Leveraging Attention Mechanism and Class Weighting for Legal Event Detection in Chinese Text

Jinhong Hu, Shaidah Jusoh*
School of Artificial Intelligence and Robotics,
Xiamen University Malaysia, Sepang, Selangor, Malaysia

***Abstract*—Event detection is an information extraction task that involves extracting specified event types from textual sequences. Currently, most event detection studies focus on English corpora; there is a lack of exploration in other linguistic contexts. Thus, a study on event detection in the Chinese corpus is essential. Sequence-based event detection has been extensively studied in the past, and many studies have utilized high-performance neural network models, such as traditional recurrent neural networks. This study aims to enhance the performance of sequence models by altering the base model, Bidirectional Long Short Term Memory (BiLSTM), to a Bidirectional Gated Recurrent Unit (BiGRU) and incorporating multi-head attention mechanisms, Conditional Random Fields (CRF), and class weights. These modifications not only improve the model's accuracy but also enhance computational efficiency by reducing the number of parameters relative to large pre-trained models such as Bidirectional Encoder Representations from Transformers (BERT). The experimental findings demonstrate that the proposed model's modification achieves an F1 Score of 83.55 for the micro standard and 78.07 for the macro standard. This presents a substantial improvement over the baseline, delivering performance nearly on par with state-of-the-art BERT-based models on the same dataset, while requiring significantly fewer parameters.**

Keywords*—Event detection; information extraction; Chinese corpus; recurrent neural network

I. Introduction

In the era of rapid advancements in big data and artificial intelligence, linguistic texts have become the foundation of nearly all the information available on the internet. Consequently, there is a growing demand for extracting more meaningful insights from these texts. Information extraction is a technique within the field of natural language processing (NLP) that aims to uncover valuable information from text. Event detection is a crucial subtask of information extraction, focusing on the identification of events mentioned in textual data. Due to the unique syntactic structures of different languages, the handling of details in natural language tasks varies. Despite the advancement of neural network technology and years of development, event detection still faces multiple challenges, including contextual dependency, event diversity, and sparse annotated data. These challenges constrain the accuracy and generalization ability of the model, particularly in complex contexts [1]. In response to these challenges, there have been emerging new approaches aimed at addressing event extraction problems, primarily categorized into two major classes: generation-based and classification-based methods. Some specific techniques have improved models, such as models based on attention mechanisms, transfer learning, and methods integrating external knowledge, which have, to some extent, alleviated the difficulties faced in event extraction [2], [3].

The motivation of this study is to develop an event detection model for Chinese legal texts. The benchmark dataset, LEVEN, a large-scale Chinese legal event detection dataset released in 2022 [4] and currently accessible through the GitHub site, has been utilized. This dataset is currently the largest of its kind for event detection in the Chinese legal domain, comprising approximately 60,000 sentences and 108 different event types. Despite the dataset's focus on the legal domain and its substantial sample size, high-performing sequence labeling models such as BERT and BiLSTM typically achieve F1 scores of around 80% on this dataset [4]. Thus, the primary objective of this study is to improve the performance of the model on the LEVEN dataset by refining the framework of the BiLSTM-based sequence labeling model. Specifically, the study explored which key modifications can effectively enhance the model's performance on this dataset.

This study set boundaries and focused exclusively on mathematical knowledge and model code implementation related to artificial intelligence and NLP. Mathematical knowledge was employed to elucidate the underlying principles of the models, while the implementation of model code utilized advanced API frameworks. A total of nine different models based on BiLSTM were implemented and trained. The enhancements on BiLSTM include modifying the basic architecture to BiGRU, introducing different multi-head attention mechanisms, replacing softmax with CRF, and adding weight adjustment to classes based on data distribution. The enhancement resulted in several beneficial improvements. Although it is possible to train more models, doing so would require an amount of time for parameter adjustment and training, yet it would not guarantee superior outcomes. Additionally, this study does not involve cross-lingual event extraction or the processing of multimodal data. It solely focuses on events within a specific linguistic context. And the enhancement of the model primarily stems from both the dataset and the model itself. All implemented models were analyzed against one another to identify the most effective approach.

This study is structured as follows: Section II addresses related topics, Section III outlines the methodology and experimental design, Section IV provides the analysis of results, and Section V concludes the study.

II. Related Topics

Topics related to this study are briefly presented in this section. Firstly, sequence labeling-based event detection mod-

*Corresponding author.

els are introduced, and relevant examples are provided. Next, the impact of data imbalance in corpus datasets are addressed, along with methods for handling it. Classic sequence models such as the Recurrent Neural Network (RNN) and its variations, including bidirectional models, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) are explored. Finally, attention mechanisms and Conditional Random Fields (CRF) in sequence labeling are discussed.

