



OPEN

An improved GPT2-based joint event extraction method with position expansion and knowledge augmentation

Tonghui An^{1,4}, Zhenling Zhang^{1,4}, Yangli Jia^{1,4}✉, Ruchao Jia^{1,4,6}, Maria Papadopoulou^{2,3,4,5} & Christophe Roche^{2,3,4,5}

With the rapid development of the Internet and social media, massive amounts of unstructured data have emerged, making event extraction increasingly important for information retrieval. In the financial domain, challenges such as long texts, redundant content, and complex structures hinder extraction tasks. To address this, we propose PosEKE-GPT2, an improved GPT2-based model that reformulates event extraction as a text generation task. The model jointly identifies event types, triggers, and arguments using structured canonical text and a sub-task extraction strategy to reduce error propagation. An expanded positional encoding mechanism enhances event representation in long texts. Furthermore, we introduce a knowledge augmentation module that dynamically selects and integrates external knowledge via prompt mechanisms and attention-based embedding optimization. Experiments on the DuEE-Fin dataset show that PosEKE-GPT2 achieves an average F1-score of 90.61, while on the FewFC dataset it reaches an average F1-score of 88.85, both outperforming baseline models. Ablation studies verify the effectiveness of the positional encoding and knowledge augmentation modules, demonstrating the model's robustness and suitability for financial event extraction across different datasets.

Keywords Event extraction, Joint extraction, Position encoding expansion, Knowledge augmentation

In recent years, the rapid development of the Internet and social media has witnessed an exponential growth in data volume and diversity, leading to increasingly complex challenges in information acquisition and management^{1,2}. The automatic extraction of valuable information from massive unstructured texts has become a critical research focus³, where event extraction, as a key technology, plays a pivotal role in transforming unstructured data into structured knowledge.

Event extraction typically comprises two subtasks: event detection and argument extraction. Event detection identifies triggers and classifies event types through sequence element classification, while argument extraction recognizes event attributes and annotates their corresponding roles. As illustrated in Fig. 1(a), conventional approaches to this problem are categorized into pipeline approaches⁴ and joint models. Pipeline event extraction decomposes the task into sequential independent subtasks^{5,6}, where each step operates in isolation. While this approach offers modularity and ease of implementation, it suffers from error propagation, ultimately compromising overall accuracy. In contrast, joint event extraction (Fig. 1(b)) employs an end-to-end framework⁷, enabling simultaneous extraction of triggers and arguments through a unified model. By leveraging interdependencies between tasks, it effectively mitigates cascading errors inherent in pipeline methods⁸.

Event extraction in the financial domain aims to rapidly and accurately extract event information from specialized texts⁹. However, such texts are typically characterized by extended length, information redundancy, complex syntactic structures, and frequent co-occurrence of multiple events as shown in the real-case in Fig. 2, posing significant challenges to practical extraction tasks.

¹School of Computer Science, Liaocheng University, Liaocheng 252059, China. ²TALOS Research Center in AI for SSH, University of Crete, Gallos Campus, Rethymno 74100, GR, Greece. ³LISTIC Laboratory (Computer Science, Systems, Information and Knowledge Processing Laboratory), Université Savoie Mont Blanc, Le Bourget du Lac Cedex, Chambéry 73376, France. ⁴KETRC (Knowledge Engineering & Terminology Research Centre), Liaocheng University, Liaocheng 252059, China. ⁵Condillac Research Group in Knowledge Engineering, Université Savoie Mont Blanc, Le Bourget du Lac Cedex, Chambéry 73376, France. ⁶Shandong Xinneng Shipbuilding Co., Ltd, Jining 273500, China. ✉email: jiayangli@lcu.edu.cn

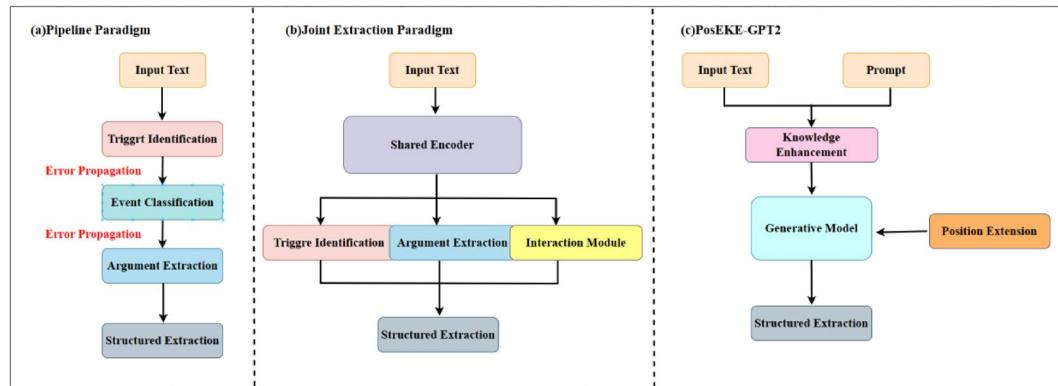


Fig. 1. Comparison of Model Architectures. **(a)** Pipeline models suffer from error propagation. **(b)** Joint extraction models require complex interaction mechanisms. **(c)** Our proposed generative paradigm (PosEKE-GPT2) simplifies the architecture through end-to-end text generation.

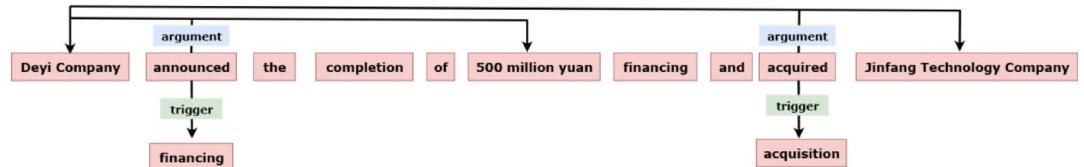


Fig. 2. Illustration of multi-event extraction from financial text.

The sentence contains two events: (1) an Enterprise Financing event (trigger: financing; arguments: Financing-party: Devi Company, Amount: 500 million yuan), and (2) an Enterprise Acquisition event (trigger: acquired; arguments: Acquirer: Devi Company, Acquiree: Jinfang Technology Company).

To address these challenges, this paper proposes PosEKE-GPT2 (Position Extension and Knowledge Enhancement on GPT2), an enhanced GPT2 model that reformulates event extraction as a text generation task. The overall architecture of our proposed method is illustrated in Fig. 1(c). Specifically, we integrate adjacent positional encodings into the original GPT2 generation framework, overcoming the limitation of fixed-length position embeddings in the base model. Additionally, we introduce an attention mechanism to further capture nonlinear relationships between input embeddings and prompt embeddings.

Experimental results demonstrate that the proposed model achieves superior performance in joint event extraction tasks, with significant improvements in precision, recall, and F1-score across event type classification, trigger identification, and argument extraction, thereby validating the model's effectiveness.

In summary, the primary contributions of this work are summarized as follows:

- 1) This work innovatively reformulates event extraction as a text generation task. Building upon the GPT2 framework, we implement a joint extraction paradigm that simultaneously identifies triggers and arguments through unified sequence generation. This architecture enables co-optimization of subtasks within a single model, effectively eliminating error propagation caused by traditional pipeline cascades.
- 2) This paper proposes a novel adjacent positional encoding fusion mechanism that doubles the input length capacity compared to conventional methods. This advancement enables precise capture of absolute positional relationships between event triggers and argument roles, thereby alleviating insufficient positional representation. The enhanced encoding significantly strengthens long-text modeling capabilities and deepens the model's semantic and structural comprehension of complex events.
- 3) This paper introduces an attention-based knowledge enhancement method that formalizes event representation via prompt engineering and integrates external knowledge embeddings to enhance contextual comprehension. This approach dynamically recalibrates knowledge relevance weights through attention mechanisms. Furthermore, during target sequence construction, a specialized token tagging strategy is introduced to explicitly delineate event types, triggers, and arguments using structural markers, thereby strengthening the model's structural awareness and boosting extraction accuracy.

Related work

Event extraction, a pivotal task in natural language processing, involves identifying critical elements from unstructured data and presenting them as structured representations for downstream applications. While researchers worldwide have extensively explored this field, particularly achieving remarkable progress through pre-trained language models (PLMs), existing methods still grapple with persistent challenges such as domain adaptation barriers and handling diverse event types across complex scenarios.

