information

Article

Chinese Mathematical Knowledge Entity Recognition Based on Linguistically Motivated Bidirectional Encoder Representation from Transformers

Wei Song [1], He Zheng [1], Shuaiqi Ma [1], Mingze Zhang [2], Wei Guo [3,*] and Keqing Ning [1,*]

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China; songwei@ncut.edu.cn (W.S.); zhenghe@mail.ncut.edu.cn (H.Z.); shuaiqi@mail.ncut.edu.cn (S.M.)
[2] State Grid Jilin Electric Power Research Institute, Changchun 130015, China; mingzezhang@petalmail.com
[3] School of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China
* Correspondence: guowei0903@ncut.edu.cn (W.G.); ningkq@ncut.edu.cn (K.N.)

Abstract: We assessed whether constructing a mathematical knowledge graph for a knowledge question-answering system or a course recommendation system, Named Entity Recognition (NER), is indispensable. The accuracy of its recognition directly affects the actual performance of these subsequent tasks. In order to improve the accuracy of mathematical knowledge entity recognition and provide effective support for subsequent functionalities, this paper adopts the latest pre-trained language model, LERT, combined with a Bidirectional Gated Recurrent Unit (BiGRU), Iterated Dilated Convolutional Neural Networks (IDCNNs), and Conditional Random Fields (CRFs), to construct the LERT-BiGRU-IDCNN-CRF model. First, LERT provides context-related word vectors, and then the BiGRU captures both long-distance and short-distance information, the IDCNN retrieves local information, and finally the CRF is decoded to output the corresponding labels. Experimental results show that the accuracy of this model when recognizing mathematical concepts and theorem entities is 97.22%, the recall score is 97.47%, and the F1 score is 97.34%. This model can accurately recognize the required entities, and, through comparison, this method outperforms the current state-of-the-art entity recognition models.

Keywords: mathematical knowledge entity recognition; LERT; BiGRU; IDCNN; CRF

Academic Editor: Anselmo Peñas

Received: 6 November 2024
Revised: 7 December 2024
Accepted: 3 January 2025
Published: 13 January 2025

Citation: Song, W.; Zheng, H.; Ma, S.; Zhang, M.; Guo, W.; Ning, K. Chinese Mathematical Knowledge Entity Recognition Based on Linguistically Motivated Bidirectional Encoder Representation from Transformers. *Information* **2025**, *16*, 42. https://doi.org/10.3390/info16010042

1. Introduction

As computer technology continues to advance and material conditions improve, artificial intelligence (AI) technologies are constantly evolving. Natural Language Processing (NLP) is progressing alongside AI at a rapid pace. NLP technology encompasses all processes of using electronic devices to process natural language. The purpose of this technology is to enable computers to correctly perceive, process, and apply the human language input, thereby achieving many complex functionalities. The NLP technical framework can be divided into three levels: small-scale, including word-level NLP techniques; medium-scale, including syntactic-level NLP techniques; and large-scale, including discourse-level NLP techniques. Named Entity Recognition (NER) is a relatively small-scale branch of NLP, specifically at the global level. Its main function is to identify and extract entity names from sentences or articles, forming the foundation for applications like knowledge graphs, data mining, question-answering systems, and machine translation. Chinese Named Entity Recognition (NER) tasks involve extracting the required entities from Chinese texts. Different recognition tasks focus on extracting different types of entities. For example,

course entity recognition focuses on identifying entities like course names, teacher names, and knowledge point entities, while news entity recognition involves identifying entities like person names and place names.

Mathematical knowledge entity recognition focuses on identifying concepts such as angles, lines, and planes; methods like "right angle", "sequence", and "set"; and theorems like "Pythagorean theorem" and "completing the square". These recognized entities can be used to construct knowledge graphs which support tasks such as knowledge-based question answering and MOOC course recommendations. Taking course recommendation as an example, knowledge point entities are extracted from course descriptions, videos, and other materials to build the course's knowledge graph. From the user's learning activities (such as courses viewed, quizzes, and classroom discussions), their required knowledge is identified, and personalized recommendations are made based on the knowledge graph, improving the user experience and enhancing learning outcomes. Therefore, the accuracy of entity recognition directly affects the effectiveness of these subsequent tasks, making the improvement of recognition accuracy a key research focus.

To address this, the latest pre-trained language model, LERT (Linguistically Motivated Bidirectional Encoder Representation from Transformers), is used to obtain semantically rich word vectors. By combining LERT's powerful language representation capabilities with Bidirectional Gated Recurrent Units (BiGRUs), Iterated Dilated Convolutional Neural Networks (IDCNNs), and Conditional Random Fields (CRFs), the model's ability to capture global contextual information is enhanced, thereby improving the accuracy of entity recognition.

2. Materials and Methods

Over time, Named Entity Recognition (NER) technology has evolved from dictionary-based rule techniques to traditional machine learning and deep learning methods.

Early methods primarily relied on rule-based and dictionary-based approaches. Researchers such as Kim J. H., Riaz K., and Xiaoheng Zhang used this technique in their respective tasks. However, this method depends on specific rules for entity recognition, and the richness of the dictionary is often insufficient, leading to ambiguities between words. The process of constructing rules is complex, requiring researchers to have a deep understanding of linguistic knowledge. Additionally, different languages have different grammatical structures, which means that language-specific rules must be developed. These rules often conflict with each other, requiring careful management. As a result, the workload for researchers increased significantly, as they had to continually revise old word sets and rules, which eventually led to these methods being replaced by more advanced machine learning techniques.

Traditional machine learning methods for Named Entity Recognition (NER) mainly include the Hidden Markov Model (HMM), the Maximum Entropy Markov Model (MEMM), and the Conditional Random Field (CRF). These approaches are primarily based on statistical probabilities and are essentially sequence labeling tasks. They require large corpora for training, where the model learns to label the input language based on the provided data. Zhao [1] applied HMM in the recognition of biomedical texts, achieving an accuracy score of 62.98% using a word-similarity-based smoothing method. Wang and others applied MEMM to address extraction, leading to significant improvements in both precision and recall. Lafferty et al. [2] proposed the CRF, a discriminative classifier that builds models for decision boundaries between different classes and can be used for classification after training. Chen [3] used the CRF for Chinese NER recognition, achieving a score of 85.25 on the MSRA dataset. Khabsa M. [4] applied the CRF to chemical entity recognition and obtained an F1 score of 83.3%. However, these methods heavily depend

on the corpus, requiring careful data selection, processing, and the construction of effective features. The choice of features directly impacts the model's performance, and this process requires considerable human effort and time. Additionally, these methods tend to have slow convergence and lengthy training times, which further adds to the challenge.

