applied sciences

MDPI

Article

Generating Chinese Event Extraction Method Based on ChatGPT and Prompt Learning

Jianxun Chen, Peng Chen * and Xuxu Wu

School of Information Network Security, People's Public Security University of China, Beijing 100038, China; 2021211440@stu.ppsuc.edu.cn (J.C.)

* Correspondence: chenpeng@ppsuc.edu.cn

Abstract: Regarding the scarcity of annotated data for existing event extraction tasks and the insufficient semantic mining of event extraction models in the Chinese domain, this paper proposes a generative joint event extraction model to improve existing models in two aspects. Firstly, it utilizes the content generation capability of ChatGPT to generate annotated data corpora for event extraction tasks and trains the model using supervised learning methods adapted to downstream tasks. Secondly, explicit entity markers and event knowledge are added to the text to construct generative input templates, enhancing the performance of event extraction. To validate the performance of this model, experiments are conducted on DuEE1.0 and Title2Event public datasets, and the results show that both data enhancement and prompt learning based on ChatGPT effectively improve the performance of the event extraction model, and the F1 values of the events extracted by the CPEE model proposed in this paper reach 85.1% and 59.9% on the two datasets, respectively, which are comparable to the existing models' values of 1.3% and 10%, respectively; moreover, on the Title2Event dataset, the performance of different models on the event extraction task can be gradually improved as the data size of the annotated corpus of event extraction generated using ChatGPT increases.

Keywords: ChatGPT; prompt learning; event extraction

check for updates

Citation: Chen, J.; Chen, P.; Wu, X. Generating Chinese Event Extraction Method Based on ChatGPT and Prompt Learning. *Appl. Sci.* **2023**, *13*, 9500. https://doi.org/10.3390/app13179500

Academic Editor: Dimitris Mourtzis

Received: 12 July 2023
Revised: 14 August 2023
Accepted: 19 August 2023
Published: 22 August 2023

1. Introduction

With the increasing popularity of the Internet, the amount of information on the web is growing exponentially. Extracting valuable information from massive data has become an essential and challenging task. Information extraction (IE) has emerged to address this issue, aiming to extract unstructured information into structured data formats. It includes various techniques such as Named Entity Recognition (NER), Relation Extraction, and Event Extraction (EE) [1]. As an essential branch of information extraction, event extraction focuses on extracting structured event information from unstructured text. The Automatic Content Extraction (ACE) conference [2] divides event extraction into two sub-tasks: event detection and argument extraction. Event detection refers to detecting trigger words that represent the occurrence and type of events, while argument extraction involves extracting event arguments, such as around the relevant characters, time, place and other entities of the trigger words. In the English domain, rich datasets are available for event extraction, including a series of English text samples covering various events in different fields and types, such as ACE 2005 [3] and CASIE [4]. Furthermore, the grammatical structures and regularity of English are much clearer and more standardized, usually following a concise subject–verb–object structure.

However, in the Chinese domain, event extraction tasks still need to be solved due to the scarcity of event data and the complexity of the text. Traditional supervised learning methods rely on a large amount of manually annotated data to train models. However, existing annotated corpora often need to be more prominent in scale, domain-specific, and expensive to annotate, leading to a poor generalization of trained models. Additionally, as event extraction involves multiple sub-tasks, different sub-tasks require the design of different prompt learning templates and information-sharing relationships between templates, which suffers from inefficient manual design and propagation of parameter errors, posing difficulties for event extraction in the Chinese domain.

To address these challenges, this paper defines the event extraction task as a generative one. It proposes improvements to the BART model [5] from data augmentation and informative learning templates. On the one hand, the content generation capability of ChatGPT (AI-Generated Content, AIGC) is applied to the corpus generation in the field of event extraction, addressing the issues of data scarcity and poor generalization in existing tasks. On the other hand, the rich semantic knowledge in the model is mined through the construction of generative input templates, in order to reduce the error propagation and to solve the inefficiency problem of manually designing templates.

2. Related Work

There are three main categories of methods for event extraction: pattern-based methods, traditional machine learning methods, and deep learning methods. Pattern-based methods [6–10] are suitable for specific text formats in a single domain but heavily rely on manually crafted rules. Traditional machine learning methods [11–14] have good portability and work effectively on simple texts but suffer from insufficient feature mining. Deep learning methods are the current mainstream paradigm for event extraction tasks, which focus on automatically mining data features through neural networks and accomplishing tasks through feature learning.

Within the realm of deep learning methods, the initial approach was the pipeline model [15–22], which divided event extraction into two sub-tasks: trigger word extraction and event argument extraction. Chen et al. [15] proposed the Dynamic Multi-Pooling Convolutional Neural Network (DMCNN), which dynamically retained lexical features using convolutional neural networks and performed multi-classification tasks for the information extraction required in both stages. Zeng [16] combined Bi-LSTM and CRF to achieve good results by defining event extraction as a sequence labelling task. Li [18] proposed Multi-Turn Question Answering (MQAEE), modelling the task as a series of machine reading comprehension models based on question–answer templates to address the problem of traditional classification-based extraction models being unable to generalize to new event types and argument roles. In general, the pipeline model is simple and intuitive, with good scalability and interpretability. However, it needs help with issues such as error accumulation and the loss of contextual information.