Event extraction in the financial domain

Event extraction technology holds particular significance in the financial domain, enabling critical information extraction from massive financial texts. However, the complex structural patterns and domain-specific characteristics of financial documents pose substantial challenges for event extraction. To address these issues, researchers have conducted extensive studies. Li et al.¹⁰ proposed Fin-PTPCG, a model integrating Fin-BERT with pseudo-trigger-aware pruned complete graphs. This framework effectively achieves multi-event detection and classification by combining domain-specific prior knowledge, pseudo-trigger mechanisms, and similarity pruning strategies. He et al.¹¹ developed DEEM-PT, an event extraction model based on graph neural networks (GNNs). It enhances multi-event information interaction through event-type-guided prompt templates and integrates critical arguments via pseudo-event proxy nodes. Zou et al.¹² introduced a generative financial event extraction method that resolves argument scattering and multi-event challenges through entity-to-document level information encoding and decoding. Hu et al.¹³ addressed contextual awareness and cross-sentence argument dispersion in financial documents by employing RoBERTa pre-trained embeddings combined with graph convolutional networks and enhanced path reasoning mechanisms. Jin et al.¹⁴ proposed the RACNN-BiLSTM framework, which significantly improves implicit causal relationship recognition in financial texts through fusion of local syntactic features, global semantic patterns, and self-attention mechanisms.

Despite notable progress, financial event extraction continues to face persistent challenges, particularly in lengthy document modeling and robustness enhancement. Current approaches frequently suffer from insufficient positional representation mechanisms when handling long-text scenarios, leading to degraded precision in identifying event elements. These unresolved issues demand further investigation and methodological innovations.

Event extraction based on joint learning

Compared to traditional pipeline approaches, joint methods demonstrate superior performance by sharing features and enabling inter-task information interaction, particularly excelling in capturing complex contextual dependencies and cross-sentence argument extraction. Cao et al.¹⁵ proposed OneEE, a model that reformulates event extraction as word-word relation identification through parallel grid tagging. It incorporates adaptive event fusion modules and distance-aware predictors to effectively mitigate error propagation. Dai et al.¹⁶ developed a cascaded decoding architecture with multi-feature fusion and condition-enhanced mechanisms, achieving robust performance in overlapping event extraction scenarios. Feng et al.² introduced a joint pointer labeling framework combining PERT pre-trained embeddings, event-type semantic augmentation, and SATT-BiLSTM feature extraction to resolve argument overlapping conflicts. Sheng et al.¹⁷ proposed SaltyFishes, a parameter-sharing joint learning framework that addresses low-resource event extraction through conditional normalization mechanisms, achieving state-of-the-art results in the CCKS-2020 financial event extraction competition. Lin et al.¹⁸ presented ONEIE, a global graph optimization framework integrating cross-task dependencies via beam search decoding and joint global feature modeling, enabling comprehensive performance improvements across multiple information extraction tasks. Chen et al.¹⁹ designed MLSL, a multi-layer sequence labeling approach for biomedical event extraction, which simplifies traditional complex workflows by explicitly modeling trigger-argument interactions while maintaining candidate trigger awareness.

While joint extraction methods provide streamlined architectures compared to non-joint approaches, their performance remains suboptimal in handling complex event interdependencies and long-range contextual dependencies, necessitating further optimization for domain-specific scenarios.

Generative event extraction

Generative event extraction is a paradigm that reformulates event extraction tasks as text generation problems. Unlike traditional classification or sequence labeling methods, this approach enables flexible mapping of input texts into structured event representations, unconstrained by fixed tag schemas. It demonstrates enhanced adaptability, particularly in multi-event coexistence scenarios. Jia et al.²⁰ developed an enhanced GPT2 model incorporating generative input modules and hybrid attention mechanisms, optimizing Transformer block outputs through layer-wise vector fusion strategies. Hsu et al.²¹ proposed DEGREE, a data-efficient event extraction framework that models the task as a conditional generation problem, achieving robust low-resource performance via manually crafted prompts. Duan et al.²² enhanced low-resource event extraction by integrating event keywords and fine-tuning BART with joint training objectives. Shi et al.²³ introduced an end-to-end joint extraction framework employing dual encoders to simultaneously leverage trigger-context interactions during text generation. Lu et al.²⁴ presented UIE, a unified text-to-structure generation framework that standardizes cross-task encoding through structured extraction languages. Chen et al.²⁵ designed CPEE, a generative joint event extraction model combining ChatGPT-based data augmentation with entity-aware prompt learning, demonstrating superior few-shot capabilities. Li et al.⁵ pioneered MQAEE, a multi-turn QA paradigm that sequentially extracts triggers and arguments via machine reading comprehension mechanisms.

Although generative approaches exhibit strong generalization capabilities and data efficiency in event extraction tasks, they still face some challenges such as generation instability and information omission. To address these issues, this paper proposes the PosEKE-GPT2 model, which enhances knowledge representation through extended positional encoding and a knowledge-augmented attention mechanism. By leveraging comprehensive textual information and capturing associations between event elements, the model significantly improves multi-event understanding and extraction capabilities, thereby mitigating information incompleteness to a certain extent.

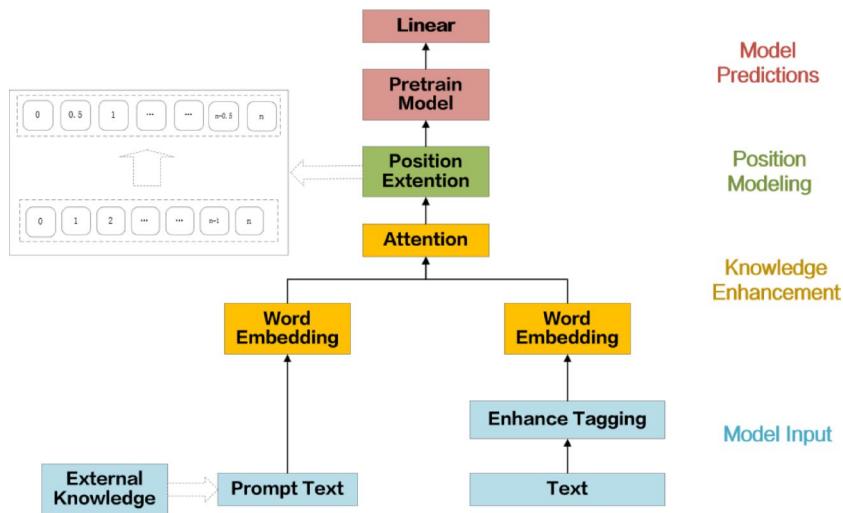


Fig. 3. PosEKE-GPT2 Model Diagram.

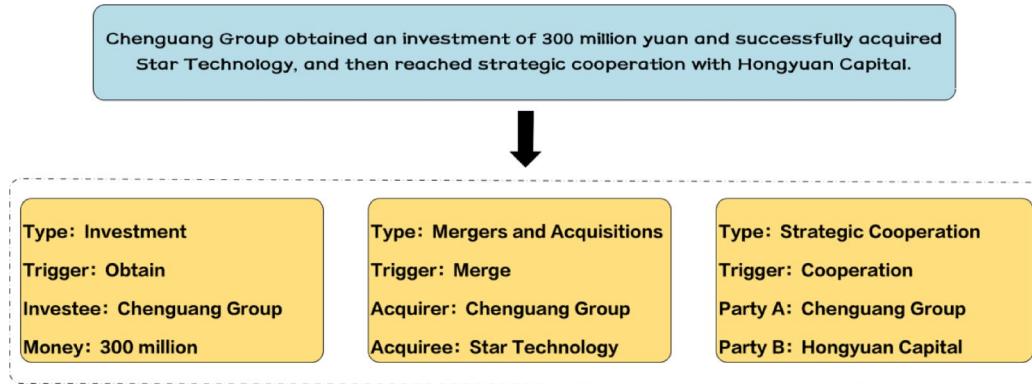


Fig. 4. Multi-event Text Example Diagram.

Model design

In this section, we elaborate on transforming event extraction into a conditional generation task based on prompt strategies, and propose an extended positional encoding method combined with a knowledge-augmented attention mechanism.

The architecture of the PosEKE-GPT2 model

PosEKE-GPT2 (Position Extension and Knowledge Enhancement on GPT2) extends the original GPT2 generative framework by enhancing positional encoding and incorporates knowledge augmentation through attention mechanisms guided by prompt strategies. As illustrated in Fig. 3, the model consists of four core modules: Model Input, Knowledge Augmentation, Positional Modeling, and Model Prediction.

Dual-Channel vocabulary prompting and Event-Augmented labeling

The model's input consists of two parts: input text and prompt text. To enable the model to better learn the meaning of text in complex contexts, this paper employs non-structured natural language text as input, allowing it to handle complex scenarios in real-world applications. Furthermore, the original data often contains multiple events, which further increases the complexity of the task. For example, Fig. 4 illustrates a multi-event extraction example from a financial news sentence.