A. Sequence Labeling

Sequence labeling is a text processing task for assigning appropriate labels to each token in an input sequence [5]. These labels can be used to represent information about parts of speech, named entities, emotional tendencies, and word sense disambiguation. It has wide applications involving entity recognition, part-of-speech tagging, sentiment analysis, entity relationship extraction, and syntactic parsing.

Event detection can be considered as a sequence labeling task, where the model processes a text sequence to generate a corresponding event label sequence. Essentially, it involves detecting trigger words with specified event labels in the text sequence. Multiple event trigger tokens may appear in a single input sentence [6]. In the example shown in Fig. 1, the input sequence is “John opens door, then sleeps”, and the output sequence highlights two trigger words, specifically the first and the sixth tokens, “open” and “sleep”. With these two trigger words and contextual information, the model can determine the corresponding event types as “open” and “sleep”.

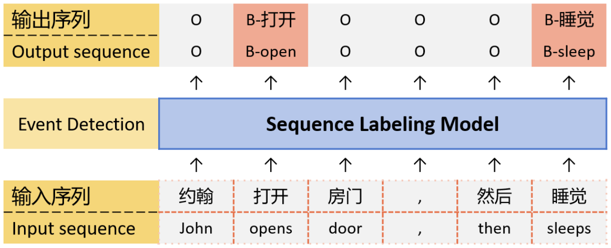


Fig. 1. Event detection based on sequence labeling.

B. Data Imbalance

Data imbalance issues are quite common in corpus datasets. In the event detection task, data imbalance refers to a large difference in the number of samples across event classes. It negatively impacts model training, as the model tends to fit the dominant classes while neglecting the less frequent classes. To address data imbalance issues in NLP, many methods have been considered, like sampling, which increases the number of samples in the minority classes [7]. Alternatively, setting sample weights assigns higher weights to imbalanced classes, encouraging the model to learn the minority classes better [8].

C. Recurrent Neural Network

A recurrent neural network is a neural network model for handling sequential data by recurrent connections, which allows information to persistently propagate within the network [9]. Due to its capability to process sequences of varying lengths, RNN excels in solving event detection based on sequence labeling.

1) Bidirectional: In a traditional RNN, only forward input can be accepted, resulting in the output being more influenced by the most recent input data, which hinders its effectiveness in processing contextual information. To address this issue, the Bidirectional Recurrent Neural Network (BRNN) model was introduced. BRNN can utilize both past and future context information in two directions when processing sequential input [10]. This architecture is beneficial in alleviating the long-range dependency problem and has improved RNN‘s performance.

2) Long Short-Term Memory: LSTM is another variant of RNN, which was designed to address the issues of gradient vanishing and exploding gradients [11]. LSTM achieves better performance in capturing long-term dependency by introducing a mechanism that involves gates and cell states to maintain and control internal memory. These three gates include the input gate, the forget gate, and the output gate.

3) Gated Recurrent Unit: GRU is also a variant of RNN that emerged after LSTM. It aims to address the issue of long-term dependency. The gate mechanism of GRU includes the update gate and reset gate, which control the flow and retention of information [12]. Comparing gate mechanisms, the structure of GRU is simpler than that of LSTM.

D. Self-Attention Mechanism

The attention mechanism was introduced to enhance the model’s ability to process sequential data in NLP [13]. It is also useful for event detection models based on sequence labeling. The key idea is that the model can dynamically focus on different parts of the input, rather than distributing fixed weights when processing an input sequence. One of the most popular attention mechanisms is the self-attention mechanism. This mechanism is currently widely used in various sequence-to-sequence tasks such as machine translation and text generation [14]. It enables the model to capture context and dependencies within the sequence better.

E. Conditional Random Field

For event detection models based on sequence labeling, softmax can be used in the inference layer to obtain final label sequences. It is a common normalization exponential function used to transform a set of values into a probability distribution, where each value represents a potential class score. However, the probability labels obtained by softmax at different positions are treated as mutually independent, which may cause a negative impact. Conversely, CRF, as an undirected probabilistic graphical model, is capable of efficiently capturing label dependencies and contextual information compared with softmax in sequence labeling [15]. Linear CRF is a frequently employed choice in sequence labeling tasks [16].

III. Methodology

This study has been conducted using the programming language Python for data processing and model training. The data processing involves extracting the required data from JSON files through the Python JSON library, and Chinese text tokenization is performed by Jieba (Chinese NLP library) [17]. PyTorch was used to develop neural networks and perform training. The research phases are illustrated in Fig. 2. The flow begins with data gathering, continues through data processing, model implementation, and experimentation, and concludes with result analysis.