To address these drawbacks, some researchers had proposed joint model methods [13,23–29], which make full use of the interaction information of trigger words and event arguments and extract them simultaneously on this basis. Li et al. [13] implemented the joint learning of trigger words and event argument extraction tasks based on traditional feature extraction methods and obtained good results with a structured perceptron model. Nguyen [23] constructed local features for text sequences and local windows, and global features for between-event trigger words, between-event arguments, and between trigger words and event arguments based on bidirectional recurrent neural networks through deep and joint learning, thus improving the performance of the event extraction task. Lu et al. [27] defined information extraction as a generative model from sequences to structures, and proposed the Text2Event model as an example for the event extraction task, which is able to learn a parallel corpus containing knowledge structures directly, thus modeling all sub-tasks of event extraction uniformly. Compared to the pipeline model, the joint model methods can effectively utilize contextual information and model the interactions among multiple sub-tasks to correct errors in individual sub-tasks. However, the models above only improved the model from the perspectives of model structure and problem definition, overlooking the prior knowledge learned by pre-trained language models.

In recent years, prompt learning techniques have emerged as a new paradigm, combining the original input text with prompt templates to allow the model to output answers based on prompt information. Prompt templates help the model learn the prior knowledge of pre-trained language models. Prompt learning has shown good performance in basic tasks such as text classification and sentiment recognition; however, this technique still lacks an effective application in the event extraction task. Additionally, acquiring high-quality annotated event corpora still poses challenges due to high costs and the lengthy annotation time. For example, as one of the most widely used event extraction datasets, the ACE 2005 database requires linguistic experts for two rounds of annotation. The complexity of event structures and the diversity of semantic expressions limit the number of event samples and event categories covered in existing annotated corpora, thus affecting the recognition and generalization performance of trained models. Therefore, in view of the deficiencies in the current event task, this paper intends to work from the perspectives of data augmentation and semantic mining to investigate the enhancement methods for the event extraction task.

3. Model

3.1. Generative Event Extraction Framework

Currently, pre-trained language frameworks have been widely applied in natural language processing. Some major pre-trained language frameworks include BERT (Bidirectional Encoder Representations from Transformers) [30], GPT (Generative Pre-Trained Transformer) [31], and BART (Bidirectional and Auto-Regressive Transformers) [5], among others. Among them, BART combines autoregressive and denoising autoencoder mechanisms, enabling the model to learn text generation and comprehension simultaneously. As a result, it has reliable generation capabilities and finds extensive applications in generative tasks. By defining the event extraction task as a sequence-to-structure text generation task, we can extract trigger words and event arguments from the text in an end-to-end manner, simplifying the complex structure prediction task into a neural network model and linearizing all trigger words and event arguments as structural expressions. For this purpose, this paper uses the BART model as the base model for the pre-trained language.

The BART model outputs linearized structural expressions $Y = y_1, \ldots, y_m$ for a given input token sequence $X = x_1, \ldots, x_n$. The principle of its working process follows.

First, BART converts the token sequence into embedding vectors through an embedding layer:

$$X_{emb} = \text{Embedding}(x_1, \ldots, x_n) \tag{1}$$

Next, the input's hidden vector representations $H = h_1, \ldots, h_n$ are computed using multiple layers of transformer encoders, where h_i represents the context-aware representation at the i-th position:

$$H = \text{Encoder}(X_{\text{emb}}) \tag{2}$$

In the above equations, Encoder$(\cdot)$ represents the computation process of an encoder layer, where each encoder layer consists of a self-attention mechanism and feed-forward neural network layer.

After the input token sequence is encoded, the decoder sequentially utilizes the hidden vectors of the input tokens to predict the output structure one by one. At the i-th step of generating text, the self-attention decoder predicts the i-th linearized token y_i and the decoder state h_i^d as follows:

$$y_i, h_i^d = \text{Decoder}\left([H; h_1^d, \ldots, h_{i-1}^d], y_{i-1}\right) \tag{3}$$

Here, Decoder$(\cdot)$ represents the computation process of a decoder layer, where each decoder layer includes a self-attention mechanism, encoder–decoder attention mechanism, and a feed-forward neural network layer.

The conditional probability of the entire output sequence $p(y|x)$ is obtained by multiplying the probabilities $p(y_i|y_{<i}, x)$ at each step:

$$p(y|x) = \prod_{i}^{|y|} p(y_i|y_{<i}, x) \tag{4}$$

Here, $y_{<i} = y_1, \ldots, y_{i-1}$ and $p(y_i|y_{<i}, x)$ is the normalized probability distribution obtained by applying softmax$(\cdot)$ overall target vocabulary.

On the BART pre-trained language model, this paper proposes a generative event extraction model based on ChatGPT and prompt learning, called ChatGPT–Prompt-Event Extractor (CPEE), which incorporates improvements from both data augmentation and input construction perspectives. The specific enhancements are shown in Figure 1.

1. Based on known event knowledge, through the interaction with ChatGPT under the rule template, the event extraction annotation corpus is used to pre-train the model through supervised learning, adapting it to downstream tasks (pre-finetuning).
2. Combining prompt learning and construct input templates for generative models, in these templates they explicitly incorporate entity markers and event knowledge to fully leverage the pre-trained language model's prior knowledge.