To address the challenges of event argument extraction in multi-event scenarios, this paper proposes a method based on dual-channel dynamic lexicon prompting and event-enhanced annotation to optimize input representation and target sequence construction for event extraction tasks. The method employs a dual-channel architecture, where the dynamic lexicon prompting mechanism constructs event-related prompt words, while explicitly modeled text is annotated with special tokens to refine the representation of event elements.

Specifically, trigger words and arguments are dynamically imported from external lexicons to automatically generate schema-agnostic lexical prompts during training. The lexicon is built from two primary sources: (1) Event schema annotations provided in the DUEE-Fin and FewFC datasets, which supply canonical trigger and

argument labels; (2) Domain-specific terminology collected from publicly available financial news corpora and knowledge bases, with expert validation for semantic relevance and contextual applicability. This design ensures traceability and reproducibility of the lexicon resource. The standardized prompt format follows:

< Trigger>/n < Argument>.

Wherein: < Trigger> denotes the event trigger words, such as “announce”, “transfer”, “bankrupt”; and < Argument> represents event entities, such as “Sony”, “Alibaba”, “China Shandong Hi-Speed Financial Group Limited”.

The event-augmented labeling strategy optimizes event element representation by introducing a special token tagging mechanism during target sequence construction. During training data preprocessing, dedicated special tokens (e.g., [1], [2], [3], [9], [10]) are assigned to key elements including event types, triggers, and arguments. These tokens are then inserted into target sequences to explicitly annotate structural information of event elements. This labeling approach ensures format consistency across target sequences, enabling the model to learn structural patterns of each event element during training. It enhances comprehension of event compositions, improves event recognition capabilities, and establishes foundational support for subsequent joint extraction tasks.

Consider the following multi-event text as an example:

“Deyi Company announced to complete financing of 500 million yuan and acquire Jinfang Technology Company”.

The target generation sequence for this text is constructed as follows:

“[1] Enterprise financing [2] Financing [3] Financing party: Deyi Company [9] Amount: 500 million [9] [10]”.

“[1] Enterprise acquisition [2] Acquisition [3] Acquirer: Deyi Company [9] Acquiree: Jinfang Technology Company [9] [10]”

In the annotation schema, [1] denotes the start position of the entire text, [2] marks the end position of the event type, [3] indicates the end position of the trigger word, [9] signifies the end position of arguments, and [10] represents the termination position of the complete input text.

During the training phase, the dual-channel architecture facilitates enhanced learning of event prior knowledge and textual information through prompt-guided modeling and target sequence modeling. Subsequent experiments demonstrate that the integration of dual-channel dynamic vocabulary prompting and special token tagging enhances the model’s generalization capability, thereby preventing overfitting to single-event patterns. By incorporating external knowledge bases, the method significantly improves the recognition accuracy for diverse events, effectively addressing event element extraction in multi-event scenarios.

Knowledge augmentation module

The first core component of the Knowledge Augmentation Module is the Word Embedding Encoding Layer. This component is constructed based on the vanilla GPT2 pre-trained model, with its primary function being the transformation of raw input text into embedding vectors interpretable by the model. Notably, to accommodate subsequent positional encoding expansion requirements, a position-agnostic processing strategy is adopted at this stage—retaining only the semantic embeddings of the text while temporarily excluding any positional encoding information. This process is illustrated in Fig. 5.

The mapping equations for the input texts S_t ($t = 1, 2, \dots, n$) and prompt texts S_p ($p = 1, 2, \dots, m$) are given in Eqs. (1) and (2):

$$X_t = E \bullet W_t \quad (1)$$

$$X_p = E \bullet W_p \quad (2)$$

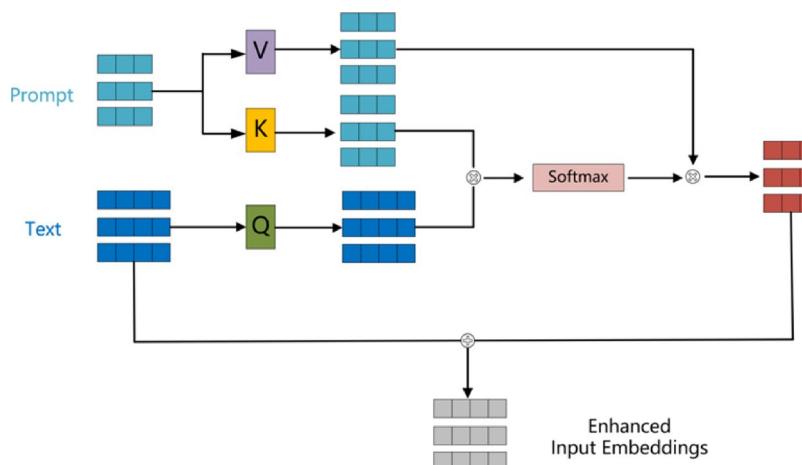


Fig. 5. Knowledge Augmentation Process Diagram.

Here, the token embedding matrix of GPT2 is denoted as $E \in \mathbb{R}^{V \times e}$, where V is the vocabulary size and e is the embedding dimension. $W_t \in \mathbb{R}^{s \times V}$ and $W_p \in \mathbb{R}^{p \times V}$ represent the one-hot representation matrices of the input text and prompt text, respectively. X_t and X_p correspond to the token embedding representations of the input text and prompt text.

To enhance the model's awareness of domain-specific knowledge, we introduce an attention-based knowledge augmentation method. Building upon the vanilla GPT2 token embeddings and given the model's reliance on external knowledge, an attention mechanism is adopted to compute attention scores between textual elements, enabling dynamic weighted fusion of external knowledge.

First, we compute the relevance between the input text and prompt text. As obtained in the previous step, the token embeddings of the input text and prompt text are denoted as $X_t \in \mathbb{R}^{b \times s \times e}$ and $X_p \in \mathbb{R}^{b \times p \times e}$, respectively, where b is the batch size, s is the input text length, e is the embedding dimension, and p is the prompt sequence length. We measure their relevance via inner product computation, denoted as A_p as shown in Eq. (3):

$$A_t[b, s, p] = X_t[b, s, p] \bullet X_p[b, p, e] \quad (3)$$

After obtaining the relevance scores A_p , we normalize them and convert them into attention weights using the Softmax function. First, we sum the exponentiated scores across all prompt positions to compute the normalization term, denoted as $Z[b, s]$, where p_i represents all possible prompt positions and m denotes the number of prompt tokens. The specific formulation is given in Eq. (4):

$$Z[b, s] = \sum_{i=0}^m e^{A_t[b, s, p_i]} \quad (4)$$

We then normalize the attention scores for each prompt position, transforming them into a probability distribution. The final attention scores, denoted as A_{score} , are computed where p_i represents the prompt position index for the current softmax normalization, as detailed in Eqs. (5) and (6).

$$A_{score}[b, s, p_i] = \frac{e^{A_t[b, s, p_i]}}{Z[b, s]} \quad (5)$$

$$A_{score}[b, s, p_i] = \frac{e^{A_t[b, s, p_i]}}{\sum_{i=0}^m p_i e^{A_t[b, s, p_i]}} \quad (6)$$

The computed attention weights A_{score} are combined with the token embeddings of the prompt text via weighted summation, which constitutes the core operation of the attention mechanism. For each input position s , the model calculates the weighted sum across all prompt position p_i based on their token embeddings, yielding the fused output representation as specified in Eq. (7):

$$A_{out}[b, s, e] = \sum_{i=0}^m A_{score}[b, s, p_i] \bullet X_p[b, p_i, e] \quad (7)$$

Finally, the weighted prompt information A_{out} is integrated into the input text's token embedding X_t to facilitate the subsequent positional encoding expansion and prediction tasks. The resulting knowledge-augmented information, denoted as K_{out} , is formulated as shown in Eq. (8):

$$K_{out} = X_t[b, s, e] + A_{out}[b, s, e] \quad (8)$$

Extended positional encoding module

In Transformer models, positional encoding serves as a critical component. Since the self-attention mechanism inherently lacks the capability to discern positional relationships between elements in a sequence, positional encoding addresses this limitation by injecting positional information, thereby enabling the model to comprehend the sequential order of elements in the sequence. Traditional absolute positional encoding methods typically employ sinusoidal function and cosine function to compute positional embeddings. However, GPT2, as a generative model, adopts learnable absolute positional encoding, where each position is assigned a trainable vector. This design allows the model to dynamically learn semantic patterns associated with different positions, making it better suited for complex task requirements.

GPT2 maintains a trainable positional embedding matrix with a maximum sequence length L and an embedding dimension d_m , as formalized in Eq. (9).