With the continuous development of machine learning, a wide variety of models and algorithms have been introduced to solve various problems. Deep learning methods based on neural networks have gradually become dominant in NER tasks. Recurrent Neural Networks (RNNs) [5] have shown great effectiveness in addressing sequence modeling problems. However, RNNs tend to focus more on later outputs, leading to issues like gradient vanishing or exploding. To address this, Hochreiter et al. [6] proposed Long Short-Term Memory (LSTM), which selectively utilizes long-term sequence information through a gating mechanism (the input gate, output gate, and forget gate), retaining useful long-sequence information and mitigating the issues present in RNNs. Zeng D et al. [7] combined LSTM with the CRF for drug entity recognition tasks. On top of LSTM, the Gated Recurrent Unit (GRU) retains two gate structures (update gate and reset gate), reducing the number of parameters in LSTM, effectively lowering training costs and minimizing the risk of overfitting in BiLSTM. By combining forward and backward LSTMs and GRUs, Bidirectional LSTM (BiLSTM) and Bidirectional GRUs (BiGRUs) are created, which capture both preceding and following contextual information in sequences, thereby improving NER performance. Wu et al. [8] applied the BiLSTM-CRF with attention to the Chinese electronic medical record NER. Quinta et al. [9] optimized the BiLSTM-CRF for Portuguese corpora, achieving high F1 results. Qiu Qinjun et al. [10] proposed an attention-based BiLSTM-CRF neural network, achieving an F1 score of 91.47% in geological NER tasks. Convolutional Neural Networks (CNNs), compared to RNNs, are more commonly used in image modeling. In text processing, CNNs may only capture a small portion of the original data through convolutions, and increasing the number of CNN layers to improve accuracy results in a significant increase in parameters, which also increases training costs and leads to overfitting. Emma Strubell et al. [11] proposed the Iterated Dilated Convolutional Neural Network (IDCNN) based on the Dilated CNN (DCNN). As the depth of the DCNN increases, the effective input width expands exponentially, quickly covering the entire length of the input sequence. During dilation, it captures rich local information that BiLSTM and BiGRUs may overlook. The depth of the DCNN network increases linearly, avoiding the exponential growth in parameters that would occur with increasing CNN layers, thus preventing the problem of parameter explosion. The IDCNN iteratively applies the dilated convolution blocks multiple times, without adding extra parameters, effectively mitigating the overfitting issues caused by simply increasing the depth. Yu Bihui et al. [12] achieved an F1 score of over 94% in entity recognition using the IDCNN. Although these methods have achieved some success in the field of NLP, when dealing with phrases or even sentences, they often overlook the semantic relationships between words and their contexts, especially in Chinese, where the same word can have different meanings in different contexts (polysemy). This limits the model's ability to accurately recognize entities, affecting overall recognition accuracy.

The BERT (Bidirectional Encoder Representation from Transformers) [13] model, introduced by Google AI in 2018, is a bidirectional encoder based on the Transformer architecture. BERT significantly enhances the relational features between characters, words, and sentences, allowing us to better understand information in different contexts. The word vectors generated by BERT have much stronger semantic representation capabilities. Additionally, during training, BERT does not require manual intervention from researchers, and different functionalities can be implemented without major modifications to the code framework, which greatly reduces training costs. BERT is a pre-trained language

representation model, and its pre-training phase includes two tasks: a Masked Language Model (MLM) and Next-Sentence Prediction (NSP). These tasks strengthen the model's learning of word and sentence relationships. Gao et al. [14] applied BERT for sentiment analysis and achieved the best results compared to traditional models. Wu Jun et al. utilized BERT embeddings combined with BiLSTM and the CRF for Chinese terminology extraction, demonstrating a clear improvement over traditional shallow machine learning models. Zhang Yi et al. combined BERT with the BiLSTM-IDCNN-CRF, achieving an F1 score of 93.91% for elementary mathematical entity recognition. Yang Chonglong et al. [15] used BERT to build word vectors, followed by BiLSTM, and then improved the IDCNN. CRF decoding was then used to create an excellent COVID-19 entity recognition model. However, BERT has some limitations. The tasks used during pre-training do not appear in downstream tasks, which can lead to a mismatch between pre-training and fine-tuning, potentially affecting BERT's performance in downstream NLP tasks.

Building on BERT, many other pre-trained language models have emerged. Kevin Clark et al. [16] identified that BERT's pre-training learning efficiency was relatively slow and proposed the Electra language model. Electra modifies the MLM strategy by replacing the original tokens with generated tokens, instead of using masking. This approach increases training speed and improves accuracy in downstream tasks. MacBERT [17] replaced the original MLM task with a corrected MLM task, where similar-meaning words are used to replace the original words. Additionally, it changed the NSP task to the Sentence Order Prediction (SOP) task to reduce the gap between BERT's pre-training and downstream tasks. PERT [18] uses the PerLM method, which employs Whole-Word Masking (WWM) and N-gram masking to select words, changing the order of characters and words in sentences. The model's goal is to restore the word order from the shuffled sentence. Recently, Cui Yiming et al. [19] proposed the LERT pre-trained language model, which injects linguistic knowledge during the pre-training phase. Specifically, it uses the LTP language analysis tool to generate the following three linguistic features: Part of Speech (POS) tagging, Named Entity Recognition (NER), and Dependency Parsing (DEP). These features are combined with the Masked Language Model (MLM) task to perform multi-task pre-training. By incorporating linguistic features, LERT possesses more powerful language representation capabilities and aligns more closely with downstream tasks, providing strong support for NLP tasks.

In this experiment, LERT will be used to generate word vectors rich in semantic information. The BiGRU and IDCNN will capture long-range dependencies and global information, while the CRF will decode the entity labels by leveraging the dependencies between labels. Together, these four components form a fast-training and highly accurate mathematical knowledge entity recognition extractor.

2.1. LERT-BiGRU-IDCNN-CRF

2.1.1. Model Architecture

The overall structure of the LERT-BiGRU-IDCNN-CRF model is shown in Figure 1. It is mainly divided into four parts: the LERT pre-trained language model layer, the BiGRU layer, the IDCNN layer, and the CRF layer. LERT is used to obtain dynamic word vectors from the dataset, and the resulting dynamic word vectors are fed into the forward and backward GRU. Through training in the GRU layer, the weights of the feature items containing forward and backward information are inspected, with emphasis on features that play a decisive role or are particularly important for the recognition task, while ignoring some irrelevant or less correlated features. The IDCNN layer extracts the local information that is ignored by the BiGRU, and, finally, the CRF layer uses the Viterbi algorithm for decoding to obtain the text labels.