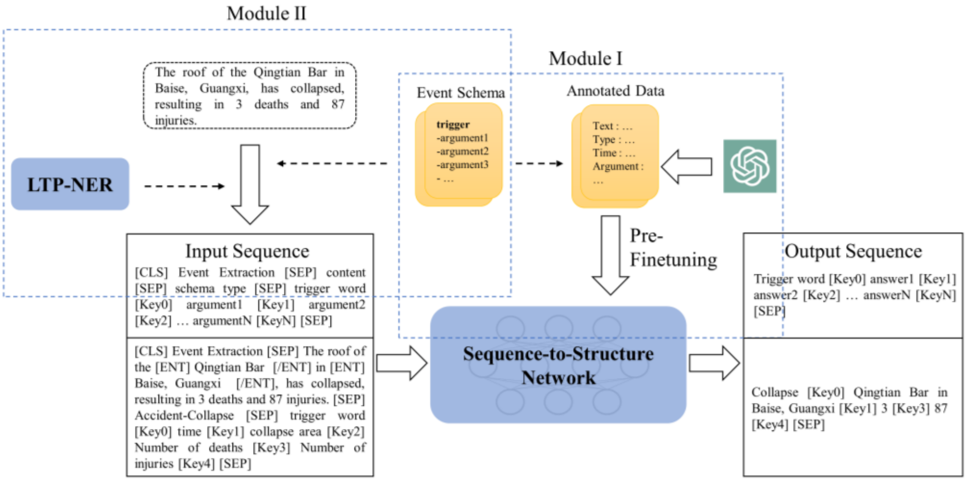

Module II
The roof of the Qingtian Bar in Baise, Guangxi, has collapsed, resulting in 3 deaths and 87 injuries.
LTP-NER
Module I
Event Schema
trigger
-argument1
-argument2
-argument3
- …
Annotated Data
Text : …
Type : …
Time : …
Argument : …
Pre-Finetuning
Input Sequence
[CLS] Event Extraction [SEP] content [SEP] schema type [SEP] trigger word [Key0] argument1 [Key1] argument2 [Key2] … argumentN [KeyN] [SEP]
[CLS] Event Extraction [SEP] The roof of the [ENT] Qingtian Bar [/ENT] in [ENT] Baise, Guangxi [/ENT], has collapsed, resulting in 3 deaths and 87 injuries. [SEP] Accident-Collapse [SEP] trigger word [Key0] time [Key1] collapse area [Key2] Number of deaths [Key3] Number of injuries [Key4] [SEP]
Sequence-to-Structure Network
Output Sequence
Trigger word [Key0] answer1 [Key1] answer2 [Key2] … answerN [KeyN] [SEP]
Collapse [Key0] Qingtian Bar in Baise, Guangxi [Key1] 3 [Key3] 87 [Key4] [SEP]

Figure 1. Generative event extraction model graph based on ChatGPT and prompt learning.

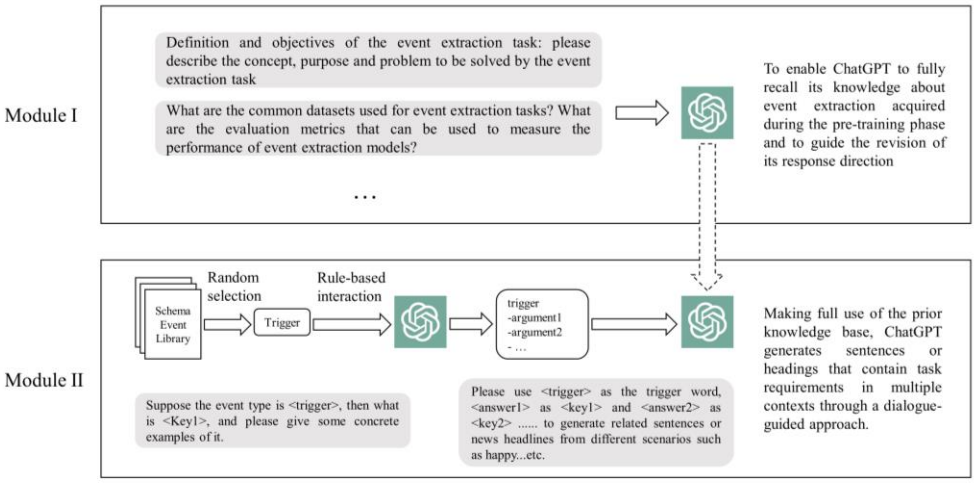
3.2. Annotation Data Generation Based on ChatGPT

To address the issue of scarce annotated data in event extraction, this paper utilizes the content generation capability of ChatGPT to generate an event extraction annotated corpus. This process involves two modules: recall and generation, as shown in Figure 2.