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,d_m} \\ P_{1,0} & P_{1,1} & \dots & P_{1,d_m} \\ \dots & \dots & \dots & \dots \\ P_{L-1,0} & P_{L-1,1} & \dots & P_{L-1,d_m} \end{bmatrix} \quad (9)$$

For the input text S_t ($t = 1, 2, \dots, n$), the position encoding p_i corresponding to the original position i of each word can be computed, where $W_p[i]$ denotes the vector in the i -th row of the learnable position embedding matrix. The formula is shown in Eq. (10).

$$P_i = W_p[i] \quad (10)$$

However, as the input sequence length increases, the fixed-range limitation of positional encoding constrained by maximum sequence length may hinder the model's ability to effectively capture long-range dependencies between distant words. When processing sequences exceeding the pre-defined maximum training length, the original positional encoding scheme becomes inapplicable. To address this issue, this paper proposes a novel positional encoding method that achieves smooth transition of positional information through adjacent positional encoding fusion. This approach breaks through the fixed-length constraints of conventional positional embeddings, enabling more continuous and scalable representation of positional relationships.

This positional encoding method helps the model better capture positional information in long texts, overcoming the input length limitation of the original GPT2 and increasing the input text volume, thereby enhancing the model's ability to model long texts.

The process of extending positional encoding involves generating new positional encodings by fusing adjacent positional encodings. As illustrated in Fig. 6, the first step requires fusing every two adjacent positions in the original positional encoding matrix E , with the formula expressed as Eq. (11):

$$E_{avg(i)} = \frac{E_i + E_{i+1}}{2}, i = 1, 2, \dots, L - 1 \quad (11)$$

Where E_i denotes the positional encoding of position i , E_{i+1} denotes the positional encoding of position $i + 1$, and $E_{avg(i)}$ represents the fused encoding, the average value of the original positions i and $i + 1$.

After obtaining the fused encoding, all positional encodings will undergo interpolation-based fusion processing to form a new extended positional encoding matrix, as shown in Eq. (12):

$$E_{ext} = [E_1, E_{avg(1)}, E_2, E_{avg(2)}, \dots, E_{L-1}, E_{avg(L-1)}, E_L] \quad (12)$$

This implementation is relatively simple and flexible. Despite introducing adjacent positional encoding fusion, the newly designed positional encoding through stacking-based design effectively reduces computational cost while avoiding the complexity caused by independently encoding each position. Additionally, the new positional encoding preserves original positional information and incorporates additional positional cues, enabling the model to capture positional relationships more accurately and ultimately enhancing the performance of event extraction tasks.

Model prediction

We reformulate the event extraction task as a joint generation task. During prediction, the model adopts a task-step generation approach, sequentially generating outputs in the order of event type \rightarrow trigger \rightarrow arguments. In each prediction step, the model generates the next target output based on the current task input and partial prediction results from the previous step.

To achieve joint extraction, we set different objectives for various subtasks and perform task-based autoregressive decoding in a predefined task sequence. Based on the input text, the model first predicts the event type y_{event} . After determining the event type, the model initializes new input and generates the event trigger word $y_{trigger}$. Subsequently, based on the trigger word, the model re-initializes new input and progressively generates arguments $y_{argument}$, including roles and corresponding entities. The specific computational formulas are shown in Eqs. (13), (14), (15), and (16).

$$P_1 = P(y_{event}|x) = \text{Softmax}(W_{event}H_{event} + b) \quad (13)$$

$$P_2 = P(y_{trigger}|x, y_{event}, y_{trigger}) = \text{Softmax}(W_{trigger}H_{trigger} + b_{trigger}) \quad (14)$$

$$P_3 = P(y_{argument}|x, y_{event}, y_{trigger}) = \text{Softmax}(W_{argument}H_{argument} + b_{argument}) \quad (15)$$

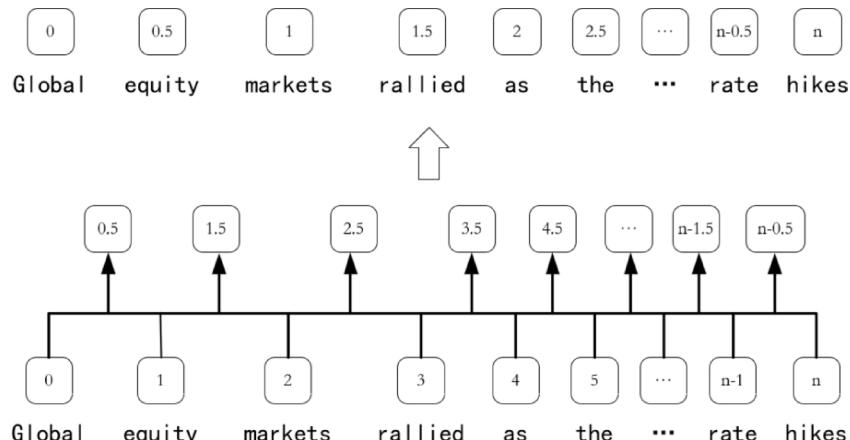


Fig. 6. Positional Encoding Extension Schematic Diagram.

$$P(y_{\text{event}}, y_{\text{trigger}}, y_{\text{argument}} | x) = P_1 \bullet P_2 \bullet P_3 \quad (16)$$

wherein, H represents the hidden states of each task, W represents the weight matrix of each task, and b is the bias term of each task. Specifically, H_{event} represents the hidden state of the input text, H_{trigger} combines event type information, and H_{argument} integrates event type and trigger word information. Therefore, W_{event} , W_{trigger} and W_{argument} are used for the mapping transformations of event type classification, trigger word recognition, and argument extraction respectively, while the corresponding bias terms b_{event} , b_{trigger} and b_{argument} are used to adjust the prediction bias of the model. Finally, the maximum value of each task's prediction is taken as the final choice to obtain the complete extraction results of event types, trigger words, and arguments.

Loss function

Given that the entire event extraction process has been rephrased as a conditional text generation task, the training objective is to maximize the accuracy of generating the target sequence given input text and prompts. Accordingly, we employ the standard token-level cross-entropy loss for sequence generation, which is a common practice for autoregressive language models like GPT-2.

The loss is defined as:

$$\mathcal{L}_{\text{loss}} = - \sum_{t=1}^T \log P(y_t | y < t, K_{\text{out}}) \quad (17)$$

where y_t denotes previously generated tokens, T is the sequence length and K_{out} is the used representation of text and prompt.

Experimental results and analysis

Dataset and parameters

The experiment adopted the DuEE-Fin²⁶ financial domain open-source document-level event extraction dataset. This dataset contains a total of 7,250 annotated texts, including 1,179 test data entries, encompassing 13 event types and 9,440 events. In addition, the FewFC dataset was incorporated, consisting of 7,185 sentences from 899 texts, containing 10 event types and 3,172 event instances.

In this experiment, we adopted gpt2-distil-chinese-cluecorpussmall as the baseline model and implemented improvements upon it. The experiments were conducted using the PyTorch framework, with multiple hyperparameters adjusted during the training process, including batch size, learning rate, and optimizer. Detailed hyperparameter configurations are presented in Table 1, while the training details regarding batch iterations and epochs are further described in the section "Training Configuration and Convergence Analysis".

Analysis of experimental results

To verify the effectiveness of the PosEKE-GPT2 model in the event extraction task, we designed three groups of experiments. First, we compared the PosEKE-GPT2 with multiple mainstream baseline models to evaluate its overall performance advantages. Second, through ablation experiments, we progressively removed different modules in the model to analyze their impact on overall performance, thereby validating their necessity. Finally, we designed five distinct methods for the knowledge-enhanced module and screened out the optimal knowledge-enhancement strategy through comparative experiments to further improve model performance.

In the three subtask experiments, the model's performance was evaluated utilizing three metrics: Precision (P), Recall (R), and F1-score (F1)²⁷. The calculation formulas for these three metrics are shown in Eqs. (17), (18), and (19), respectively:

$$P = \frac{TP}{TP + FP} \quad (17)$$

$$R = \frac{TP}{TP + FN} \quad (18)$$

$$F1 = \frac{2 \bullet P \bullet R}{P + R} \quad (19)$$

Parameter	Value
Batch_Size	10
Lr	5e-4
Optimizer	Adam
Epoch	7
Dropout	0.1

Table 1. Experimental parameter settings.

Comparison with mainstream baseline models

In this experimental section, we demonstrate the effectiveness of PosEKE-GPT2 in the event extraction task through comparison with the following models:

- BERT²⁸: BERT itself serves as a powerful pre-trained language model that can effectively capture contextual information, which is the most critical component in pipeline-based event extraction. It inherently possesses the capability to handle both sequence labeling tasks and classification tasks.
- GPT2²⁹: As a generative model, GPT2 learns the latent relationships between trigger words and arguments based on contextual information, accomplishing event type classification, trigger identification, and argument extraction simultaneously through joint extraction.
- BERT + MMOE + CRF³⁰: By leveraging BERT to extract semantic information, ensuring precise modeling of semantic features for each token, we introduce a multi-gate mixture-of-experts module that facilitates effective information sharing across different subtasks of event extraction through shared learning and expert gating mechanisms. Finally, CRF is incorporated into the output layer of the model to model dependency relationships among labels.
- JEEDG³⁰: By explicitly separating shared parameters and task-specific parameters, the introduction of a dual-layer gated network enhances the extraction and filtering capabilities of semantic knowledge.
- CasEE³¹: The multi-level event extraction framework based on the BERT encoder identifies event core elements through three decoders: event type detection, trigger word extraction, and argument extraction in sequence, combined with self-attention mechanism and conditional fusion function to achieve structured semantic parsing.