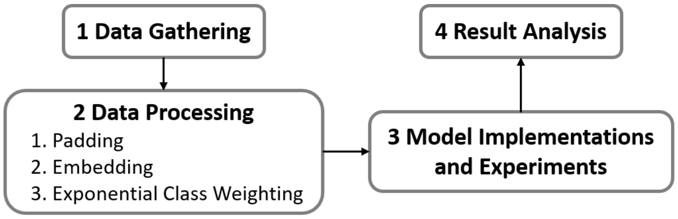
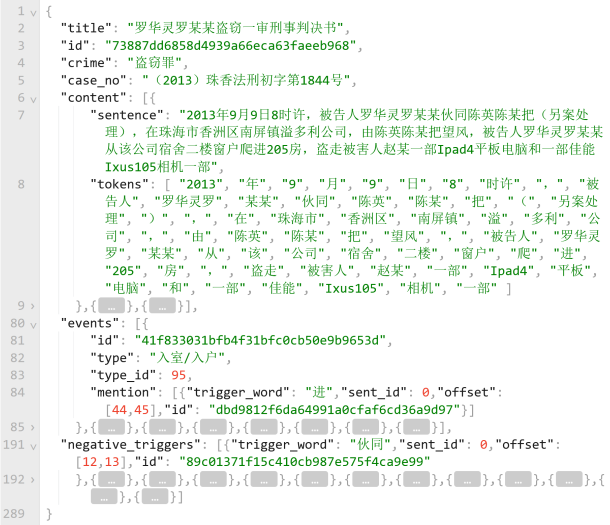


Fig. 2. Flowchart of the research framework.

A. Data Gathering

The LEVEN Chinese legal event detection dataset was obtained, comprising roughly 60,000 training samples across 108 legal event types. Each sample is a complete and lengthy sentence that contains multiple event triggers [18]. Fig. 3 demonstrates a sample of a JSON file that contains the title of the text, the unique ID, as well as the content of the text, which includes multiple sentences with corresponding tokens.



```
{
  "title": "罗华灵罗某某盗窃一审刑事判决书",
  "id": "73887dd6858d4939a66eca63faeeb968",
  "crime": "盗窃罪",
  "case_no": "（2013）珠香法刑初字第1844号",
  "content": [{
      "sentence": "2013年9月9日8时许，被告人罗华灵罗某某伙同陈英陈某把（另案处
        理），在珠海市香洲区南屏镇溢多利公司，由陈英陈某把望风，被告人罗华灵罗某某
        从该公司宿舍二楼窗户爬进205房，盗走被害人赵某一部Ipad4平板电脑和一部佳能
        Ixus105相机一部",
      "tokens": [ "2013", "年", "9", "月", "9", "日", "8", "时许", "，", "被
        告人", "罗华灵罗", "某某", "伙同", "陈英", "陈某", "把", "（", "另案处
        理", "）", "，", "在", "珠海市", "香洲区", "南屏镇", "溢", "多利", "公
        司", "，", "由", "陈英", "陈某", "把", "望风", "，", "被告人", "罗华灵
        罗", "某某", "从", "该", "公司", "宿舍", "二楼", "窗户", "爬", "进",
        "205", "房", "，", "盗走", "被害人", "赵某", "一部", "Ipad4", "平板",
        "电脑", "和", "一部", "佳能", "Ixus105", "相机", "一部" ]
    },{…},{…}],
  "events": [{
      "id": "41f833031bfb4f31bfc0cb50e9b9653d",
      "type": "入室/入户",
      "type_id": 95,
      "mention": [{"trigger_word": "进","sent_id": 0,"offset":
        [44,45],"id": "dbd9812f6da64991a0cfaf6cd36a9d97"}]
    },{…},{…},{…},{…},{…},{…},{…}],
  "negative_triggers": [{"trigger_word": "伙同","sent_id": 0,"offset":
    [12,13],"id": "89c01371f15c410cb987e575f4ca9e99"
    },{…},{…},{…},{…},{…},{…},{…},{…},{…},{…},{
      …},{…}]
}
```

Fig. 3. An example of training data in the JSON file.

The sentence in the sample contains a close number of negative and positive triggers. During the training process, a single sentence is used as one piece of data instead of multiple sentences. The aim is to reduce the length of the input sequence and alleviate the model's computational burden.

B. Data Processing

Tokenization is an initial step in data processing. This step has been employed for both training data and testing data. It is worth noting here that the LEVEN dataset is exceptionally clean and does not require any cleaning operations. To ensure the completeness of the data information input to the model, we do not employ stop word removal, as stop words may have some semantic information for Chinese. Padding, embedding, and exponential class weighting utilized in the proposed models will be discussed in the following subsections.

1) Padding: In this study, the sequence labeling model requires that the input data fed into the model have the same length. As shown in the length distribution chart of the data samples in Fig. 4, sentence lengths range from 25 to 350 tokens, with most clustering around 50 and only a few exceeding 150. In the experiment, the model's input length was set to the maximum length observed in the dataset, and shorter samples were padded to reach this specified length.