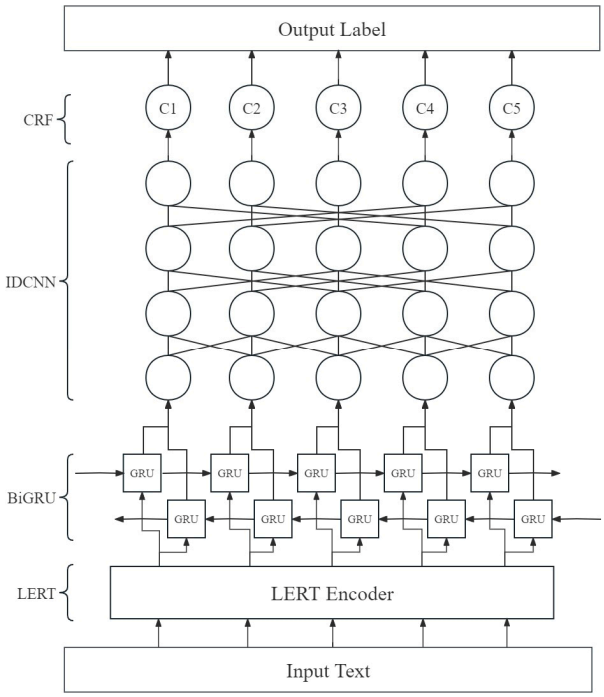

Figure 1. The whole model architecture.

2.1.2. LERT Pre-Trained Language Mode

LERT completes two tasks during the pre-training, one of which is masked language modeling (MLM) and the other is the linguistic task, which includes Part of Speech (POS), Named Entity Recognition (NER), and Dependency Parsing (DEP). The essence of the MLM (Masked Language Model) task is to predict the masked characters by randomly masking certain characters in a sentence, allowing the model to predict what the masked characters are. In the MLM task, LERT adopts Chinese Whole-Word Masking (WWM) and N-gram masking. Unlike the original MLM, WWM masks the entire word instead of breaking it down into subwords. For example, in MLM, the word "摩天楼" (skyscraper) is broken into three subwords: "摩" (mo), "天" (tian), and "楼" (lou). If "天" is selected to be masked, and WWM will also mask "摩" and "楼" as [MASK]. N-gram masking marks a continuous sequence of N words as [MASK]. For instance, in a 2 g scenario, both "世界" (world) and "公园" (park) in the phrase "世界公园" (world park) would be masked entirely. The LTP is used as a boundary tool to divide the words. N-g include 1 g, 2 g, 3 g, and 4 g, with masking probabilities of 40%, 30%, 20%, and 10%, respectively. LERT applies the WWM and N-gram masking methods to replace 15% of the characters in the corpus with the special token [MASK]. Of this 15%, 80% of the characters are replaced with [MASK], 10% are replaced with random characters, and 10% remain unchanged. Examples of replacement are shown in Table 1.

Table 1. Masking and replacement examples.

Percentage	Replacement Examples	
	Before	**After**
80%	Today is Sunday.	Today is [MASK].
10%	Today is Sunday.	Today is fun.
10%	Today is Sunday.	Today is Sunday.

In the pre-training phase of LERT, three linguistic tasks are incorporated: Part of Speech (POS), Named Entity Recognition (NER), and Dependency Parsing (DEP). LERT

first uses the LTP to extract linguistic features (i.e., POS, NER, and DEP) from the training corpus and uses them as weakly supervised labels during the pre-training phase, performing classification training across the three linguistic tasks. The losses from the MLM task and the three linguistic tasks are jointly computed to obtain the final pre-trained LERT language model. This multi-task training significantly enhances LERT's semantic understanding capabilities. POS provides fundamental lexical and syntactic information; this information helps the model more accurately distinguish entity words (e.g., proper nouns) from common nouns, thereby guiding entity boundary detection. Integrating NER learning into the pre-training phase means the model starts building an understanding of entity features, distributions, and contextual patterns before it even fine-tunes on domain-specific data. As a result, during downstream NER tasks, the model no longer needs to learn from scratch what entities are or how to identify them—it already possesses some inherent entity recognition capabilities. Dependency parsing provides information about syntactic structures and dependency relations between words. This higher-level structural knowledge indirectly strengthens the model's grasp of the contexts in which entities appear.

Moreover, with the addition of the NER task in the pre-training phase, LERT outperforms other pre-trained models in the downstream mathematical Named Entity Recognition tasks, which is clearly demonstrated in the subsequent experiments.

Transformer is a sequence-to-sequence (Seq2Seq) task, which is mainly composed of the encoder and the decoder. The length of the input sequences and output sequences of this network can be changed. As shown in Figure 2, LERT contains a multi-layer Transformer architecture, which mainly uses the encoder part of the Transformer, and its structure is shown in Figure 3.

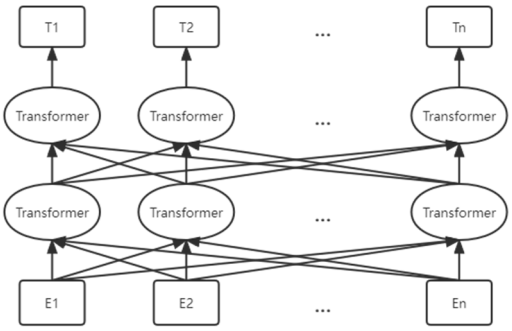

Figure 2. The structure of LERT.

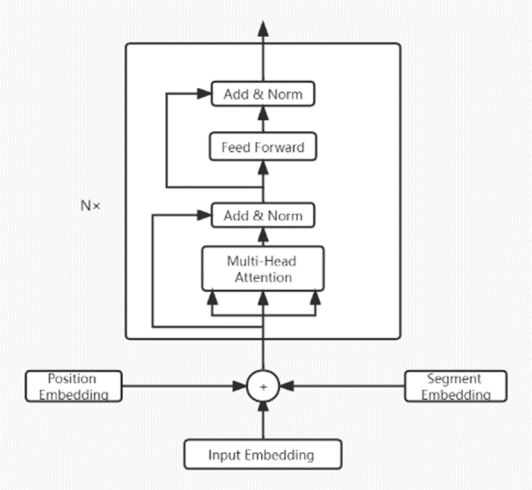

Figure 3. The structure of the encoder.