1. Recall module: A continuous dialogue is established to allow ChatGPT to recall the event extraction knowledge it acquired during training. During this phase, questions related to event extraction and requests for data generation are posed to ChatGPT while continuously guiding and correcting its answers. Through extensive training on a large amount of textual data, ChatGPT learns the inherent logic of language and improves its generative ability by predicting the next word or sentence, accumulating a significant amount of semantic knowledge. In the first phase, an "activation question library" related to event extraction is created after multiple attempts. By continuously asking ChatGPT questions from the "activation question library", the model is guided to recall the knowledge it acquired during training regarding event extraction. The "activation question library" includes the basic definition of event extraction tasks, particularly the meaning of trigger words and event arguments, the construction of commonly used datasets for event extraction, and relevant examples. Correct answers are encouraged and supported, while incorrect answers are corrected and guided,

thus activating ChatGPT's relevant knowledge on event extraction and establishing sufficient contextual connections.

2. Generation module: To enable ChatGPT to generate sentences that meet the task requirements in various contexts based on given trigger words and event arguments, the process begins with the random extraction of event trigger words. A trigger word is randomly selected from a schema event library as the trigger word for the current task. Then, a question-and-answer filling mechanism based on rules is used to ask ChatGPT about the corresponding event arguments for the newly initiated conversation based on the trigger word. The randomly extracted trigger word and its related event arguments are combined to form the task requirements for the current iteration. The task requirements and different contextual prompts are then inputted into ChatGPT. Through leveraging prior knowledge and guided by the dialogue, ChatGPT is directed to generate sentences in various contexts that fulfill the task requirements. Finally, the generated corpus is manually reviewed and edited to ensure quality and accuracy. Human intervention allows for better control over the content and logic of the generated corpus while eliminating potential errors or inappropriate text.



Definition and objectives of the event extraction task: please describe the concept, purpose and problem to be solved by the event extraction task

What are the common datasets used for event extraction tasks? What are the evaluation metrics that can be used to measure the performance of event extraction models?

To enable ChatGPT to fully recall its knowledge about event extraction acquired during the pre-training phase and to guide the revision of its response direction

Suppose the event type is <trigger>, then what is <Key1>, and please give some concrete examples of it.

Please use <trigger> as the trigger word, <answer1> as <key1> and <answer2> as <key2> ...... to generate related sentences or news headlines from different scenarios such as happy...etc.

Making full use of the prior knowledge base, ChatGPT generates sentences or headings that contain task requirements in multiple contexts through a dialogue-guided approach.

Figure 2. ChatGPT generates a flowchart of event extraction annotation data.

Before training the pre-trained language model on task-specific data, it is beneficial to pre-train the model in the relevant domain through supervised training [32]. This effectively improves the model's performance in that domain. Therefore, in the case of the generative model BART, which has been trained on a large-scale general corpus, task pre-finetuning is performed. The event extraction annotated corpus generated by ChatGPT is used to fine-tune the model's parameters in the information extraction domain. This allows the model to acquire domain-specific knowledge in event extraction, correcting the deviation between general tasks and event extraction tasks.

3.3. Generative Input Template Construction

An input template is explicitly constructed for text generation tasks to fully leverage the rich semantic knowledge within the model entirely, explicitly incorporating entity markers and event knowledge. The specific process is as follows:

1. Entity prompts are introducing to encode entities in the text explicitly. Namely, the Named Entity Recognition (NER) technique from the LTP (Language Technology Platform) is employed to identify entities in the input text. Afterwards, semantic entity markers are explicitly added on both sides of the entities, incorporating entity information into the input sequence. Taking Figure 3 as an example, for the sentence "NIO has laid off 62 employees in the United States, affecting multiple departments."

the entity "NIO" is recognized. While constructing the input template, an entity semantic marker is added as "[ENT] NIO [/ENT]".

2. The event schema is introduced as event information and added to the input sequence. Specifically, at the beginning of the text, "Event Extraction" is added as a task prompt to indicate the current task mode to the model. Then, the event schema, including the trigger words and event arguments, is enumerated and concatenated to the end of the text, constraining the generated label content.
3. Negative samples are randomly generated in two ways during training. Firstly, incorrect event schemas are intentionally added as event information to the input sequence, making it impossible for the text to correspond effectively to the event schema. For example, they are combining the topic "Lay off" with the event type "Strike". Secondly, incorrect event arguments are added to the correct event schema and included as event information in the input sequence, causing the model to fail to find the correct label corresponding to the incorrect event argument and, for instance, adding the wrong event argument "Suspension period." to the event information for "Lay off". By introducing negative samples into the training data, the model can learn to differentiate between correct and incorrect answers and become more inclined to generate the copyrighters during the generation process.

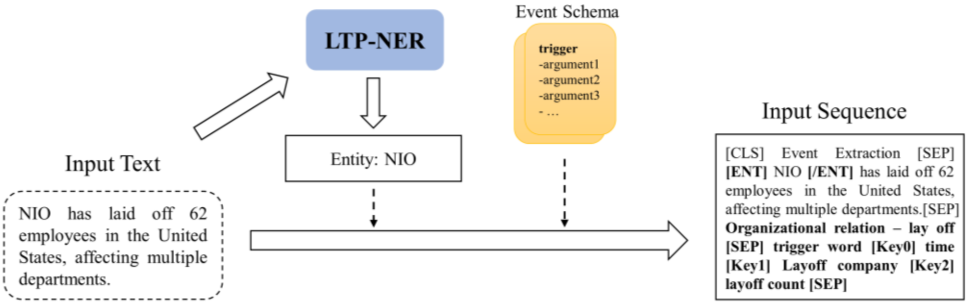

Event Schema
LTP-NER
trigger
-argument1
-argument2
-argument3
- …
Input Sequence
Input Text
Entity: NIO
NIO has laid off 62 employees in the United States, affecting multiple departments.
[CLS] Event Extraction [SEP] [ENT] NIO [/ENT] has laid off 62 employees in the United States, affecting multiple departments.[SEP] Organizational relation – lay off [SEP] trigger word [Key0] time [Key1] Layoff company [Key2] layoff count [SEP]

Figure 3. Generative input template construction diagram.