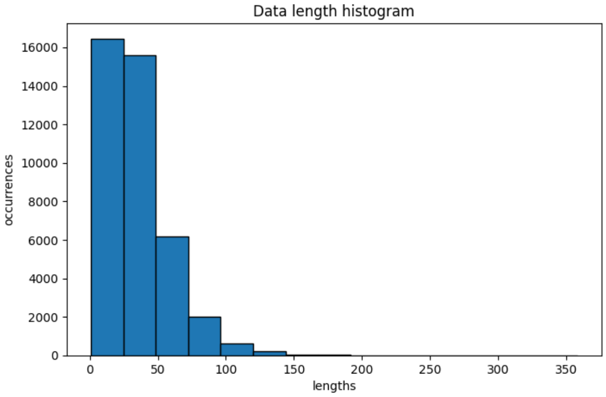


Fig. 4. Data length distribution.

Readers might wonder whether setting the input length to the maximum significantly increases training time or reduces performance. In PyTorch's RNN-based models, however, the actual length of each input can be specified, and computations for the padding portion are ignored, thereby preventing additional training overhead. Furthermore, since the test data also contains longer samples, the padding length was set to the maximum to ensure that no test input would be truncated. In sequence labeling tasks, truncation directly results in the model predicting only the first half of the data.

2) Embedding: The word embeddings used in this experiment were not trained from scratch; instead, pre-trained word embeddings [19] were utilised. These pre-trained word embeddings were trained on Baidu Encyclopedia data, and a vocabulary of 60,000 tokens was included, with each token having an embedding dimension of 300. Once the word embedding matrix is converted into a NumPy array, it occupies approximately 700 MB of memory.

For each processed text data with completed padding, it tries to look up in the word embedding matrix to obtain the corresponding word vector. Rather than pre-converting all training data into word vector format and loading it into memory, which would be impractical due to the significant space requirements, this study uses the PyTorch DataLoader API. This allows the model to dynamically load and pre-process data as needed during the training process. It's important to note that in this experiment, the pre-trained word embedding matrix was not frozen but was adjusted concurrently during the model training process.

3) Exponential class weighting: In the training data set, positive samples accounted for only 24.8% of the total samples. Negative samples refer to the instances where the event did not occur, and the positive samples encompass 108 event classes. The training dataset exhibits a pronounced class imbalance, where the class with the fewest samples contains only 5 instances, while the category with the most samples contains around 6000, as shown in Fig. 5.

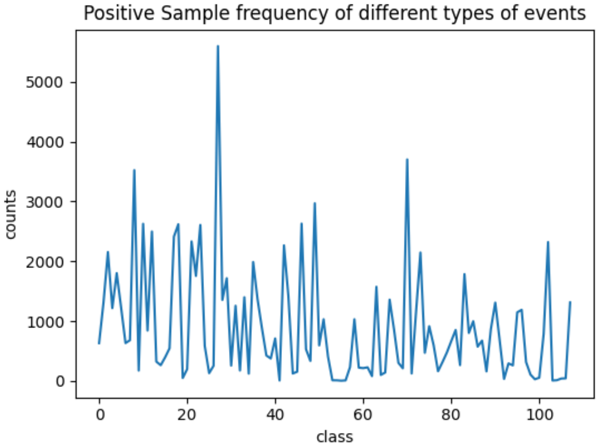


Fig. 5. Distribution of samples across different classes.

To balance the training data, it is necessary to adjust the weights of different data categories. In this experiment, an exponential weighting of class sample quantities was employed, which is defined as follows:

$$w_i = {c_i}^{-\alpha} = \frac{1}{{c_i}^{\alpha}} \tag{1}$$

where, w_i is the weight of a specific class, c_i is the sample quantity of the class and α is a hyperparameter ranging from 0 to 1. From the given formula, it can be observed that if a class has a larger sample quantity, it will receive a smaller weight, while classes with fewer samples will receive larger weights. This allows the model to pay more attention to classes with fewer samples during training. When choosing the value of α, a balanced consideration is required. If α approaches 0, the weights for all classes tend to be close to 1, effectively resulting in no weight adjustment. If α approaches 1, classes with fewer samples will receive significantly larger weights compared to classes with more samples, potentially leading to overfitting. Considering the experiment's requirements, α was ultimately set to 0.1 to balance the model's focus on different class samples. Thus, in this study, α is set to 0.1; the larger the sample size of a class, the smaller its weight.

C. Model Implementations and Experiments

A total of nine models (see Table I) with various enhancements based on BiLSTM were implemented and experimented with in this study. These enhancements include modifying the basic architecture to BiGRU, introducing different multi-head attention mechanisms, replacing softmax with CRF, and adding weight adjustment to classes based on data distribution. Each model employed dropout [20]. The dropout rate was set to 0.3 or 0.4, adjusted based on the degree of model overfitting. The loss function is the multi-class cross-entropy loss function. The models were all compiled using the Adam optimizer [21].