The initial input e_0 for the single-layer Transformer can be described as (1). After passing through the multi-head attention module, e_0 is added to the output of attention and normalized (Add & Norm) to obtain the intermediate result e_{mid}. After being sent to the fully connected feed-forward neural network (FNN), e_{mid} is normalized again with the output of the FNN to obtain the output e_{out} of this layer. The output of this layer is taken as the input of the next layer and it is passed through the Transformer layer again. At the last, the final output is obtained through the N-layer Transformer and the entire process can be represented as (2)–(4), where e_{m-1} is both the output of the $m-1$ layer and the input of m layer, $m \in [1, n]$, where n is the number of encoder layers.

$$e_0 = Embedding_{token}(\text{in}) + Embedding_{segment}(\text{in}) + Embedding_{position}(\text{in}) \quad (1)$$

$$e_{mid} = LayerNorm(e_{in} + MultiHeadAttention(e_{in})) \quad (2)$$

$$e_{out} = LayerNorm(e_{mid} + FFN(e_{mid})) \quad (3)$$

$$e_m = EncoderLayer(e_{m-1}) \quad (4)$$

2.1.3. BiGRU

The structure of the GRU is shown in Figure 4. The GRU contains two kinds of gating structures: the update gate and the reset gate. The reset gate aims to retain the past information, which is helpful for prediction purposes. The updated gate decides how much input information is required for the new hidden state. In other words, the updated gate can filter out the less useful parts of the input information and keep the most useful part. The input of the GRU consists of two parts: the precious activation h_{t-1} at time $t-1$ and the current input x_t at time t. The gating mechanism can be expressed as (5)–(8):

$$r_t = \sigma(X_t W_{xr} + h_{t-1} W_{hr} + b_r) \quad (5)$$

$$z_t = \sigma(X_t W_{xz} + h_{t-1} W_{hz} + b_z) \quad (6)$$

$$\widetilde{h_t} = \tanh(X_t W_{xz} + (r_t \odot h_{t-1}) W_{hh} + b_h) \quad (7)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t} \quad (8)$$

where w is the parameter matrix, r_t is the set of reset gates, b is the bias, z_t is the updated gate, and $\odot$ is the element-wise multiplication. The σ function maps the data for each element in z_t and r_t to a value in the 0–1 range. When all the elements in r_t are set to 0, the model will discard all the hidden information in the past, leaving only the input information at the current time. On the contrary, when r_t is set to 1, all past information is regarded as useful and the model merges them into the current input.

The GRU transmits information from front to back, but in Chinese NER tasks, the current position not only needs information passed from previous positions but also requires reference to the information that follows. The BiGRU is the combination of the forward GRU and the backward GRU. After training the forward GRU, the sequence is reversed to train the backward GRU, allowing the model to capture information from both directions for better performance in tasks like NER. By combining forward and backward GRUs, the BiGRU is more able to capture the intrinsic connections between the characters at the beginning and end of a sentence, thus enhancing its ability to capture contextual relationships. This approach also effectively reduces the impact of vanishing gradients, significantly improving the model's recognition capabilities.

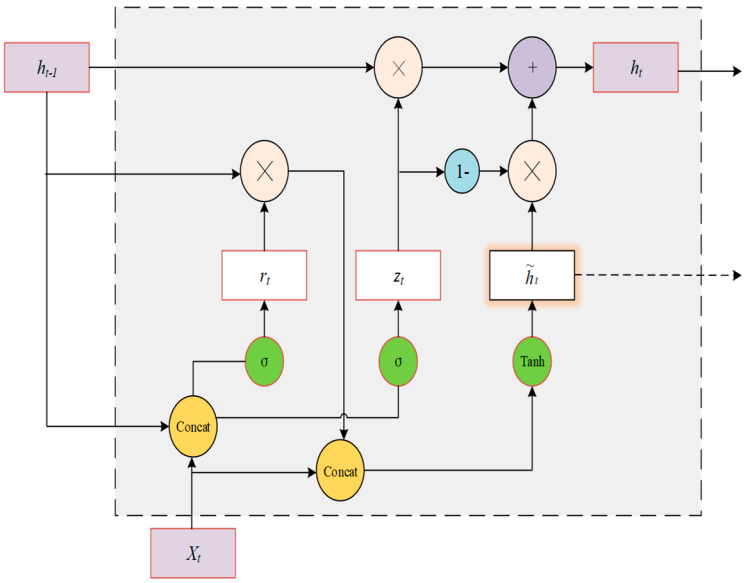

Figure 4. The structure of the GRU unit.

2.1.4. IDCNN

Dilated Convolutional Neural Networks (DCNNs) build upon traditional Convolutional Neural Networks (CNNs) by applying dilation to the convolution kernel, which increases the receptive field—the area in the neural network that a neuron can perceive. In traditional CNNs, the convolution kernel slides across the region, and dilation is introduced to skip the data within the dilation width while keeping the kernel size unchanged during the convolution process. This allows a fixed-size kernel to have a wider data view. On the left side of Figure 5, the original convolutional kernel expands outward with a dilation rate of 1, forming a 3×3 receptive field. In the middle of Figure 5, the kernel expands outward with a dilation rate of 2, forming a receptive field of 7×7. On the right side of Figure 5, the dilation rate is 4, resulting in a receptive field size of 15×5.

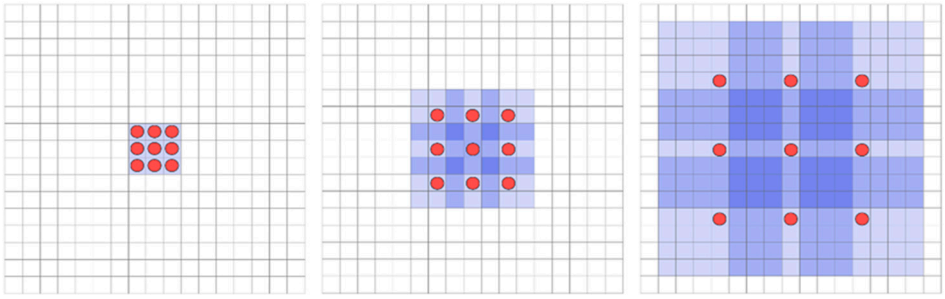

Figure 5. Changes in the CNN expansion. The receptive field in the left image is 3×3; in the middle image, it is 7×7; and in the right image, it is 15×15.