The performance comparison between the proposed PosEKE-GPT2 model and benchmark models is shown in Table 2.

From the experimental results in Table 2, it can be seen that the performance of each model varies significantly in the event extraction task. In event type classification, BERT achieved an F1 of 93.86, demonstrating its strong contextual understanding, while the original GPT2 reached an F1 of 90.06. The PosEKE-GPT2 model proposed in this study achieved the best performance with an F1 of 94.88 through knowledge enhancement and positional encoding extension, verifying its advantage in capturing fine-grained semantic information. In trigger extraction, PosEKE-GPT2 led all comparison models with an F1 of 91.22, outperforming BERT-MMOE-CRF's 85.60 and JEEDG's 86.58, highlighting the effectiveness of its knowledge enhancement in trigger recognition. In argument extraction, PosEKE-GPT2 again achieved the highest F1 of 85.74, exceeding CasEE's 81.24 and BERT's 66.7, indicating that the positional encoding extension effectively captures complex argument relations. Overall, PosEKE-GPT2 achieved the highest mean F1 of 90.61, surpassing GPT2's 85.15 and CasEE's 86.41, fully demonstrating the synergistic effect of knowledge enhancement and positional encoding extension in multi-task joint learning, particularly in integrating cross-subtask contextual information and modeling long-range dependencies.

As shown in Table 3, similar trends were observed in the FewFC dataset. In event type classification, PosEKE-GPT2 achieved the best F1 of 92.48, outperforming GPT2's 82.69 and CasEE's 88.64, confirming its superior semantic representation ability. In trigger extraction, PosEKE-GPT2 reached an F1 of 91.70, clearly higher than GPT2's 82.88 and CasEE's 82.88, showing its robustness in identifying diverse event triggers. In argument extraction, PosEKE-GPT2 achieved an F1 of 82.36, surpassing BERT's 68.16 and CasEE's 76.91, further validating the contribution of the positional encoding extension in capturing complex argument dependencies. Overall, PosEKE-GPT2 achieved the highest mean F1 of 88.85, significantly outperforming GPT2's 80.85 and CasEE's 82.81, demonstrating the consistent effectiveness of the proposed improvements across different datasets.

In summary, the proposed PosEKE-GPT2 model consistently demonstrated superior performance on both the DuEE-Fin and FewFC datasets, confirming its robustness and effectiveness in domain-specific event extraction tasks across different benchmarks.

Ablation experiment

To analyze the contribution of each module to the overall event extraction task, this paper conducted the following ablation experiments: removing the extended position embedding (ext_pos) and knowledge-enhanced

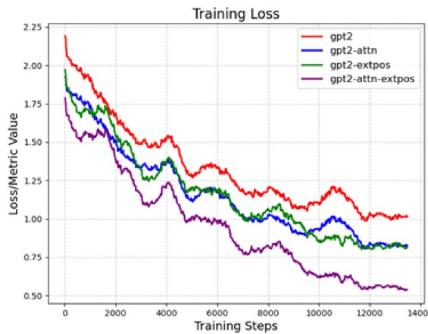
Model	Event Type Classification			Trigger Extraction			Argument Extraction			Mean F1
	P	R	F1	P	R	F1	P	R	F1	
BERT	93.66	94.05	93.86	83.2	80.23	81.65	80.93	61.5	66.7	80.80
JEEDG	-	-	-	86.46	86.70	86.58	66.36	67.40	66.88	76.73
BERT-MMOE-CRF	-	-	-	83.72	87.70	85.60	66.01	66.43	66.22	75.91
CasEE	86.81	90.42	88.27	89.94	89.58	89.73	83.15	85.06	81.24	86.41
GPT2	88.48	89.64	90.06	85.32	86.37	85.84	78.90	80.24	79.56	85.15
PosEKE-GPT2(ours)	94.30	95.36	94.88	90.58	91.75	91.22	85.08	86.42	85.74	90.61

Table 2. Comparative experiment results table (Duee-Fin).

Model	Event Type Classification			Trigger Extraction			Argument Extraction			Mean F1
	P	R	F1	P	R	F1	P	R	F1	
BERT	89.92	79.00	84.00	80.17	61.82	69.82	74.63	62.73	68.16	73.99
BERT-CRF-Joint	90.83	79.78	84.43	80.72	63.00	70.81	74.23	61.18	67.14	74.13
CasEE	87.50	87.73	88.64	77.90	78.53	82.88	76.83	77.12	76.91	82.81
GPT2	82.08	83.30	82.69	82.35	83.41	82.88	76.38	77.62	76.99	80.85
PosEKE-GPT2(ours)	91.88	93.10	92.48	91.18	92.24	91.70	81.74	82.99	82.36	88.85

Table 3. Comparative experiment results table (FewFC).

Model	Event Type Classification			Trigger Extraction			Argument Extraction			Mean F1
	P	R	F1	P	R	F1	P	R	F1	
-all (GPT2)	88.48	89.64	90.06	85.32	86.37	85.84	78.90	80.24	79.56	85.15
-ext_pos	90.30	91.46	90.88	88.40	86.45	84.92	81.75	83.09	82.42	86.07
-KB	91.52	92.67	92.09	88.19	86.24	84.71	82.94	84.28	83.60	86.80
PosEKE-GPT2(ours)	94.30	95.36	94.88	90.58	91.75	91.22	85.08	86.42	85.74	90.61

Table 4. Ablation study results Table.**Fig. 7.** Ablation Study Training Loss Plot.

(KB) modules respectively, and observed the changes in model performance. The experimental results are shown in Table 4.

From the comparison of experimental results, the full-version PosEKE-GPT2 model demonstrates clear advantages in event extraction. In event type classification, it achieves an F1 of 94.88. Removing the position expansion module slightly reduces the F1 to 90.88, while removing the knowledge enhancement module results in an F1 of 92.09, indicating that knowledge enhancement mainly stabilizes performance across subtasks rather than directly boosting classification. For trigger extraction, the full model attains an F1 of 91.22; removing position expansion or knowledge enhancement slightly alters the F1 to 84.92 and 84.71 respectively, suggesting the modules jointly balance performance rather than individually maximizing trigger recognition. In argument extraction, the full model reaches the highest F1 of 85.74, whereas removing position expansion or knowledge enhancement reduces it to 81.75 and 82.94 respectively, showing both modules are crucial for capturing complex argument relations, with position expansion having a slightly stronger effect. Overall, the mean F1 of the full model is 90.61, surpassing the variant without position expansion 86.07, without knowledge enhancement 86.80, and the original GPT2 baseline 85.15, fully illustrating the synergistic effect of the two modules in multi-task joint learning.

As shown in Figs. 7 and 8, PosEKE-GPT2 demonstrates superior convergence in both training loss and validation loss compared to other incomplete models, exhibiting a more stable optimization process and stronger generalization capability.

Qualitative case analysis

To further demonstrate the advantages of the proposed model, we present a representative case from financial news, as illustrated in Fig. 9. The example sentence is: “Tencent announced a 500-million-yuan investment yesterday and completed the acquisition of a gaming studio in Shanghai.”

As shown in Fig. 9, the baseline model correctly identified the “investment” event, extracting “Tencent” as the investor and “500 million yuan” as the amount. However, it failed to detect the subsequent “acquisition”

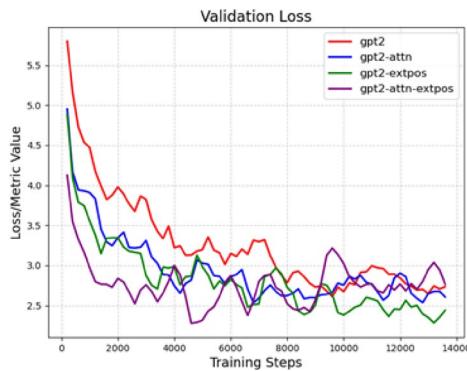


Fig. 8. Ablation Study Validation Loss Plot.