Table I provides information on the nine models. The inference layer of BiLSTM and BiGRU is softmax, while in other models, softmax was replaced with CRF to enhance the dependency information between labels. The A+BiGRU+CRF model introduced the first version of attention, which is self-attention after the embedding layer, before the BiGRU layer. The BiGRU+A+CRF model introduced the second version of attention, which is self-attention after BiGRU, before the CRF. The BiGRU+EA+CRF introduced the third version of attention, which is multi-head attention, after the BiGRU, before the CRF. In BiGRU+A+CRF+W, the W represents the utilization of exponential class weights.

TABLE I. Names and Details of Models

No.	Model	Detail
1	BiLSTM	BiLSTM Model+Softmax
2	BiGRU	BiGRU Model+Softmax
3	BiLSTM+CRF	BiLSTM Model+CRF
4	BiGRU+CRF	BiGRU Model+CRF
5	A+BiGRU+CRF	First Version Attention+BiGRU+CRF
6	BiGRU+A+CRF	BiGRU+Second Version Attention+CRF
7	BiGRU+EA+CRF	BiGRU+Third Version Attention+CRF
8	BiGRU+A+CRF+W	BiGRU+A+CRF with Class Weight
9	BiGRU+EA+CRF+W	BiGRU+EA+CRF with Class Weight

All models used the same dataset. The training dataset comprises 395,322 samples, while the validation dataset contains 92,451 samples, representing approximately 20% of the total data. In this experiment, the default validation dataset served as the testing set and did not participate in model training. The graph in Fig. 6 illustrates the percentage distribution of positive samples from different classes in the training and testing datasets. The class distribution in the training and testing sets is very similar, ensuring an effective evaluation of the model's performance in this experiment.

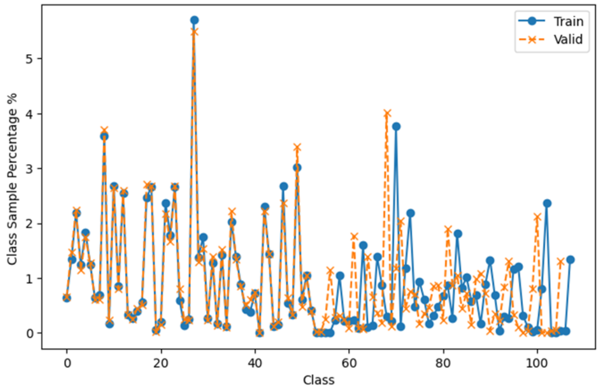


Fig. 6. Percentage distribution of different classes in the training and validation datasets.

The pipeline of the BiGRU+EA+CRF model is shown in Fig. 7. First, the input tokens are passed through the embedding layer to obtain word vectors. Subsequently, the word vectors pass through the BiGRU layer to obtain hidden vectors. Next, in the multi-head attention layer, the query is the word vectors before BiGRU, and the key and value are the hidden vectors after BiGRU. This attention process allows hidden vectors to adjust information with word vectors. Finally, vectors weighted by attention pass through the CRF layer to generate the label sequence. Three different versions of multi-head attention mechanisms have been employed in experiments. The first version is self-attention after the embedding layer, before the BiGRU layer.

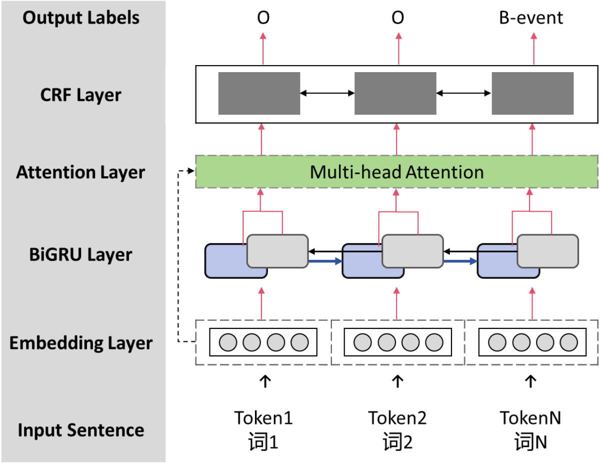


Fig. 7. Framework of BiGRU+EA+CRF model.

The second version is self-attention after BiGRU before the CRF. Placing a self-attention layer after the embedding layer and before the BiGRU layer aims to create a more informed and contextually rich representation of the input sequence early in the network. This allows the BiGRU layer to focus on modeling temporal and sequential dependencies without the additional burden of capturing long-range relationships, potentially leading to improved performance on various NLP tasks.