Strubell introduced Dilated Convolutional Neural Networks (DCNNs) into the field of Natural Language Processing (NLP). In Named Entity Recognition (NER) tasks, DCNNs are typically one-dimensional, and they are mostly applied to vector sequences representing token embeddings, rather than two-dimensional grids, as often seen in image processing. This allows the DCNNs to efficiently capture context across a sequence of tokens, adapting a convolutional approach to the needs of sequential data in NLP. This can be expressed as (9).

$$c_t = W_c \oplus x_{t \pm kl} \tag{9}$$

where c_t is the output, x_t is the input, $\oplus$ is the vector concatenation, l is the dilation width, and w_c is the parameter matrix. As the dilation rate increases, the receptive field expands

exponentially, while the number of parameters only increases linearly. This significantly reduces the training cost compared to traditional CNNs.

By stacking more layers in a DCNN, the dilation width increases exponentially, and the receptive field also grows exponentially, while the number of parameters increases linearly. This means that during training, the cost does not grow excessively as the receptive field expands. However, simply increasing the depth of the DCNN can lead to overfitting. To address this, the Iterated Dilated Convolutional Neural Network (IDCNN) uses an iterative method, repeatedly applying the same stack of dilated convolutions, as shown in Figure 6. The output of one iteration serves as the input for the next, allowing the same parameters to be reused in a cyclic manner. An IDCNN consists of multiple dilated convolution blocks, and each block contains multiple layers of the DCNN. This approach helps mitigate overfitting while maintaining the model's ability to capture extensive global information. The whole process can be expressed as (10)–(14):

$$c_t^{(0)} = D_1^{(0)} x_t \tag{10}$$

$$c_t^{(j)} = \mathrm{Re}lu\left(D_{2^{m-1}}^{(j-1)} c_t^{(j-1)}\right) \tag{11}$$

$$b_t^{(1)} = Block(x_t) \tag{12}$$

$$b_t^{(k)} = Block\left(b_t^{(k-1)}\right), k \in [2, n] \tag{13}$$

$$h_t = W_o b_t^{(n)} \tag{14}$$

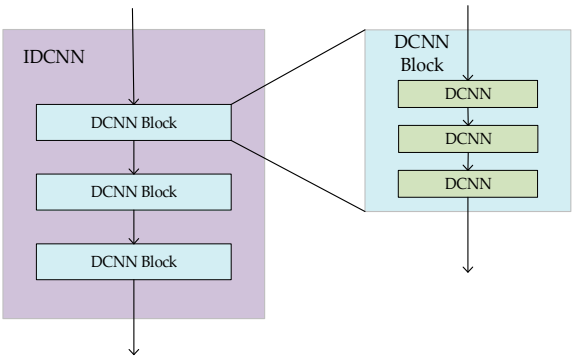

Figure 6. The structure of the IDCNN unit.

The input feature sequence can be represented as $x = [x_1, \ldots, x_t, \ldots, x_T]$, and the output score vector sequence can be represented as h_t. Let the dilation width be y and the x-th DCAN layer be represented as $D_y^{(x)}$. By passing the input through the DCAN layer $D_1^{(0)}$ in the first DCNN block, we obtain $C_t^{(0)}$. A DCNN layer stacked within a DCNN block (where m represents the number of DCAN layers) is then defined. After passing through multiple DCAN layers, we obtain the output $b_t^{(1)}$ of the initial DCAN block (and we define a DCNN block as Block(*)). The output of the previous DCAN block is used as the input for the next layer, and this process is repeated n times. Finally, the output of the entire IDCNN is obtained (where W_o is the parameter matrix). The IDCNN has strong parallel computing capabilities.

2.1.5. CRF

The Conditional Random Field (CRF) is a discriminative probabilistic undirected graphical model. In mathematical NER tasks, the CRF is used to perform sequence labeling on the output from the IDCNN layer. While LERT can capture rich semantic information,

the BiGRU can grasp long-range dependencies within a sentence, and the IDCNN can capture local information; however, none of these three components can effectively learn the dependencies between labels.

In the BIO labeling scheme used in this experiment, there are hidden rules, such as the following: the "I-X" tag cannot appear before the "B-X" tag (for example, the "I-KNOW" tag must follow the "B-KNOW" tag and cannot precede it). Additionally, the "I-X" tag for one entity type cannot appear after the "B-X" tag of another entity type (for instance, the "I-KNOW" tag can only follow the "B-KNOW" tag, not the "B-PRIN" tag, and the same applies for "I-PRIN"). In this aspect, the CRF is highly effective in learning these label dependencies. It uses adjacent labels and input features to produce the globally optimal label sequence.

By letting the input sequence be $x = (x_1, x_2, \ldots, x_n)$ and the corresponding label sequence be $y = (y_1, y_2, \ldots, y_n)$, the conditional probability of the CRF, $P(y|x)$, can be expressed as (15) and (16):

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right) \tag{15}$$

$$Z(x) = \sum_{y} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right) \tag{16}$$

The features λ and u, as well as the corresponding weights t and s, are learned through later stages of training. From Equations (15) and (16), it can be seen that the calculation at the current position is influenced not only by the current input x but also by the previous sequence label y_{i-1}.

3. Results

3.1. Text Annotation

In Named Entity Recognition (NER) tasks, the training corpus needs to be annotated as a sequence. Sequence labeling can generally be divided into two types: raw labeling and joint labeling. In raw labeling, each character is independently labeled with a tag, while in joint labeling, characters belonging to the same type are grouped and labeled together. Typically, raw labeling is used to address joint labeling problems. In Chinese NER tasks, it is essential not only to identify the entity category but also to determine the position of the entity and the boundaries between different entities. Considering this, using raw labeling is more appropriate. IO is the simplest of the three schemes, containing only two tags: "I" and "O". "I-X" indicates a character that is part of an entity ("X" represents the entity type, such as "I-PER" for a person's name), while "O" denotes irrelevant characters that are not part of any entity. This scheme has significant drawbacks because it lacks boundary markers, making it impossible to distinguish between adjacent entities. BIO is a more commonly used labeling scheme, which builds upon IO by adding the "B" tag to indicate the beginning of an entity. It includes "B-X", "I-X", and "O". BIO largely solves the boundary ambiguity issue present in the IO system, making it a well-performing labeling method for raw tasks. BIOES is an extension of BIO that adds an "E-X" tag for the end of an entity and an "S-X" tag for single-character entities. BIOES is more complex than IO and BIO, providing clearer entity boundaries. However, its increased complexity leads to longer training times. The complexity of the labeling system is closely related to the recognition accuracy—the more complex the labeling scheme, the higher the accuracy under the same conditions. However, this also increases the corresponding training time. For the mathematical knowledge entity recognition task in this study, the BIO labeling scheme will be used for sequence labeling.

