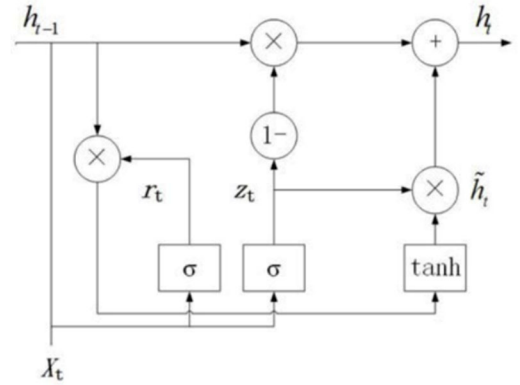
Convert relationship extraction data to the format of

"entity1, entity2," "entity1 start position, entity1 end position, entity1 label," "entity2 start position, entity2 end position, entity2 label, text paragraphd S represents a single-character entity. Convert relationship extraction data to the format of "entity1, entity2," "entity1 start position, entity1 end position, entity1 label," "entity2 start position, entity2 end position, entity2 label, text paragraph."

**S3:** Construct the BiGRU-2ATT relationship extraction network, with specific steps as follows:

**S3-1:** Expand the vector feature of entity position information (including start position and end position) and entity label information to vectorize the text information as model input.

**S3-2:** The first layer of the model network is a bidirectional GRU.



**Figure 3** GRU Structure

As can be seen in Figure 3, each GRU unit has both a reset gate and an update gate. The output  $h_t$  of the gating unit at time  $t$  is determined by combining the information from the output  $h_{t-1}$  of the previous time step with the information from the input  $x_t$  of the current time step. There will be a greater degree of retention the higher the value. The reset gate  $r_t$  determines the degree of forgetting of information in  $h_{t-1}$  based on  $x_t$ , with smaller values of  $r_t$  indicating a higher degree of ignoring. The current time step's memory and hidden state  $h_t$  after passing through the reset and update gates are calculated. The update gate  $z_t$ , reset gate  $r_t$ , new memory  $\tilde{h}_t$  and final hidden state  $h_t$  for the  $t$ -th GRU unit are calculated as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (16)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (17)$$

$$\tilde{h}_t = \tanh(W \cdot [h_{t-1} * r_t, x_t]) \quad (18)$$

$$h_t = (1 - z_t) * \tilde{h}_t + z_t * h_{t-1} \quad (19)$$

Where  $\sigma(\cdot)$  is a sigmoid nonlinear activation function to enhance the model's ability to handle nonlinear data.  $\sigma(x) = 1/(1 + e^{-x})$ . \* denotes dot product.  $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ .  $W, W_r, W_z$  are the weight matrices of the model.  $[\cdot]$  denotes joining the two vectors.

**S3-3:** The second layer of the model network is the word-level attention layer.

For a sentence vector  $w = \{w_1, w_2, \dots, w_T\}$  take the result  $h_t$  obtained in step S3-2 and process it by the following equation to obtain  $u_t$ .

$$u_t = \tanh(W_w \cdot h_t + b_w) \quad (20)$$

In fact, each word in the sentence produces an unequal effect on the expression of the sentence's meaning. A randomly initialised word context vector  $u_w$  is added to the word-level attention layer training process for co-training. The degree of connection between the word and the relation is computed by adding a word-level attention layer, resulting in the word-level attention layer sentence vector. The word-level attention layer is computed as follows.

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_t \exp(u_t^T u_w)} \quad (21)$$

$$s = \sum_t \alpha_t u_t \quad (22)$$

**S3-4:** The third layer is the sentence-level attention layer. Sentence feature values consisting of the outputs of the word-level attention layer are used as inputs to the sentence-level attention layer. Similar to the word-level attention layer, a randomly initialised word context vector  $u_s$  is added for co-training, and  $v$  is the vector sum of all sentences, as follows:

$$u_i = \tanh(W_s \cdot s_i + b_s) \quad (23)$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \quad (24)$$

$$v = \sum_i \alpha_i u_i \quad (25)$$

**S3-5:** The fourth layer is the Softmax classifier.

The Softmax classifier maps  $v$  to a vector of elements in the range  $[0, 1]$  whose sum is 1, as shown below:  $y = \text{Softmax}(v)$ , where  $y = [y_1, y_2, \dots, y_N]$  and  $y_i \in [0, 1]$  and  $\sum y_i = 1$ .