4. Experimental Analysis

4.1. Experimental Datasets

The article conducts experiments on the publicly available datasets DuEE1.0 and Title2Event. As shown in Table 1, DuEE1.0 [33] is a Chinese event extraction dataset released by Baidu, consisting of 17,000 sentences with event information across 65 event types. The event types were selected based on the hot topics from Baidu's Top Searches, ensuring strong representativeness. These 65 event types include not only common types such as "marriage", "resignation", and "earthquake", typically found in traditional event extraction evaluations, but also current event types such as "likes" that reflect the current era. Title2Event [34], on the other hand, is a Chinese title dataset released by Deng, designed for open event extraction tasks. This dataset comprises 42,915 news titles collected from Chinese web pages across 34 domains. It includes 70,947 manually annotated event triplets and 24,231 unique predicates. The dataset has undergone two rounds of expert reviews, making it the largest manually annotated Chinese dataset for sentence-level event extraction currently available. The data presentation of two publicly available Chinese event extraction datasets is shown in Table 2.

Table 1. Dataset information.

Datasets	Number of Training Sets	Number of Validation Sets	Number of Test Sets	Number of Events
DuEE1.0	13,478	1790	4372	19,640
Title2Event	34,295	4286	4288	70,879

Table 2. Data presentation of two publicly available Chinese event extraction datasets.

Datasets	Trigger Word	Arguments	Sentence	Event Extraction
DuEE1.0	layoffs	Time, Layoff Side, Number of Layoffs	900 laid-off employees officially bid farewell to Oracle, where did those who left the foreign companies go?	{Layoff side: Oracle} {Number of Layoffs: 900}
	Defeat	Time, loser, winner, name of the match	Atletico Madrid defeated Juve 2-0 at home, back to Juve, Crowe hat trick tease Atletico Madrid	{Winner: Atletico Madrid} {Loser: Juve}
	. . .	. . .	. . .	. . .
Title2Event	Carry out	Subject, Object	Market supply is sufficient, prices are basically stable, consumption order is normal, Qingdao carries out special law enforcement action on price order.	{Subject: Qingdao} {Object: special law enforcement action on price order}
	on the list	Subject, Object	Two cities in Zhejiang province on the list of China's happiest provincial capitals and planned cities in 2021.	{Subject: Two cities in Zhejiang Province} {Object: China's happiest provincial capitals and planned cities in 2021}
	. . .	. . .	. . .	. . .

Firstly, from the two datasets of DuEE1.0 and Title2Event, we obtain the event schema, including trigger words and corresponding event arguments, and then randomly extract the trigger words in the event schema, we can obtain the event extraction annotation corpus by generating a given number of training corpora through ChatGPT. The text contained in the event corpus can be categorized into six themes, including six categories such as social, financial, sports, current affairs, science and technology, and others, and each theme contains more than 500 instances. Among them, the proportion of other categories is more than 50%. Table 3 below provides some examples of the corpus.

Table 3. Data presentation of ChatGPT event extraction annotation corpus.

Item No.	Trigger Word	Arguments	Sentence
1	Support	Participating Subjects, Participating Objects, Location	Donors generously support hospitals to expand beds and medical resources to better care for the health needs of vulnerable groups.
2	Support	Participating Subjects, Participating Objects, Location	The professional support of the maintenance staff guarantees the safety and operational efficiency of the Convention Center's facilities and facilitates a wide variety of meetings and events for community development.
3	listing	Listing Body, Listing Date, Location	By the end of 2022, Huawei plans to list on the London Stock Exchange to strengthen its global capital markets participation and transparency.
. . .	. . .	. . .	. . .

In the given instances, ChatGPT provides event prompts to guide the consistency of event arguments. For example, in sentence 1, under the event prompt "charity event", the generated event arguments "donors" and "vulnerable groups" maintain good consistency. However, ChatGPT can also make errors and deviate from the trigger words. For instance, in sentence 2, the trigger word changed from "support" to "guarantee". The reason behind this is that when dealing with complex questions, ChatGPT may encounter ambiguity in the semantic or contextual information present in the data, leading to errors or confusion in identifying trigger words. Additionally, when ChatGPT generates event arguments as task requirements by randomly rendering trigger words and template dialogues, the factual accuracy of the generated sentences about real-world events may deviate significantly. For example, in sentence 2, the listing event of Huawei and the location are randomly generated event arguments dependent on template dialogues. The generated sentence can only maintain semantic coherence but may not align with actual real-world events.