The dataset consists of sentences and labels, both of which are strings. However, for the model, the input data type must be numeric. For Chinese characters, we can use the character dictionary provided in the open-source BERT pre-trained model package by Google. For labels, we follow the order of entity categories. Within each category, we place B-tags before I-tags. After these, we add the special tokens "<START>" and "<EOS>" to indicate the start and end, respectively. For convenience, we prepend "<PAD>" at the beginning. Finally, the dictionary for the label is shown in Table 2.

Table 2. Label dictionary.

Label	Numeric
<PAD>	0
B-KNOW	1
I-KNOW	2
B-PRIN	3
I-PRIN	4
O	5
<START>	6
<EOS>	7

3.2. Dataset

In this experiment, the dataset contains two types of entities. One type is mathematical concept entities, such as "集合" (set) and "三角形" (triangle), which are labeled as "KNOW". The other type consists of mathematical theorems and laws, such as "贝叶斯定理" (Bayes' theorem) and "格林公式" (Green's theorem), which are labeled as "PRIN". The BIO labeling scheme is used for annotation.

The dataset is divided into the training set, test set, and validation set, with the number of entities in each set shown in Table 3.

Table 3. Dataset entities.

Dataset	TOTAL	KNOW	PRIN
Train	10,100	9733	367
Test	1484	1378	106
Dev	1631	1550	81

3.3. Evaluation Metrics

In this experiment, three evaluation metrics are used: precision (P), recall (R), and the harmonic mean F_1. The formulas for the three metrics are as follows:

$$P = \frac{CT}{CA} \times 100\% \tag{17}$$

$$R = \frac{CT}{TA} \times 100\% \tag{18}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{19}$$

where CT represents the total number of entities successfully recognized by the model, CA represents the total number of all entities recognized by the model, and TA represents the total number of all entities in the standard results.

3.4. Experimental Environment and Parameter Settings

The model for this experiment is built based on the PyTorch framework, and the environment settings are shown in Table 4.

Table 4. Table type style.

Environment	Version
Transformers	4.22.2
GPU	Tesla v100-pcie
Python	3.8
Pytorch	1.12.1 + gpu
Pytorch-crf	0.7.2

The experimental parameter settings are shown in Table 5. The Transformer layers and hidden layer dimensions are set according to the LERT-base parameters, with 12 layers and 768 hidden units. The maximum sentence length inputted into the model is set to 128. Learning rate decay is applied to control changes in the learning rate during training. Weight decay is used to reduce model complexity, and dropout is employed to prevent or mitigate overfitting during training. The convolution kernel size is set to 3×3; the number of dilated convolution blocks is set to 4; and the dilation rates are set to 1, 1, and 2.

Table 5. Table type styles experimental parameter settings.

Parameter	Value
Max length	128
Batch size	16
Hidden	768
Transformer layer	12
Epochs	50
Dropout	0.5
Kernel size	3
Block number	4
Dilation	1,1,2
Kernel number	120

3.5. Result

The training process consists of 50 epochs, but starting from the 30th epoch, the model's accuracy, recall, and F1 scores no longer show significant improvements. Figure 7 shows the accuracy, recall, and F1 scores during the first 30 epochs of training. The model's loss during these 50 epochs is displayed in Figure 8. With each epoch, the model's loss consistently decreases, eventually leveling off and staying below 0.18.

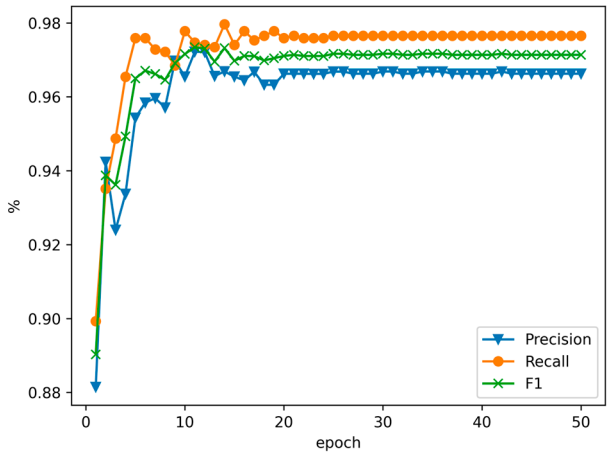

Figure 7. Model effect.

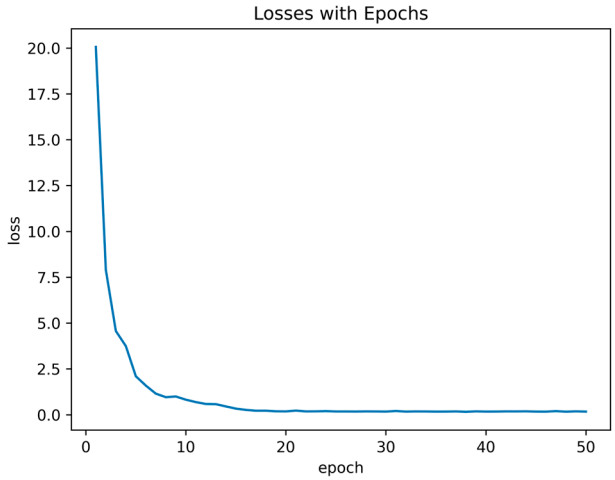

Figure 8. Model loss.

In this process, the best model achieves an accuracy score of 97.22%, a recall score of 97.46%, and an F1 score of 97.34%. When this model is tested on the validation set, the accuracy, recall, and F1 score for the two types of entities are obtained, as shown in Table 6.

Table 6. The evaluation results of the models on the development set.

Entities	Evaluation Metrics		
	Precision	**Recall**	**F1**
KNOW	97.210	97.659	97.434
PRIN	97.436	93.827	95.597

Through conducting tests on the development set, the proposed LERT-BiGRU-IDCNN-CRF model is found to perform well in recognizing mathematical concepts and theorems. Among these, the recognition of mathematical concept entities is the most effective, with 1512 out of 1550 concept entities successfully identified. For theorem entities, 76 out of 81 are correctly recognized, with slightly weaker performance compared to concept entities. The possible reasons for this discrepancy could be as follows:

- Quantity: The larger number of concept entities compared to theorem entities may have led to a difference in performance due to the sample size.
- Data repetition: Certain concept entities appeared more frequently in the dataset, allowing the model to more accurately recognize those repeated entities.
- Entity name length: The names of theorem entities are often longer, which increases the difficulty and affects the accuracy of entity recognition.