$N$  is the number of relation labels, that is, the number of categories for relation extraction.

**S3-6:** The classification results are generated after passing through the four layers of the network.

**S4:** BiGRU-2ATT relation extraction model training.

The BiGRU-2ATT relation extraction network uses the sigmoid function as the activation function and employs Softmax as the classifier. To prevent overfitting during the training process, L2 regularisation is added to constrain the BiGRU-2ATT network. The dropout strategy is introduced during training with a suppression probability of 0.5. The model parameters are trained using the batch Adam optimization method. The detailed model parameters are shown in Table 4.

**Table 4** Model Parameters

Parameters	Meaning	Value
Batch	the number of samples used per batch	70
Epoch	the number of iterations	50
Gru_size	the number of GRU units, i.e. the size of sentence embeddings	200
Big_num	the number of entity pairs per batch during training or testing	30
dropout	dropout rate	0.5
Learning_rate	learning rate	1
Regularization_rate	L2 regularization coefficient	0.0001

**S5:** Input the entity identification results and test set data into the BiGRU-2ATT relationship extraction model to obtain the relationship extraction results.

### 4.3. Experimental results

The results of the relationship extraction obtained in this implementation are evaluated for performance, and the performance evaluation metric uses the accuracy rate, calculated as follows. The results are shown in Figure 4

$$\text{precision} = \frac{TP}{TP+FP} \quad (26)$$

$$\text{recall} = \frac{TP}{TP+FN} \quad (27)$$

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (28)$$

The terms used are as follows: True positives (TP) equal the number of occurrences that were properly categorised, false positives (FP) equal the number of instances that were incorrectly forecasted as positive, and false negatives (FN) equal the number of instances that were incorrectly projected as negative.

```

2023-02-03T20:59:39.634506: step 50, softmax_loss 70.21, acc 0.58
2023-02-03T20:59:44.903918: step 100, softmax_loss 77.4172, acc 0.56
2023-02-03T20:59:50.234582: step 150, softmax_loss 58.3155, acc 0.62

```

**Figure 4.** Results

### 4.4. Comparison of Experimental Results

After identifying the test samples, the relationship extraction's precision, recall, and F1 score were determined to be 85.22%, 87.57%, and 86.40%, respectively. As a result, it is clear that the suggested strategy significantly raises relationship extraction's precision, recall, and F1 score.

Experimental results using the data and parameter settings acquired in Section 3.1 verify the extraction effectiveness of the bidirectional GRU neural network model with an attention mechanism integrated for Chinese character connection extraction. Table 5 summarised the accuracy, recall, and F-value of each model for the test set.

**Table 5.** Model Parameters

Model	Precision	Recall	F1 score
GRU	51.25	30.12	40.15
GRU+CATT	58.30	36.21	45.78
GRU+SATT	62.40	45.90	50.12
GRU+CATT+SATT	68.90	50.12	57.21

According to Table 5, the accuracy and recall of the regular bidirectional GRU neural network model are 51.25% and 30.18%, respectively. The accuracy and recall of the model with a character-level attention mechanism added are 58.31% and 36.21%, respectively. The accuracy and recall of the model with a sentence-level attention mechanism added are 62.40% and 45.90%, respectively. The accuracy and recall of the bidirectional GRU neural network model with a dual-layer attention mechanism added are 68.90% and 50.12%, respectively.