4.2. Experimental Results

To evaluate the proposed model (CPEE) in this paper, we compared it with several existing models that have achieved good experimental results in the field of event extraction in recent years, which serve as baseline models. Precision (P), recall (R), and F1-measure were adopted as evaluation metrics. The evaluation method comprehensively assessed trigger words and event arguments collected for each model. Specifically, we compared our model with baseline models using sequence labelling and machine reading comprehension-based approaches. The running experimental environment was Linux, the GPU model was Tesla V100, the memory was 32 G, and the deep learning framework was Pytorch1.12.1 [35]. The experimental parameters were Epoch = 20, Batch size = 16, Max length = 1024, dropout = 0.5, and lr = 0.01.

Using the data enhancement method proposed in this paper, 4000 pieces of data were generated based on 3000 randomly selected event trigger words. Subsequently, the CPEE model proposed in this paper was run and compared with other models in the public dataset DuEE1.0 with Title2Event for experiments, and the results are shown in Table 4.

Table 4. Performance comparison of different models on public datasets.

Dataset	Model	P	R	F1
DuEE1.0	ERNIE + CRF [36]	-	-	75.9
	Bert4keras [30]	85.5	73.7	79.2
	MRC [16]	82.6	85.0	83.8
	CPEE	87.6	82.8	85.1
Title2Event	BERT-CRF [37]	41.1	41.3	41.2
	MRC [16]	44.5	44.8	44.7
	Seq2seqMRC [38]	49.8	50.1	49.9
	CPEE	62.3	57.8	59.9

Through the analysis of the comparative experimental results, it can be observed that the CPEE model outperforms mainstream event extraction models in terms of F1-score on both Chinese datasets. Specifically, on the DuEE1.0 dataset, the CPEE model achieves a 9.2% improvement in F1-score compared with the baseline model ERNIE + CRF and a 1.3% improvement compared to the MRC model. Although its recall rate is lower than that of the MRC model, the CPEE model performs better by achieving higher precision, indicating its ability to extract accurate event information. On the Title2Event dataset, compared with the other three models, the CPEE model exhibits significant advantages, achieving good results in precision, recall, and F1-score. It shows an 18.7% improvement in the F1-score compared with the sequence labelling model BERT-CRF and a 10% improvement compared with the Seq2seqMRC model. This indicates that CPEE is better at comprehensively capturing accurate and actual trigger words and event arguments.

To further validate the effectiveness of different models, we conducted ablation experiments on the Title2Event dataset, and the results are shown in Table 5. The introduction of the ChatGPT-annotated data corpus for pre-finetuning improves the F1-score for the extraction task by 2.2%, while the introduction of semantic annotation results in an F1-score improvement of more than 1.7%. The results indicate that pre-finetuning with the ChatGPT-annotated data corpus is more capable of improving the model's adaptability to the event extraction tasks. Semantic annotation, on the other hand, effectively transfers the knowledge of the pre-trained language model by encoding event labels as natural language vocabulary. These contributions are of great significance in improving the performance of event extraction tasks.

Table 5. Comparison of ablation experimental performance.

Model	P	R	F1
Full model	62.3	57.8	59.9
Pre-finetuning	59.0	56.6	57.7
Prompt template	60.8	55.9	58.2

In order to further validate the effectiveness of the ChatGPT-annotated corpus, the experimental data were expanded to different degrees, and different models were compared. Figure 4 shows the experimental results of different event extraction models on the ChatGPT-annotated corpus, using the F1 value as the evaluation index of the experiment and the Title2Event dataset as the public dataset. The results show that the performance of different models on the event extraction task gradually improves as the data size of the event extraction annotation corpus generated using ChatGPT increases.

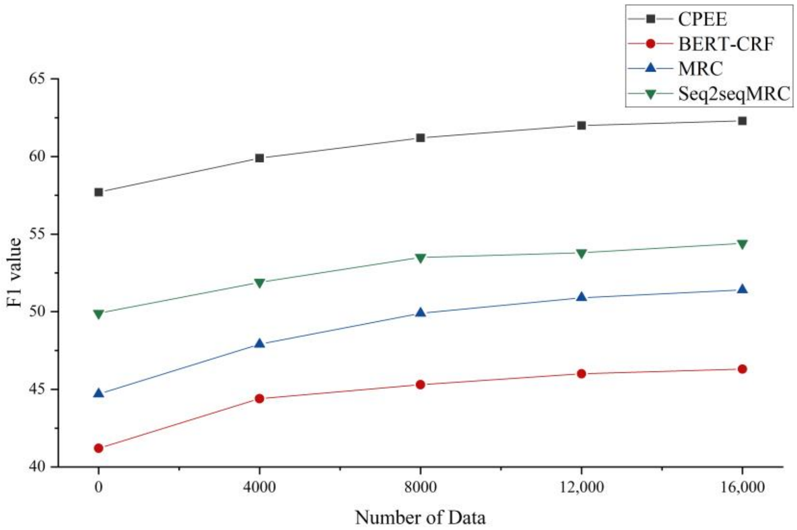


Figure 4. ChatGPT-annotated corpus experimental results.