Tencent announced a 500 million yuan investment yesterday and completed the acquisition of a gaming studio in Shanghai		
GPT-2		PosEKE-GPT2
Event:Investment	Event:Investment	Event:Acquisition
Trigger:investment	Trigger:investment	Trigger:acquisition
Investor:Tencent	Investor:Tencent	Investor:Tencent
Amount:500 million yuan	Amount:500 million yuan	Amount:A gaming studio in Shanghai

Fig. 9. Quantitative Analysis.

event and did not link the target entity “a gaming studio in Shanghai” to the corresponding trigger. This suggests that the baseline model struggles with multi-event sentences involving long-range dependencies, often being influenced by the most salient event and suffering from trigger omission and argument loss.

In contrast, PosEKE-GPT2 successfully extracted both events in their entirety, accurately identifying all triggers and associated arguments—including the distantly located acquisition target. This performance underscores the complementary benefits of the two core modules. The Knowledge Enhancement Module integrates external financial knowledge bases, supplying semantic priors that aid in recognizing less frequent yet domain-relevant triggers such as “acquisition,” thereby mitigating omissions common in the baseline. Meanwhile, the Positional Encoding Extension Module improves the model’s ability to capture long-range dependencies by interpolating intermediate positional encodings through averaging adjacent token representations. This enhancement facilitates the connection between triggers and their distant arguments, such as associating “acquisition” with “a gaming studio in Shanghai” effectively addressing the argument-missing issue observed in the baseline.

This case clearly illustrates that knowledge enhancement boosts trigger detection, while extended positional encoding significantly improves long-distance argument linking. Overall, the two modules enable accurate and comprehensive extraction of multiple events from complex financial sentences.

Comparative experiments on knowledge-enhanced modules

In the experimental section, to verify the effectiveness of fusing knowledge vectors and input text word embedding vectors through the attention mechanism for knowledge enhancement, this paper further designs five different fusion methods for comparative experiments. These five methods are as follows:

- Direct Addition (ADD): Directly add the knowledge vector to the input text embedding vector to test the effectiveness of the simple vector superposition approach.
- Prepend concatenation (Pre-concat): The knowledge vector is concatenated to the beginning of the input text to provide additional contextual background information.
- Post-concatenation (Post-concat): Concatenate the knowledge vector to the end of the input text and observe the impact of knowledge integration at different positions on text comprehension.
- Graph Attention Network Fusion (GAT): Adopt a Graph Attention Network to fuse knowledge vectors and input text word embedding vectors, thoroughly modeling the semantic associations between them and enhancing the depth of information interaction.

Model	Event Type Classification				Trigger Extraction				Argument Extraction				Mean F1
	P	R	F1		P	R	F1		P	R	F1		
ADD	91.88	93.03		92.45	89.32	90.37		89.84	82.54	83.88	83.21	88.50	
Pre-concat	92.48	93.64		92.06	91.16	92.20		91.68	81.20	82.54	81.86	88.53	
Post-concat	93.30	94.46		94.88	89.32	90.37		89.84	79.77	81.11	80.44	88.39	
GAT	93.09	93.24		93.66	87.94	88.99		88.47	82.26	83.61	82.93	88.35	
Attention	94.30	95.36		94.88	90.58	91.75		91.22	85.08	86.42	85.74	90.61	

Table 5. Comparison results table of Knowledge-enhanced Methods.

Ours	Natural Language	Keywords	QA-Format
<Trigger> [Candidate event types]	Please identify the event type described in the text. Possible types include: type1,type2,...	Event-Type:type1,type2	Q: What type of event is described in the text? Options: type1,type2,...
<Argument> [Candidate roles]	Please extract the arguments of different roles in the event. Possible roles include: role1,role2,...	Roles:role1,role2	Q: What participant information is contained in the text? Role types: role1,role2,...

Fig. 10. Illustration of Different Prompt Design Strategies.

- Attention mechanism fusion (ATTN): By introducing attention mechanisms, the model can learn the relevance between input text and prompt text, enhance semantic representation, and improve the performance of generation tasks.

Detailed experimental results are presented in Table 5.

From the experimental results in Table 5, it can be seen that different knowledge fusion strategies have notable effects on event extraction performance. The attention-based fusion method performs best, achieving an average F1 value of 90.61, which surpasses all other strategies, demonstrating its effectiveness in enabling fine-grained interaction between text and external knowledge through dynamic weighting. The direct addition method reaches an average F1 value of 88.50, ranking lowest, suggesting that simple vector summation may introduce semantic conflicts. Forward concatenation and backward concatenation strategies obtain average F1 values of 88.53 and 88.39 respectively. By adjusting the position of knowledge embeddings, they mitigate some information conflicts, but static concatenation still leads to uneven representation distribution. The graph attention network strategy achieves an average F1 of 88.35, indicating limited structural modeling ability in long-text event extraction scenarios. Overall, within the joint event extraction framework, the attention mechanism enables context-aware dynamic fusion, improving cross-modal knowledge integration and outperforming the sub-optimal backward concatenation strategy by 2.22% points, highlighting the key role of dynamic knowledge fusion in capturing complex semantic relationships.

Comparative analysis of prompting strategies

To validate the effectiveness of our proposed concise prompting strategy—which is characterized by its simplicity and direct use of lexical knowledge—we compare it against several alternative, more elaborate prompting designs in a controlled ablation study. A comparative visualization of these four prompting strategies is presented in Fig. 10. This comparison includes:

- Ours (T1): Our proposed approach employs a concise template following the “<Trigger>\n<Argument>” format, delivering pure lexical knowledge without additional instructional markers or syntactic structures. The prompt consists solely of newline-separated lists of trigger words and argument roles, facilitating direct association learning between lexical knowledge and textual context.
- Natural(T2): This strategy uses natural language instructions to frame the task in a human-readable form. Adopting prompts such as “Please identify the event type described in the text. Possible types include: <Trigger_list>” and “Please extract the arguments for different roles in the event. Possible roles include: <Argument_list>”, it evaluates the model’s ability to comprehend and respond to intuitive conversational directives.
- Keywords(T3): It is a minimalist keyword-style prompt that reduces instructional context to its bare essentials. Using succinct formulations like “Event Type: <Trigger_list>” and “Arguments: <Argument_list>”, this

- method examines the model's reliance on rich instructional context and its capacity to infer task requirements from minimal semantic cues.
- QA-Format(T4): This approach reformulates the extraction task as an interactive question-answering session. With prompts such as “What type of event is described in the text? Options: <Trigger_list>” and “What participant information is contained in the text? Role types: <Argument_list>”, it explores alternative task formulations that may activate different reasoning pathways in the language model.

As shown in Table 6, different prompt formulations lead to clear variations in performance, confirming that prompt design plays a crucial role in generative event extraction. The QA-style prompt achieved the highest F1-score of 94.35 in event type classification, but its performance in argument extraction dropped significantly to only 80.44. This indicates that framing the task as a question favors coarse-grained classification but provides limited guidance for capturing fine-grained structural information. The keyword-based prompt produced the weakest overall results with a mean F1-score of 87.84, demonstrating that overly simplified instructions are insufficient for guiding complex extraction tasks. In comparison, the natural language prompt demonstrated balanced improvements across all subtasks, attaining a mean F1-score of 89.32, which suggests that intuitive and human-readable instructions enhance the model's generalization ability. Notably, our structured prompt design delivered the best overall performance with a mean F1-score of 90.61, while achieving particular advantages in both trigger and argument extraction. These findings highlight the sensitivity of generative models to prompt formulation and demonstrate the effectiveness of structured supervision combined with knowledge injection for improving robustness and accuracy.

Training configuration and convergence analysis

To evaluate the adequacy of the training setup, we further investigated the impact of different batch size and epoch configurations on model performance. The objective of this experiment was to assess the convergence behavior of the proposed model and to examine its robustness under varying training conditions.

Table 7 presents a comprehensive comparison of different batch sizes and training epochs, demonstrating that both factors significantly influence model performance. When the number of epochs is fixed at seven, larger batch sizes consistently lead to improved results. Models trained with batch sizes of four or six achieve relatively lower mean F1 scores, while a batch size of eight yields stronger performance. The highest overall mean F1 score of 90.61 is attained with a batch size of ten. Although a batch size of four results in a marginally higher F1 score for argument extraction compared to a batch size of eight, this advantage is offset by a decline in overall performance, indicating that very small batch sizes may provide limited regularization at the cost of reduced stability.

Holding the batch size constant at ten and, increasing the number of training epochs from four to seven consistently improves model outcomes. The mean F1 score rises from 87.37 at four epochs to 89.28 at five epochs, and further to 89.92 at six epochs, reaching a peak of 90.61 at seven epochs under the B10E7 configuration. Similarly, the argument extraction F1 score improves to 85.74 at seven epochs, confirming that longer training enhances both overall performance and argument extraction capabilities. As shown in Fig. 11, the training loss plateaus around the seventh epoch, suggesting that the model has converged. However, extending training to eight epochs leads to a slight degradation in performance, with the mean F1 decreasing to 89.15 and argument extraction F1 dropping to 82.43, suggesting the onset of overfitting.