The third version is multi-head attention after BiGRU before the CRF, while the query is word vectors before the BiGRU layer, but key and value are the hidden vectors after the BiGRU layer. By adding a multi-head attention layer after the BiGRU layer, representation learning would be enhanced. This increases the model's ability to capture diverse and complex dependencies and improve its performance.

IV. Result Analysis

A. Evaluation

To evaluate the results obtained from the experiments, precision, recall, and F1-score based on micro and macro standards were used, which are defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\text{-}score = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

TP as True Positive is the total number of correctly predicted positive samples. FP as False Positive is the total number of incorrectly predicted positive samples. FN as False Negative is the total number of incorrectly predicted negative samples. Micro refers to micro-averaging, which calculates metrics by aggregating results across all classes. Macro refers to macro-averaging, which calculates metrics separately for each class.

TABLE II. Confusion Matrix for Binary Classification Problem

Confusion Matrix		Actual Values	
		1	0
Model Predictions	1	TP	FP
	0	FN	TN

Table II illustrates a confusion matrix, where the rows represent the model's predicted values and the columns represent the true label values of the samples.

B. Analysis

To analyze the obtained results, the performance of the nine models was compared and presented in Table III. The performance of BiLSTM and BiGRU is extremely close by both micro and macro standards. Similarly, the performance between BiLSTM+CRF and BiGRU+CRF shows similar results. This demonstrates that, despite the structural differences between BiLSTM and BiGRU, only a small performance discrepancy was observed in the experiments. By comparing the performance of the BiLSTM and BiGRU models with and without CRF, it is clear that CRF results in a slight increase in precision, but a decrease in recall, while the F1-scores remain unchanged. This indicates that precision is enhanced, but recall is compromised by CRF, without any overall performance loss. Consequently, CRF can be considered beneficial in contexts where precise identification of event types is required while maintaining overall performance.

Among the three versions of attention mechanisms integrated in the BiGRU+CRF model, minimal performance discrepancies were observed. However, the third version model, BiGRU+EA+CRF, achieves the highest precision across both macro and micro criteria. This suggests the benefit of attention mechanisms for precisely identifying event types. The third attention mechanism integrates both shallow and deep layer information, reduces interference in shallow layers, and more effectively adjusts information in deep layers compared with the other two attention mechanisms.

The results from utilizing exponential class weights in BiGRU+A+CRF+W and BiGRU+EA+CRF+W highlight an enhancement in recall by 5-7% and F1-score by 3-4%, without losing precision by macro criteria. This emphasizes the significant improvement of exponential class weights in event detection tasks with imbalanced datasets.

Based on the results of the nine models presented in Table III, the BiGRU+EA+CRF+W model yields the best overall performance. In spite of the moderate performance on the micro standard, it achieved the highest recall and F1 score and ranked second on precision on the macro standard. Results obtained in Table III were also compared with the published results of four models [18], namely DMCNN, BERT, BERT+CRF, and DMBERT (see Table IV). Based on the results of the four models published by other researchers

TABLE III. The Performance of Nine Models on the Testing Dataset

No.	Model	Micro			Macro		
		Precision	Recall	F1-score	Precision	Recall	F1-score
1	BiLSTM	81.73	86.17	83.90	77.20	76.73	76.96
2	BiGRU	81.98	**86.29**	84.08	78.47	75.95	76.13
3	BiLSTM+CRF	85.29	83.44	**84.36**	79.02	73.50	75.25
4	BiGRU+CRF	84.35	84.01	84.18	79.86	73.98	75.80
5	A+BiGRU+CRF	85.47	80.64	82.99	79.44	70.20	73.51
6	BiGRU+A+CRF	85.17	82.23	83.67	80.21	72.06	74.62
7	BiGRU+EA+CRF	**86.41**	79.66	82.90	**81.61**	70.67	74.61
8	BiGRU+A+CRF+W	82.45	84.43	83.43	80.66	76.36	77.21
9	BiGRU+EA+CRF+W	83.58	83.51	83.55	81.29	**77.07**	**78.07**

TABLE IV. The Performance of Other Four Models [18]