To clearly demonstrate the effectiveness of the LERT-BiGRU-IDCNN-CRF model used in this experiment, this section compares the experimental model with various other models.

First, the performance of the LERT component, which constructs word vectors, is tested. Four different pre-trained models for word vector construction are introduced: BERT, RoBERTa, MacBERT, and PERT. Each of these models is used to build a "-BiGRU-IDCNN-CRF" model, all using the base version to ensure consistent network parameters.

- BERT, proposed by Google AI, is a pre-trained language model with two key tasks in the pre-training phase: the masked language model (MLM) task and next-sentence prediction (NSP).

- RoBERTa is a fine-tuned version of BERT, with increased model parameters, larger training datasets, and the ability to remove the NSP task. It uses a dynamic masking strategy for MLM.
- MacBERT modifies the MLM task, using Whole-Word Masking (WWM) and N-gram methods to select masking candidates, replacing them with synonyms instead of using [MASK].
- PERT eliminates the NSP task and shuffles word order during pre-training, with the prediction target being the original word order.

Table 7 shows the best results of different pre-trained models, and Figures 9–11 display the overall performance of these models across various metrics.

Table 7. The performance of different pre-trained models.

Model	Evaluation Metrics		
	Precision	**Recall**	**F1**
LERT	97.221	97.468	97.344
BERT	96.244	96.232	96.238
RoBERTa	96.398	96.232	96.315
MacBERT	96.232	96.664	96.448
PERT	96.213	97.900	97.049

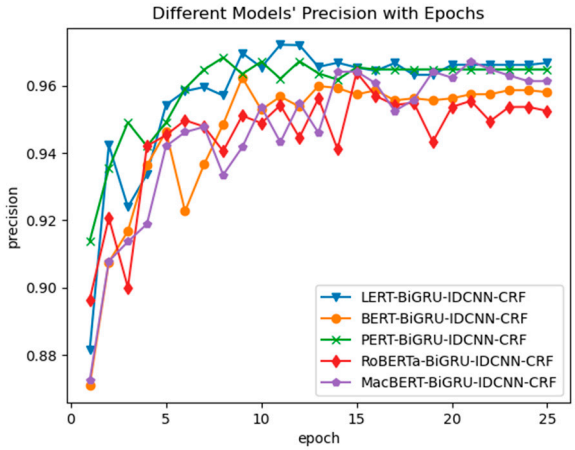

Figure 9. Precision comparison of the models.

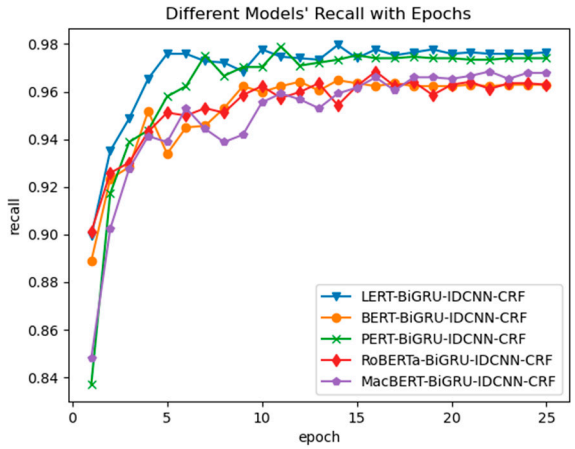

Figure 10. Recall comparison of the models.

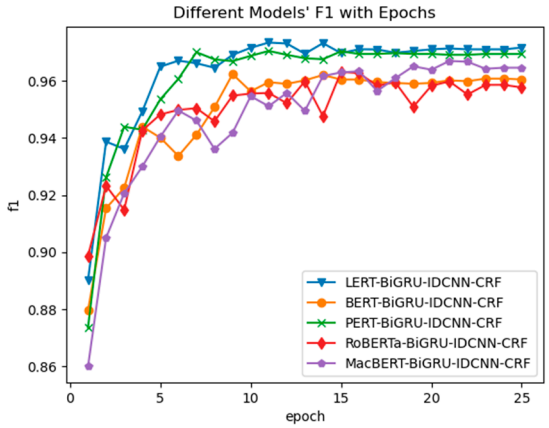

Figure 11. F1 comparison of the models.

It can be observed that the LERT pre-trained language model, which incorporates three linguistic tasks, outperforms the other four pre-trained language models in mathematical knowledge entity recognition. The recognition performance of MacBERT and PERT is better than that of BERT and RoBERTa, indicating that improved pre-training strategies can significantly enhance the connection between the pre-training phase and the downstream entity recognition tasks, thereby improving the recognition results.

Next, the BiGRU and BiLSTM are tested by comparing the BiGRU model with the BiLSTM model (while retaining the IDCNN and CRF, and conducting a horizontal comparison of different word vector models), as shown in Table 8. The recognition performance of the BiGRU and BiLSTM is almost equivalent, with BiGRU models using LERT and PERT showing better recognition results than BiLSTM, while BiGRU models using BERT, RoBERTa, and MacBERT perform worse than BiLSTM. However, a common point is that the training time for the BiGRU model is shorter than that for BiLSTM, as shown in Figure 12.

Table 8. Comparison between the BiGRU and BiLSTM.

	Model	Precision	Recall	F1
BiGRU	LERT	97.221	97.468	97.344
	BERT	96.244	96.232	96.238
	RoBERT	96.398	96.232	96.315
	MacBERT	96.232	96.664	96.448
	PERT	96.213	97.900	97.049
BiLSTM	LERT	96.106	96.726	96.415
	BERT	95.695	96.603	96.147
	RoBERT	95.756	96.726	96.239
	MacBERT	96.248	96.850	96.548
	PERT	95.141	96.047	95.592

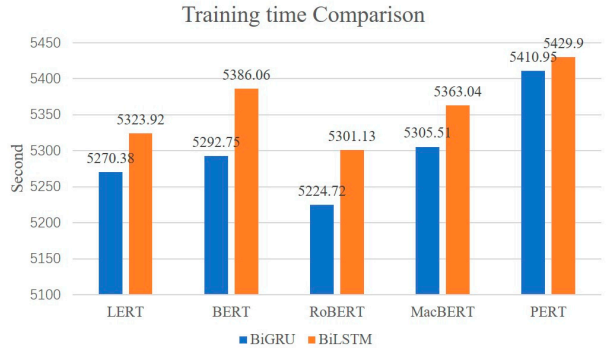

Figure 12. Comparison of training times between the BiGRU and BiLSTM.