The specific values are shown in Table 6. With the increasing corpus size from 0 to 16,000, the F1 value of the model CPEE improves from 57.7% to 62.3%, which is a 4.6% improvement; the BERT-CRF improves from 41.2% to 46.1%, which is a 4.9% improvement, the MRC improves from 44.7% to 51.4%, which is a 6.7% improvement, and the Seq2seqMRC improved from 49.9% to 54.4%, an improvement of 4.5%, indicating that larger-scale training data play a positive role in improving model performance. However, the rate of improvement in F1 values of individual models on the event extraction task decreases as the data size continues to increase. For example, the model CPEE improves the F1 value by 3.5% at data sizes from 0 to 8000, yet only by 1.1% at data sizes from 8000 to 16,000. It is hypothesized that the initial annotated corpus enables the model to capture the basic patterns and rules of ChatGPT-generated data, which results in rapid performance improvement. However, as the size of the labeled corpus increases further, the new data samples have less information uniqueness compared to the existing data, leading to a gradual slowdown in the model's performance improvement for the new samples.

Table 6. ChatGPT-annotated corpus experimental results.

		0	**4000**	**8000**	**12,000**	**16,000**
Title2Event	CPEE	57.7	59.9	61.2	62.0	62.3
	BERT-CRF	41.2	44.4	45.3	46.0	46.1
	MRC	44.7	47.9	49.9	50.9	51.4
	Seq2seqMRC	49.9	51.9	53.5	53.8	54.4

5. Conclusions

Given the scarcity of annotated data for event extraction tasks and the need for more semantic exploration of event extraction models, this paper proposes a generative event extraction model CPEE based on ChatGPT and prompt learning with event extraction as the research object. The model extends and improves existing methods in two aspects: data augmentation and input construction. Firstly, to address the annotated data scarcity, this paper uses ChatGPT to generate a labelled dataset for event extraction tasks and trains the model using supervised learning methods adapted to downstream tasks. Secondly, to fully explore the rich semantic knowledge within the model, the paper proposes the construction of generative input templates and explicitly adds entity markers within the templates. The experimental results on the public datasets DuEE1.0 and Title2Event demonstrate that the proposed model achieves higher extraction accuracy than existing event extraction methods based on reading comprehension and sequence labelling. It provides a valuable solution for event extraction tasks in the Chinese domain. The application of ChatGPT in generating labelled data for event extraction tasks can alleviate the high costs of annotating data for new domains and events, thus possessing significant practical value. The ablation experiment further confirms that the ChatGPT-annotated data corpus is more helpful in improving the model's performance.

When using ChatGPT for data enhancement, the recall phase for ChatGPT in the early stage is very critical, and it is necessary to guide and correct the content of ChatGPT's answers, so that it fully understands the inputs and outputs of the task of generating an event extraction corpus; in the process of generating, it is necessary to adjust the consistency of the trigger word, event arguments, and the generated text in a timely manner, and the effect of the trained model is not ideal if it is only pursuing the rapid generation of a large number of event extraction labeled corpora without controlling the quality of labeled precursors. However, the improvements in this paper are limited to data augmentation and input construction. In the future, it would be worth considering optimizing the model's internal structure through prompt tuning to enhance the model's understanding and generation capabilities for event semantics. Additionally, the use of ChatGPT for generating annotated data is currently explored only for event extraction tasks. In the future, researchers could explore their application to other natural language processing tasks, reducing reliance on large-scale annotated data and expanding its applicability, thereby promoting the development and advancement of related tasks.