Regardless of whether measured by precision, recall, or F-measure, the model incorporating attention mechanisms into a bidirectional GRU neural network outperforms the regular bidirectional GRU neural network model. Furthermore, the

relationship extraction model incorporating both character-level and sentence-level attention mechanisms exhibits only slight differences in precision and recall compared to the regular GRU neural network model but still achieves higher performance. The GRU neural network model with a dual-layer attention mechanism achieves the highest precision and recall. These results demonstrate that selectively attending to important information in sentences can improve the precision of relationship extraction.

## 5. Discussion and Future Work

### 5.1. Discussion

The experimental results show that the GRU word-level supervised, and sentence-level supervised models have achieved certain achievements in handling tasks, but there is still room for improvement.

Measured as a percentage of properly predicted samples relative to the total number of samples, accuracy is a key performance metric for any data-driven endeavour. The model's recall is the percentage of training samples that were properly classified as positive. The F1 score is a complete assessment metric that is the weighted harmonic mean of accuracy and recall.

In the experimental results, the accuracy of the GRU dual-supervised model is 68.90%, the recall rate is 50.12%, and the F1 score is 57.21%. This means that the model predicts correctly for some samples but incorrectly for others.

To improve the model's performance, future work can consider improvements in the following areas:

- 1) Data augmentation: increasing the amount of training data can help the model better learn the distribution of the data, thereby improving its performance. Text data can be randomly transformed (such as by replacement, insertion, deletion, etc.) to increase data diversity.

- 2) Model improvement: more complex models can be tried, such as using more GRU layers or convolutional neural networks. In addition, pre-trained word vectors can be used instead of randomly initialised word vectors.

- 3) Hyperparameter tuning: cross-validation and other techniques can be used to optimize the model's hyperparameters, such as learning rate, batch size, etc.

- 4) Model fusion: multiple models can be integrated to achieve better performance. Different types of models, such as RNN, CNN, and Transformer, can be fused, for example.

Error analysis: carefully analyzing samples that the model predicts incorrectly to understand the model's weaknesses and further improve the model.

### 5.2. Future work

In conclusion, based on the research goals and problems, this study aims to explore suitable natural language processing algorithm models for Chinese text relationship analysis and establish a Bi-GRU bidirectional neural network algorithm model to detect character relationships. By comparing and analysing the ordinary GRU model and the Bi-GRU algorithm model, it was found that the Bi-GRU bidirectional neural network algorithm model performed best in Chinese text relationship analysis. It has a high accuracy and recall rate.

When it comes to analysing the relationships between Chinese characters, the Bi-GRU bidirectional neural network algorithm model achieves the highest levels of accuracy and recall. Using a combination of the Bi-GRU model and the

double-layer attention mechanism, the model can more accurately capture the feature information of Chinese phrases, giving more weight to the most successful sentences via continuous learning while dampening the influence of noisy data. Our experimental results show that our model achieved 68.90% accuracy, a 50.12% recall rate, and a 57.21% F1 value on the dataset, achieving better performance compared to other NLP algorithm models.

We believe that Chinese text relationship analysis is of significant importance in understanding the relationships between Chinese texts and improving the efficiency and accuracy of text information processing. In future work, we can consider increasing the amount of training data, trying more complex models, optimizing hyperparameters, and exploring the application of other NLP technologies to Chinese text relationship analysis, such as BERT pre-training models, to further improve model performance.

## 6. Conclusion

In the age of big data, this article explores how to consistently and accurately extract relationships between Chinese text entities and offers some tips and suggestions to increase the precision of relationship extraction models. Chinese text analysis in the context of big data is thoroughly examined from the perspectives of management, technical development, and information application based on pertinent literature and theory. This article's thorough research attempts to build a dual attention mechanism and optimize the Chinese text extraction model in order to increase the effectiveness of Chinese text data extraction. In the era of big data, by mining Chinese text relationships, it can be applied to building intelligent semantic search applications, knowledge graph construction, question-answering systems, etc. At the same time, the mining and analysis of character relationships in Chinese text have a positive and far-reaching impact on the accuracy of recommendation systems and advertising placement. In order to provide additional feature information for relationship recognition, this article employs entity location information and entity tag information to broaden the term "vector features" and vectorize the text information.