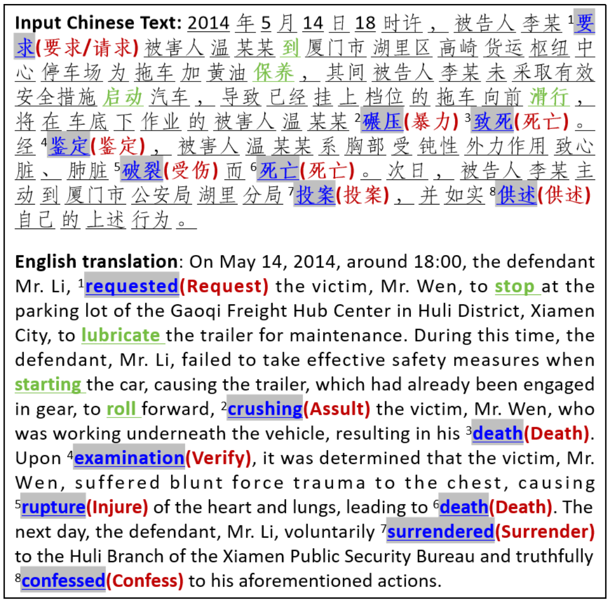
No.	Model	Micro			Macro		
		Precision	Recall	F1-score	Precision	Recall	F1-score
1	DMCNN	**85.88**	79.70	82.67	80.55	73.31	75.03
2	BERT	84.19	84.31	84.25	79.61	76.76	77.33
3	BERT+CRF	82.82	84.56	84.19	79.77	77.65	77.83
4	DMBERT	83.77	**86.22**	**85.48**	**81.57**	**80.90**	**80.34**

(as presented in Table IV), the large BERT-based model DMBERT demonstrates remarkable performance. However, compared to the best RNN-based model in this study which is BiGRU+EA+CRF+W, DMBERT performs very closely by the macro precision and shows only a slight 2%-3% improvement by the micro recall and F1 score. On the other hand, BERT-based models have much larger parameters than the RNN-based models. Consequently, the BiGRU+EA+CRF+W model can train and infer faster for similar performance.

C. Discussion

Fig. 8 illustrates the capability of the proposed model, BiGRU+EA+CRF+W, by showcasing one of the output samples in which legal event types were extracted from an input Chinese text. In addition to showcasing the result of the Chinese sample, the corresponding English translation is provided to aid reader comprehension.

The Chinese text in the sample contains about 100 tokens. The blue tokens indicate legal event triggers, the red tokens indicate corresponding event types, and the green tokens indicate non-legal event triggers that are not detected by the model. The BiGRU+EA+CRF+W model successfully detects all 8 legal events. Non-legal events such as “stop”, “lubricate”, “starting”, and “roll”, although representing events, are not relevant to legal matters; thus, the model does not extract them. This example demonstrates the model’s exceptional performance in accurately identifying several legal occurrences within a lengthy Chinese text. It is worth noting that the proposed RNN-based models are currently unable to handle English event detection.



Input Chinese Text: 2014 年 5 月 14 日 18 时许 ， 被告人 李某 [1]要求(要求/请求) 被害人 温 某某 到 厦门市 湖里区 高崎 货运 枢纽 中心 停车场 为 拖车 加 黄油 保养 ， 其间 被告人 李某 未 采取有效 安全措施 启动 汽车 ， 导致 已经 挂 上 档位 的 拖车 向前 滑行 ， 将 在 车底 下 作业 的 被害人 温 某某 [2]碾压(暴力) [3]致死(死亡) 。 经 [4]鉴定(鉴定) ， 被害人 温 某某 系 胸部 受 钝性 外力作用 致心脏 、 肺脏 [5]破裂(受伤) 而 [6]死亡(死亡) 。 次日 ， 被告人 李某 主动 到 厦门市 公安局 湖里 分局 [7]投案(投案) ， 并 如实 [8]供述(供述) 自己 的 上述 行为 。

English translation: On May 14, 2014, around 18:00, the defendant Mr. Li, [1]**requested**(**Request**) the victim, Mr. Wen, to **stop** at the parking lot of the Gaoqi Freight Hub Center in Huli District, Xiamen City, to **lubricate** the trailer for maintenance. During this time, the defendant, Mr. Li, failed to take effective safety measures when **starting** the car, causing the trailer, which had already been engaged in gear, to **roll** forward, [2]**crushing**(**Assult**) the victim, Mr. Wen, who was working underneath the vehicle, resulting in his [3]**death**(**Death**). Upon [4]**examination**(**Verify**), it was determined that the victim, Mr. Wen, suffered blunt force trauma to the chest, causing [5]**rupture**(**Injure**) of the heart and lungs, leading to [6]**death**(**Death**). The next day, the defendant, Mr. Li, voluntarily [7]**surrendered**(**Surrender**) to the Huli Branch of the Xiamen Public Security Bureau and truthfully [8]**confessed**(**Confess**) to his aforementioned actions.

Fig. 8. Event detection result of model BiGRU+EA+CRF+W on the sample. Blue tokens indicate legal event triggers. Red tokens indicate corresponding event types. Green tokens indicate non-legal event triggers that are not detected by the model.

V. Conclusion

This study presents research on Chinese event detection. Most event detection research focuses on English corpora, with a limited exploration of Chinese corpora. This study aims to enhance the performance of the sequential labeling model BiLSTM for Chinese event detection. Four modifications, including replacing the base model with the BiGRU, incorporating the attention mechanism, replacing softmax with CRF, and utilising exponential class weights, were introduced. Experiments were conducted using the largest Chinese legal text event detection dataset, LEVEN. The experimental results indicate a significant improvement over the basic model. The proposed model performs nearly comparably to the state-of-the-art models based on BERT, yet with fewer parameters.