Author Contributions: Conceptualization, J.C. and P.C.; Software, X.W.; Formal analysis, P.C.; Data curation, J.C.; Writing—original draft, J.C.; Visualization, X.W.; Funding acquisition, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by 2022 Post Graduate Course Construction Project of People's Public Security University of China (2022YJSKCJS037) and Fundamental Research Funds for the Central Universities (2022JKF02018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Hogenboom, F.; Frasincar, F.; Kaymak, U.; de Jong, F.; Caron, E. A Survey of Event Extraction Methods from Text for Decision Support Systems. *Decis. Support Syst.* **2016**, *85*, 12–22. [CrossRef]
2. Garofolo, J. *Automatic Content Extraction (ACE)*; University of Pennsylvanial: Philadelphia, PA, USA, 2005. Available online: http://itl.gov/iad/mig/-tests/ace/2005 (accessed on 3 October 2021).
3. Walker, C.; Strassel, S.; Medero, J.; Maeda, K. *ACE 2005 Multilingual Training Corpus LDC2006T06*; Web Download; Linguistic Data Consortium: Philadelphia, PA, USA, 2006.
4. Satyapanich, T.; Ferraro, F.; Finin, T. CASIE: Extracting Cybersecurity Event Information from Text. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8749–8757. [CrossRef]
5. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Cedarville, OH, USA, 2020; pp. 7871–7880.
6. Riloff, E. Automatically Constructing a Dictionary for Information Extraction Tasks. In Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, DC, USA, 11 July 1993; AAAI Press: Washington, DC, USA, 1993; pp. 811–816.
7. Kim, J.-T.; Moldovan, D.I. Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction. *IEEE Trans. Knowl. Data Eng.* **1995**, *7*, 713–724. [CrossRef]
8. Riloff, E.; Shoen, J. Automatically Acquiring Conceptual Patterns without an Annotated Corpus. In Proceedings of the Third Workshop on Very Large Corpora, Boston, MA, USA, 30 June 1995.
9. Yangarber, R. Scenario Customization for Information Extraction. Ph.D. Thesis, New York University, New York, NY, USA, 2000.
10. Jiang, J. An Event IE Pattern Acquisition Method. *Comput. Eng.* **2005**, *31*, 96–98.
11. Chieu, H.L.; Ng, H.T. A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, AB, Canada, 28 July 2002; American Association for Artificial Intelligence: Washington, DC, USA, 2002; pp. 786–791.
12. Llorens, H.; Saquete, E.; Navarro-Colorado, B. TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Coling 2010 Organizing Committee, Beijing, China, 23–27 August 2010; pp. 725–733.
13. Li, Q.; Ji, H.; Huang, L. Joint Event Extraction via Structured Prediction with Global Features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Long Papers, Sofia, Bulgaria, 4–9 August 2013; ACL: Stroudsburg, PA, USA, 2013; pp. 73–82.
14. Ahn, D. The Stages of Event Extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events, Sydney, Australia, 23 July 2006; Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; pp. 1–8.
15. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 167–176.
16. Zeng, Y.; Yang, H.; Feng, Y.; Wang, Z.; Zhao, D. A Convolution BiLSTM Neural Network Model for Chinese Event Extraction. In Proceedings of the Natural Language Understanding and Intelligent Applications: 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, 2–6 December 2016; Volume 10102, pp. 275–287.
17. Nguyen, T.H.; Grishman, R. Graph Convolutional Networks with Argument-Aware Pooling for Event Detection. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, LA, USA, 2 February 2018; AAAI Press: Washington, DC, USA, 2018; pp. 5900–5907.
18. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. Event Extraction as Multi-Turn Question Answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 829–838.
19. Du, X.; Cardie, C. Event Extraction by Answering (Almost) Natural Questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 671–683.
20. Yu, Y.F.; Zhang, Y.; Zuo, H.Y.; Zhang, L.F.; Wang, T.T. Multi-turn Event Argument Extraction Based on Role Information Guidance. *Acta Sci. Nat. Univ. Pekin.* **2023**, *59*, 83–91. [CrossRef]
21. Ding, N.; Li, Z.; Liu, Z.; Zheng, H.; Lin, Z. Event Detection with Trigger-Aware Lattice Neural Network. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 347–356.
22. Lai, V.D.; Nguyen, T.N.; Nguyen, T.H. Event Detection: Gate Diversity and Syntactic Importance Scores for Graph Convolution Neural Networks. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 5405–5411.

23. Nguyen, T.H.; Cho, K.; Grishman, R. Joint Event Extraction via Recurrent Neural Networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 300–309.
24. Sha, L.; Qian, F.; Chang, B.; Sui, Z. Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [CrossRef]
25. Sheng, J.; Guo, S.; Yu, B.; Li, Q.; Hei, Y.; Wang, L.; Liu, T.; Xu, H. CasEE: A Joint Learning Framework with Cascade Decoding for Overlapping Event Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 164–174.
26. Nguyen, T.M.; Nguyen, T.H. One for All: Neural Joint Modeling of Entities and Events. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 6851–6858. [CrossRef]
27. Lu, Y.; Lin, H.; Xu, J.; Han, X.; Tang, J.; Li, A.; Sun, L.; Liao, M.; Chen, S. Text2Event: Controllable Sequence-to-Structure Generation for End-to-End Event Extraction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2795–2806.
28. Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; Wu, H. Unified Structure Generation for Universal Information Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 5755–5772.
29. Huang, K.-H.; Hsu, I.-H.; Natarajan, P.; Chang, K.-W.; Peng, N. Multilingual Generative Language Models for Zero-Shot Cross-Lingual Event Argument Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 4633–4646.
30. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186.
31. Radford, A.; Narasimhan, K. *Improving Language Understanding by Generative Pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
32. Winata, G.I.; Madotto, A.; Lin, Z.; Liu, R.; Yosinski, J.; Fung, P. Language Models Are Few-Shot Multilingual Learners. In Proceedings of the 1st Workshop on Multilingual Representation Learning; Association for Computational Linguistics, Punta Cana, Dominican Republic, 11 November 2021; pp. 1–15.
33. Li, X.; Li, F.; Pan, L.; Chen, Y.; Peng, W.; Wang, Q.; Lyu, Y.; Zhu, Y. DuEE: A Large-Scale Dataset for Chinese Event Extraction in Real-World Scenarios. In Proceedings of the Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, 14–18 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 534–545.
34. Deng, H.; Zhang, Y.; Zhang, Y.; Ying, W.; Yu, C.; Gao, J.; Wang, W.; Bai, X.; Yang, N.; Ma, J.; et al. Title2Event: Benchmarking Open Event Extraction with a Large-Scale Chinese Title Dataset. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 6511–6524.
35. Shao, Y.; Geng, Z.; Liu, Y.; Dai, J.; Yan, H.; Yang, F.; Zhe, L.; Bao, H.; Qiu, X. CPT: A Pre-Trained Unbalanced Transformer for Both Chinese Language Understanding and Generation. *arXiv* **2021**, arXiv:2109.05729.
36. Zhang, Z.Y.; Han, X.; Liu, Z.Y.; Jiang, X.; Sun, M.S.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 1441–1451.
37. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring Pre-Trained Language Models for Event Extraction and Generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5284–5294.
38. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. MT5: A Massively Multilingual Pre-Trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 483–498.

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