These findings indicate that a batch size of ten combined with seven training epochs achieves the optimal balance between convergence and generalization.

Length-sensitivity and efficiency analysis

To further evaluate the effectiveness of the proposed extended positional encoding mechanism, experiments were conducted from two aspects: length generalization and computational efficiency.

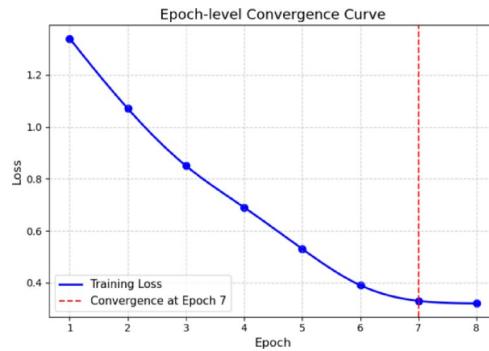
For the length generalization test, the dataset was divided into four intervals based on sequence length: short texts (0–99 tokens), medium-short texts (100–199 tokens), medium-long texts (200–299 tokens), and long texts (300 + tokens). The sample sizes for each interval are provided in Table 8.

As presented in Table 9, PosEKE-GPT2 consistently outperforms the baseline GPT2 across all length intervals, achieving higher mean F1 scores and demonstrating the robustness of the proposed mechanism. Although the improvement is moderate on short and medium-short texts, a more substantial gain is observed on medium-long and long texts. Notably, in the 300 + token interval, PosEKE-GPT2 attains a mean F1 score of 87.74, significantly surpassing the 75.66 achieved by GPT2. This result confirms that the proposed positional encoding extension effectively alleviates the performance degradation commonly associated with longer sequences.

Model	Event Type Classification			Trigger Extraction			Argument Extraction			Mean F1
	P	R	F1	P	R	F1	P	R	F1	
QA_Format	93.70	95.01	94.35	88.86	90.03	89.44	79.93	80.96	80.44	88.07
Keywords	91.88	93.19	92.53	89.32	90.49	89.90	80.60	81.63	81.11	87.84
Natural	93.09	94.40	93.74	90.70	91.86	91.28	82.42	83.45	82.94	89.32
Ours	94.30	95.36	94.88	90.58	91.75	91.22	85.08	86.42	85.74	90.61

Table 6. Prompt strategy result Table.

Model	Event Type Classification			Trigger Extraction			Argument Extraction			Mean F1	Time(min/Total)
	P	R	F1	P	R	F1	P	R	F1		
B4E4	88.38	89.87	89.17	87.16	88.63	87.88	82.15	83.59	82.86	86.63	136.5
B6E4	89.09	90.57	89.82	85.78	87.62	86.51	81.24	82.64	81.93	86.08	131.5
B8E4	87.27	88.50	88.88	85.82	86.21	85.51	81.67	82.77	82.22	85.53	132.5
B10E4	93.09	94.24	93.69	85.19	86.34	85.76	82.03	83.33	82.67	87.37	139.4
B4E5	89.09	90.16	89.62	83.94	85.35	85.64	85.00	86.03	85.51	86.92	169.5
B6E5	88.89	90.11	89.60	91.16	90.61	91.61	82.82	83.95	83.38	88.20	166.3
B8E5	89.09	92.39	91.74	89.32	90.41	89.86	81.24	82.27	81.75	87.78	164.1
B10E5	91.88	93.11	92.49	89.32	90.71	90.01	83.02	85.90	85.34	89.28	173.9
B4E6	87.27	88.58	87.92	87.40	88.57	87.98	81.95	82.98	82.46	86.12	204.4
B6E6	88.48	89.87	89.17	89.24	90.71	89.97	80.59	82.04	81.31	86.81	198.0
B8E6	90.09	91.31	92.70	89.16	91.22	91.68	81.31	82.56	81.93	88.77	199.5
B10E6	91.88	93.18	94.53	91.16	92.24	91.70	84.80	84.05	83.53	89.92	207.0
B4E7	91.27	92.58	91.92	88.40	89.57	88.98	81.95	82.98	82.46	87.79	195.1
B6E7	93.09	93.48	93.78	88.86	90.33	89.59	77.79	79.24	78.51	87.29	197.4
B8E7	93.09	94.31	93.70	91.16	92.22	91.68	81.31	82.56	81.93	89.10	198.0
B10E7(Ours)	94.30	95.36	94.88	90.58	91.75	91.22	85.08	86.42	85.74	90.61	209.2
B4E8	92.48	93.60	93.04	91.16	92.19	90.67	80.72	81.87	81.29	88.33	270.0
B6E8	89.70	90.70	92.20	87.16	88.56	91.85	80.60	81.96	81.27	88.44	263.6
B8E8	91.27	92.66	91.96	89.79	90.79	90.05	84.13	85.57	84.84	88.95	264.6
B10E8	93.19	94.11	93.60	90.70	92.15	91.42	81.87	83.00	82.43	89.15	278.9

Table 7. Ablation study on training configurations (Batch size & Epochs).**Fig. 11.** Epoch-Level Convergence Curve.

Length Interval	Category	Number of Samples
0–99	Short texts	2274
100–199	Medium-short texts	2084
200–299	Medium-long texts	1624
300+	Long texts	1268

Table 8. Distribution of samples across sentence length intervals.

In this study, we compared PosEKE-GPT2 with two baseline models: GPT2-base and GPT2 with sinusoidal positional embeddings. As summarized in Table 10, PosEKE-GPT2 achieves the highest mean F1 score of 90.61%, while maintaining comparable inference speed and per-epoch training time to GPT2-base. These results indicate that the proposed extension incurs negligible computational overhead while consistently improving performance across evaluations.

These findings collectively demonstrate that the proposed positional encoding extension mechanism (1) enhances model robustness across varying text lengths, performing particularly well on long-text scenarios, and (2) delivers consistent accuracy gains without sacrificing computational efficiency. These results underscore

Interval	Model	Event Type Classification				Trigger Extraction				Argument Extraction				Mean F1
		P	R	F1		P	R	F1	P	R	F1			
0-99	PosEKE-GPT2	93.94	95.13		94.53	95.69	97.17		96.43	75.84	77.21	76.52	89.16	
	GPT2	95.15	96.34		95.74	91.60	92.08		91.33	70.70	72.07	71.38	86.15	
100-199	PosEKE-GPT2	92.29	93.70		92.99	93.66	94.76		94.21	76.26	77.35	76.80	88.00	
	GPT2	91.67	92.76		92.21	89.76	90.01		90.33	70.65	70.73	70.18	84.24	
200-299	PosEKE-GPT2	92.87	93.90		93.39	89.99	91.42		90.70	77.83	79.13	78.38	87.49	
	GPT2	91.14	92.33		91.73	93.72	95.15		94.43	71.01	72.38	71.69	85.95	
300+	PosEKE-GPT2	92.86	93.95		93.40	91.71	92.86		92.28	76.90	78.17	77.53	87.74	
	GPT2	76.83	78.01		77.41	89.04	90.51		89.77	59.13	60.50	59.81	75.66	

Table 9. GPT2 vs. PosEKE-GPT2 across length Intervals.

Model	Mean F1 Inference Speed (samples/s)	Inference Speed (samples/s)	Training Time (min/epoch)
GPT2	85.51	35.2	18.5
GPT2- Sinusoidal PE	88.71	34.8	19.1
PosEKE-GPT2 (ours)	90.61	35.2	18.7

Table 10. Performance and efficiency comparison of positional encoding Mechanisms.

the practical value of PosEKE-GPT2 for document-level event extraction tasks that involve long-distance dependencies.

Conclusion

In this paper, we propose a model named PosEKE-GPT2 and elaborate on its architecture. We then conduct experiments on the Duee-Fin and FewFC datasets, validating the effectiveness of PosEKE-GPT2 in financial domain event extraction tasks. Experimental results demonstrate that our model surpasses all baseline models in the Mean F1 metric, proving its overall superiority.

Compared to traditional methods, PosEKE-GPT2 significantly improves extraction performance in jointly extracting multiple events through positional extension and knowledge enhancement strategies. The positional extension enables the model to adapt to longer texts, enhances adaptability to dataset length, and strengthens contextual understanding capabilities. The knowledge enhancement strategy utilizes external knowledge to generate prompt words, improving the model's ability to model domain-specific terminology and contextual semantics. Ablation experiments further validate the effectiveness of both modules in the joint extraction task.

Although PosEKE-GPT2 has achieved good performance in financial event extraction tasks, there is still room for optimization. In the future, we will explore causal reasoning methods to enable the model to understand causal relationships between events and improve the interpretability of extraction.