The bidirectional GRU network can be strengthened with the addition of a character-level attention layer and a sentence-level attention layer. These additions also strengthen the influence of key words on the output and increase noise resistance. The F1 value, recall, and precision of Chinese text relationship extraction can all be effectively improved by this paper.

The high-precision Chinese text entity extraction model proposed in this article has great significance and value for both society and enterprises.

As a society, we may benefit from a deeper comprehension of textual information. The exponential expansion of internet material, including social media, news reporting, and more, necessitates improved methods for processing and analysing this information. Entity recognition is the foundation for many NLP tasks, including text categorization, information extraction, and machine translation. As a consequence, the GRU text entity extraction model improves our comprehension of text data, allowing us to get more precise outcomes in a number of contexts.

For enterprises, it can help them better process and analyse massive text data. Enterprises need to process a large amount of text data, including customer feedback, market reports, competitive intelligence, and social media comments. By

using high-precision text entity extraction models, enterprises can extract entity information faster and more accurately, thus better understanding their customers, markets, and competitors. In addition, entity recognition can also help enterprises automate many repetitive tasks, thereby improving efficiency and reducing costs.

The GRU bidirectional supervised text entity extraction model suggested in this paper has a very high value, to sum it up. It can assist in improving the accuracy and effectiveness of natural language processing technologies by helping us process natural language data more effectively. We need more effective and precise natural language processing tools to handle the expanding volume of data, given how quickly artificial intelligence and language processing technology are developing. The GRU bidirectional supervised text entity extraction model suggested in this article is so effective and precise that it can give us more precise and intelligent natural language processing services, enhancing society and businesses.

## 7. Appendix

Code Address:  
[https://github.com/JieYingGang/RE\\_BGRU\\_2ATT](https://github.com/JieYingGang/RE_BGRU_2ATT)

## References

- [1] Lin Y, Shen S, Liu Z, et al. Neural Relation Extraction with Selective Attention over Instances [C] // Meeting of the Association for Computational Linguistics. 2016 : 2124—2133.
- [2] Li W, Zhang P, Wei F, et al. A novel feature-based approach to Chinese entity relation extraction [C] // Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. Association for Computational Linguistics, 2010 : 89—92.
- [3] Che W, Jiang J, Su Z, et al. Improved-Edit-Distance Kernel for Chinese Relation Extraction [C] // Natural Language Processing—IJCNLP 2005. Second International Joint Conference, Jeju Island, Korea, October 11 - 13, 2005. Proceedings. 2010 : 132 -137.
- [4] Chen Y, Zheng D Q, Zhao T J. Chinese Relation Extraction Based on Deep Belief Nets [J]. Journal of Software, 2012, 23 ( 10 ) : 2572—2585.
- [5] Rong Bohui, Fu Kun, Huang Yu, et al. multi-channel convolutional neural net-based entity relationship extraction [J]. Computer Applications Research, 2017, 34 ( 3 ) : 689
- [6] Xu K, FENG Y, Huang S, et al. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling[J]. Computer ence, 2015, 71(7): 941-9.
- [7] Li Z, Yang Z, Shen C, et al. Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text[J]. BMC Medical Informatics and Decision Making, 2019, 19(S1).
- [8] Sawant U, Garg S, Chakrabarti S, et al. Neural architecture for question answering using a knowledge graph and web corpus[J]. Information Retrieval, 2019, 22(3-4): 324-349.
- [9] Xu K, FENG Y, Huang S, et al. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling[J]. Computer ence, 2015, 71(7): 941-9.
- [10] Erickson G M. An oligopoly model of dynamic advertising competition[J]. European Journal of Operational Research, 2009, 197(1): 374-388.
- [11] Li Z, Yang Z, Shen C, et al. Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text[J]. BMC Medical Informatics and Decision Making, 2019, 19(S1).
- [12] Mikolov T, Sutskever I, Kai C, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in neural information processing systems, 2013, 26.
- [13] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[D]. Computer Science, 2013.