However, this study has several limitations. Firstly, experiments were conducted solely on the Chinese dataset LEVEN, which is limited to legal events. Future research could explore other Chinese event detection datasets and even consider creating datasets due to scarcity. Secondly, the exploration of attention mechanisms was limited to multi-head attention, which exhibits modest effectiveness. Future work could explore alternative attention mechanisms and modifications of them. Lastly, further investigation into different methods for data imbalance in event detection is needed, though the exponential class weights approach effectively alleviates the issue.

Acknowledgment

This publication is supported by the Xiamen University Malaysia Research Fund (XMUMRF/2022-C9/IECE/0034).

References

[1] H. Chen, K. Liu, P. Liu, J. Wang, and N. Ge, "Robustness analysis and evaluation study of chinese text event detection models," in *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*, 2023, pp. 632–639.

[2] Q. Li, J. Li, J. Sheng, S. Cui, J. Wu, Y. Hei, H. Peng, S. Guo, L. Wang, A. Beheshti, and P. S. Yu, "A survey on deep learning event extraction: Approaches and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 5, pp. 6301–6321, 2024.

[3] S. Lou, X. Xie, W. Liu, and W. Jiang, "Enhancing chinese event prediction with prompt-driven knowledge augmentation," *Applied Sciences*, vol. 15, no. 23, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/23/12543

[4] F. Yao, C. Xiao, X. Wang, Z. Liu, L. Hou, C. Tu, J. Li, Y. Liu, W. Shen, and M. Sun, "LEVEN: A large-scale Chinese legal event detection dataset," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 183–201. [Online]. Available: https://aclanthology.org/2022.findings-acl.17

[5] Z. He, Z. Wang, W. Wei, S. Feng, X. Mao, and S. Jiang, "A survey on recent advances in sequence labeling from deep learning models," *arXiv preprint arXiv:2011.06727*, 2020.

[6] Y. Zhou, Z. Qu, Y. Liang, Y. Zhang, J. Zhang, and L. Ni, "Towards a word-granularity paradigm for chinese event detection: Targeting long-tail challenges in syntax and semantics," *Alexandria Engineering Journal*, vol. 134, pp. 135–151, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016825011718

[7] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2018, pp. 112–117.

[8] S. Henning, W. Beluch, A. Fraser, and A. Friedrich, "A survey of methods for addressing class imbalance in deep-learning based natural language processing," *arXiv preprint arXiv:2210.04675*, 2022.

[9] T. Xu, K. De Barbaro, D. H. Abney, and R. F. A. Cox, "Finding structure in time: Visualizing and analyzing behavioral time series," *Frontiers in Psychology*, vol. 11, p. 1457, 2020.

[10] C. Birnie and F. Hansteen, "Bidirectional recurrent neural networks for seismic event detection," *Geophysics*, vol. 87, no. 3, p. KS97–KS111, 2022. [Online]. Available: http://dx.doi.org/10.1190/geo2020-0806.1

[11] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, p. 5929–5955, 2020. [Online]. Available: http://dx.doi.org/10.1007/s10462-020-09838-1

[12] I. D. Mienye and T. G. Swart, "A comprehensive review of deep learning: Architectures, recent advances, and applications," *Information*, vol. 15, no. 12, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/12/755

[13] D. Soydaner, "Attention mechanism in neural networks: where it comes and where it goes," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13 371–13 385, 2022.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] W. A. Shakir, "Enhancing named entity recognition through neural architectures," in *2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2024, pp. 1–5.

[16] J. Lafferty, A. McCallum, F. Pereira *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Icml*, vol. 1, no. 2. Williamstown, MA, 2001, p. 3.

[17] fxsjy, "Jieba: Chinese text segmentation library," 2023. [Online]. Available: https://github.com/fxsjy/jieba

[18] F. Yao, C. Xiao, X. Wang, Z. Liu, L. Hou, C. Tu, J. Li, Y. Liu, W. Shen, and M. Sun, "Leven: A large-scale chinese legal event detection dataset," *arXiv preprint arXiv:2203.08556*, 2022.

[19] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical reasoning on chinese morphological and semantic relations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2018, pp. 138–143. [Online]. Available: http://aclweb.org/anthology/P18-2023

[20] X. Shen, X. Tian, T. Liu, F. Xu, and D. Tao, "Continuous dropout," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 3926–3937, 2018.

[21] Y. Hong and J. Lin, "On convergence of adam for stochastic optimization under relaxed assumptions," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 10 827–10 877. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/14bb27f680bee45d83bc769738e7f9b5-Paper-Conference.pdf