Data availability

The Duee-Fin dataset analyzed during the current study is publicly available on the AI Studio platform at [<https://aistudio.baidu.com/datasetdetail/186939>] (<https://aistudio.baidu.com/datasetdetail/186939>). The FewFC dataset analyzed during the current study is publicly available on GitHub at [<https://github.com/TimeBurningFish/FewFC>] (<https://github.com/TimeBurningFish/FewFC>).

Received: 17 July 2025; Accepted: 3 October 2025

Published online: 11 November 2025

References

1. Xiang, Y. et al. Multi-Modal military event extraction based on knowledge fusion. *Computers Mater. Continua.* **77** (1), 97–114. <https://doi.org/10.32604/cmc.2023.040751> (2023).
2. Feng, S. & Zhao, H. Joint Extraction Model for Financial Events Based on Pointer Labeling. *Journal of Changchun University of Technology.* **44**(05), 441–448 (2023). <https://doi.org/10.15923/j.cnki.cn22-1382/t.2023.5.09> (2023).
3. Wang, L. & Information Collection and Triplet Information Extraction from Unstructured Text Data [Master's thesis] Nanjing Audit University, (2023). <https://doi.org/10.27835/d.cnki.gnjsj>
4. Zhu, Y., Cao, Y., Zhong, J. & Zheng, Y. A survey on event extraction technology. *Comput. Sci.* **49** (12), 264–273. <https://doi.org/10.11896/jsjkx.211100226> (2022).
5. Li, F. et al. Event Extraction as Multi-turn Question Answering. in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, 829–838 (2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.73>
6. Van Nguyen, M., Min, B., Dernoncourt, F. & Nguyen, T. in *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies.* 4363–4374 (2022). 4363–4374 (2022). <https://doi.org/10.18653/v1/2022.nacl-main.324>
7. Zhang, Z. & Ji, H. Abstract Meaning Representation Guided Graph Encoding and Decoding for Joint Information Extraction. in *Proc. The 2021 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL-HLT2021)*, 39–49 (2021). <https://doi.org/10.18653/v1/2021.nacl-main.4>

8. Han, R., Ning, Q. & Peng, N. J. a. p. a. Joint event and temporal relation extraction with shared representations and structured prediction. In *proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 434–444 (2019). <https://doi.org/10.18653/v1/D19-1041>
9. Li, X., Cheng, W. & Tang, X. A. Joint extraction method for financial events based on Multi-layer convolutional neural networks. *Comput. Sci.* 65 (24), 89–99. <https://doi.org/10.13266/j.issn.0252-3116.2021> (2021).
10. Li, Y., Geng, C. & Fin -BERT-based method for event extraction in the Chinese financial domain. *Comput. Eng. Appl.* 60 (14), 123–132. <https://doi.org/10.3778/j.issn.1002-8331.2304-0224> (2024).
11. He, L., Li, Z. & Song, J. Prompt Template-guided Method for Document-level Financial Event Extraction. *Data Analysis and Knowledge Discovery*. 1–14 (2024). <https://doi.org/10.11925/infotech.2096-3467.2024.0707>
12. Zou, J., Liu, Y., Qi, Y., Cao, H. & Liu, L. A Generative Approach for Comprehensive Financial Event Extraction at the Document Level. In *Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF '23).Association for Computing Machinery, New York, NY, USA*, 323–330 (2023). <https://doi.org/10.1145/3604237.3626844>
13. Hu, J. & He, W. Document-level Chinese financial event extraction based on RoBERTa and global graph neural network. *J. Chin. Inform. Process.* 37 (02), 107–118. <https://doi.org/10.3969/j.issn.1003-0077.2023.02.011> (2023).
14. Jin, F. Implicit causal relationship extraction in the financial domain based on RACNN and BiLSTM. *Comput. Sci.* 49 (07), 179–186. <https://doi.org/10.11896/jsjkx.210500190> (2022).
15. Cao, H. et al. & Ji D. OneEE: A One-Stage Framework for Fast Overlapping and Nested Event Extraction. in *Proceedings of the 29th International Conference on Computational Linguistics*, 1953–1964(2022). <https://doi.org/10.48550/arXiv.2209.02693>
16. Dai, Z., Tian, S., Yu, L. & Yang, Q. CMCEE: A joint learning framework for cascade decoding with multi-feature fusion and conditional enhancement for overlapping event extraction. *J. ID A.* 28, 717–732. <https://doi.org/10.3233/IDA-230284> (2024).
17. Sheng, J. et al. Joint Learning Framework for the CCKS-2020 Financial Event Extraction *Task Data Intell.*, 3(3), 444–459 https://doi.org/10.1162/dint_a_00098 (2021).
18. Lin, Y., Ji, H., Huang, F. & Wu, L. A Joint Neural Model for Information Extraction with Global Features. in *Proceedings of the 58th annual meeting of the association for computational linguistics*. 7999–8009 (2020). <https://doi.org/10.18653/v1/2020.acl-main.713>
19. Chen, G., Wu, P., Gu, J. & Qian, L. Multi-layer Sequence Labeling-Based Joint Biomedical Event Extraction.in *D. F. Wong, Z. Wei, & M. Yang (Eds.), Natural Language Processing and Chinese Computing: 13th National CCF Conference, NLPCC Hangzhou, China, November 1–3, 2024, Proceedings, Part II*,135–148(2024). <https://doi.org/10.48550/arXiv.2408.05545>
20. Jia, R., Zhang, Z., Jia, Y., Papadopoulou, M. & Roche, C. J. I. A. Improved GPT2 event extraction method based on mixed attention collaborative layer vector. *IEEE ACCESS*. 12 (12), 160074–160082. <https://doi.org/10.1109/ACCESS.2024.3487836> (2024).
21. Hsu, I., Huang, K., Boschee, E., Miller, S. & Natarajan.,Chang, K. A Data-efficient Generation-based Event Extraction Model. in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1890–1908(2022). <https://doi.org/10.18653/v1/2022.naacl-main.138>
22. Duan, J., Liao, X., An, Y., Wang, J. J. B. D. M. & Analytics KeyEE: enhancing low-resource generative event extraction with auxiliary keyword sub-prompt. *Big Data Min. Analytics*. 7 (2), 547–560. <https://doi.org/10.26599/BDMA.2023.9020036> (2024).
23. Shi, G., Su, Y., Ma, Y. & Zhou, M. A. H. Detection and Generation Framework with Separate Encoders for Event Extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3163–3180 (2023). <https://doi.org/10.18653/v1/2023.eacl-main.231>
24. Lu, Y. et al. Unified Structure Generation for Universal Information Extraction.In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5755–5772 (2022).
25. Chen, J., Chen, P. & Wu, X. J. A. S. Generating Chinese event extraction method based on ChatGPT and prompt learning. *Appl. Sci.* 13 (17), 9500. <https://doi.org/10.18653/v1/2022.acl-long.395> (2023).
26. Han, C. et al. Duee-fin: A Large-scale Dataset for Document-level Event Extraction. In *Natural Language Processing and Chinese Computing: 11th CCF International Conference, NLPCC Guilin, China, September 24–25, 2022, Proceedings, Part I*. Springer-Verlag, Berlin, Heidelberg,172–182(2022). <https://doi.org/10.1007/978-3-031-17120-8>
27. Du, R. et al. Comparative study of tools for copy number variation detection using next-generation sequencing data. *Sci. Rep.* 15, 22145. <https://doi.org/10.1038/s41598-025-06527-3> (2025).
28. Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. in *Proceedings of the conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186(2019). <https://doi.org/10.18653/v1/N19-1423>
29. Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.
30. Feng, X., Zhao, X., Feng, X. A. & Soft Parameter Sharing-based method for joint event Extraction.Computer applications research. 40(01), 91–96 (2023). <https://doi.org/10.19734/j.issn.1001-3695.2022.06.0252>
31. Sheng, J. et al. CasEE: A Joint Learning Framework with Cascade Decoding for Overlapping Event Extraction.in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 164–174 (2021). <https://doi.org/10.48550/arXiv.2107.01583>

Author contributions

Tonghui An: Software, Conceptualization, Methodology, Formal analysis, Visualization, Writing-original draft. Zhenling Zhang: Investigation, Data curation, Funding acquisition, Writing-review & editing. Yangli Jia: Supervision, Resources, Writing-review&editing. Ruchao Jia: Validation, Data curation, Software. Maria Papadopoulou: Validation, Software. Christophe Roche: Validation, Software. Tonghui An and Zhenling Zhang are the co-first authors.

Funding

This work is supported by the National Natural Science Foundation of China under Grant (No. 81973695) and the Fund Project of Liaocheng University under Grant (No.318052126).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.J.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025