Next, the IDCNN is tested through ablation experiments. The results are shown in Table 9, where the BiGRU and CRF are retained, and different word vector models are used for horizontal comparison. The results clearly show that the models incorporating the IDCNN significantly outperform those without the IDCNN. This demonstrates that the IDCNN's ability to capture global information can effectively improve the overall recognition performance of the model.

Table 9. Comparison of IDCNNs.

IDCNN	Model	Precision	Recall	F1
reserve	LERT	97.221	97.468	97.344
	BERT	96.244	96.232	96.238
	RoBERT	96.398	96.232	96.315
	MacBERT	96.232	96.664	96.448
	PERT	96.213	97.900	97.049
remove	LERT	95.993	96.788	96.389
	BERT	95.746	96.232	95.989
	RoBERT	95.174	96.912	96.035
	MacBERT	95.708	96.726	96.215
	PERT	95.575	96.047	95.810

4. Discussion

This paper constructs the LERT-BiGRU-IDCNN-CRF model to perform Chinese Named Entity Recognition (NER) for mathematical concepts and theorems, using the BIO labeling scheme to create the dataset. The dataset is fed into the LERT model, where LERT learns the dependencies between characters and sentences. Internally, the Transformer-based encoder captures word vectors enriched with contextual information, which are then passed to the BiGRU module. The BiGRU checks the weights for both forward and backward information, with each GRU cell using gating mechanisms like the reset gate and update gate to control the amount of information. The IDCNN complements the BiGRU by capturing local information, while the CRF takes into account both the current input sequence and the previous label information to determine the optimal label at each step and perform decoding, ultimately producing the entity results. After testing, this model achieves good performance in recognizing mathematical knowledge entities, with accuracy, recall, and F1 scores of 97.22%, 97.47%, and 97.34%, respectively. Comparative experiments highlight the advantages of this model. The combination of LERT, BiGRU, IDCNN, and CRF allows the model to fully capture global information and intrinsic relationships, making the LERT-BiGRU-IDCNN-CRF model more valuable than other recognition models. It can effectively support downstream tasks like knowledge graphs, question-answering systems, and recommendation systems. In the future, efforts should be made to increase the size of the training set and fine-tune the training parameters to further enhance the model's performance. Additionally, the model can be applied to different datasets and domains, such as medicine, transportation, and technology, expanding its application scope and enabling entity recognition for specialized terms in various fields.

Author Contributions: Conceptualization, W.S. and H.Z.; methodology, H.Z.; software, H.Z.; validation, H.Z., W.G. and K.N.; formal analysis, H.Z. and W.G.; investigation, H.Z. and K.N.; resources, W.S. and H.Z.; data curation, H.Z. and S.M.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z., S.M. and M.Z.; visualization, H.Z.; supervision, W.S.; project administration, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: I would like to express my heartfelt gratitude to all collaborators who played a significant role in conducting this research. Special thanks goes to Wei Song for providing guidance in conceptualization and essential resources, and to Wei Guo, Keqing Ning, and Shuaiqi Ma for their contributions to validation. I am also grateful for Wei Guo's expertise in formal analysis and Keqing Ning's thorough efforts in investigation. Shuaiqi Ma's assistance in data curation and valuable input during the review and editing process were instrumental in shaping this work. Additionally, I thank Wei Song for his ongoing supervision throughout the project. I acknowledge all authors for their invaluable contributions, and we confirm that each has reviewed and approved the final manuscript version.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhao, S. Named entity recognition in biomedical texts using an HMM model. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA/BioNLP), Geneva, Switzerland, 28–29 August 2004.
2. Lafferty, J.; McCallum, A.; Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML 2001, Williamstown, MA, USA, 28 June–1 July 2001.
3. Chen, W.; Zhang, Y.; Isahara, H. Chinese named entity recognition with conditional random fields. In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Sydney, Australia, 22–23 July 2006.
4. Khabsa, M.; Giles, C.L. Chemical entity extraction using CRF and an ensemble of extractors. *J. Cheminform.* **2015**, *7*, S12. [CrossRef]
5. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
6. Hochreiter, S. *Long Short-Term Memory*; Neural Computation MIT-Press: Cambridge, MA, USA, 1997.
7. Zeng, D.; Sun, C.; Lin, L.; Liu, B. LSTM-CRF for drug-named entity recognition. *Entropy* **2017**, *19*, 283. [CrossRef]
8. Wu, G.; Tang, G.; Zhang, Z.; Wang, Z. An attention-based BiLSTM-CRF model for Chinese clinic named entity recognition. *IEEE Access* **2019**, *7*, 113942–113949. [CrossRef]
9. de Castro, Q.; Vitor, P.; da Silva, N.F.F.; da Silva Soares, A. Portuguese named entity recognition using lstm-crf. In *Computational Processing of the Portuguese Language: 13th International Conference, PROPOR 2018, Canela, Brazil, 24–26 September 2018*; Proceedings 13; Springer International Publishing: Berlin/Heidelberg, Germany, 2018.
10. Qiu, Q.; Wu, L.; Tao, L.; Li, W. BiLSTM-CRF for geological named entity recognition from the geoscience literature. *Earth Sci. Inform.* **2019**, *12*, 565–579. [CrossRef]
11. Strubell, E.; Verga, P.; Belanger, D.; McCallum, A. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv* **2017**, arXiv:1702.02098.
12. Yu, B.; Wei, J. IDCNN-CRF-based domain named entity recognition method. In Proceedings of the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Weihai, China, 14–16 October 2020; IEEE: Piscataway, NJ, USA, 2020.
13. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
14. Gao, Z.; Feng, A.; Song, X.; Wu, X. Target-dependent sentiment classification with BERT. *IEEE Access* **2019**, *7*, 154290–154299. [CrossRef]
15. Yang, C.; Sheng, L.; Wei, Z.; Wang, W. Chinese named entity recognition of epidemiological investigation of information on COVID-19 based on BERT. *IEEE Access* **2022**, *10*, 104156–104168. [CrossRef]
16. Clark, K. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
17. Cui, Y.; Che, W.; Liu, T.; Qing, B.; Yang, Z. Pre-training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3504–3514. [CrossRef]
18. Cui, Y.; Yang, Z.; Liu, T. PERT: Pre-training BERT with permuted language model. *arXiv* **2022**, arXiv:2203.06906.
19. Cui, Y.; Che, W.; Wang, S.; Liu, T. Lert: A linguistically-motivated pre-trained language model. *arXiv* **2022**, arXiv:2211.05344.