

Article

Modeling Graph Neural Networks and Dynamic Role Sorting for Argument Extraction in Documents

Qingchuan Zhang ^{1,2}, Hongxi Chen ^{1,2}, Yuanyuan Cai ^{1,2,*} , Wei Dong ^{1,2}  and Peng Liu ³

¹ National Engineering Research Centre for Agri-Product Quality Traceability, Beijing Technology and Business University, Beijing 100048, China

² China Food Flavor and Nutrition Health Innovation Center, Beijing Technology and Business University, Beijing 100048, China

³ Department of Agricultural Food Standardization Institute, China National Institute of Standardization, Beijing 100191, China

* Correspondence: caiyuanyuan@bitbu.edu.cn

Abstract: The existing methods for document-level event extraction mainly face two challenges. The first challenge is effectively capturing event information that spans across sentences. The second challenge is using predefined orders to extract event arguments while disregarding the dynamic adjusting of the order according to the importance of argument roles. To address these issues, we propose a model based on graph neural networks which realizes the semantic interaction among documents, sentences, and entities. Additionally, our model adopts a dynamic argument detection strategy, extracting arguments depending on their number in correspondence with each role. The experimental results confirm the outperformance of our model, which surpasses previous methods by 7% and 1.9% in terms of an F1 score.

Keywords: graph neural networks; attention networks; event detection; argument extraction



Citation: Zhang, Q.; Chen, H.; Cai, Y.; Dong, W.; Liu, P. Modeling Graph Neural Networks and Dynamic Role Sorting for Argument Extraction in Documents. *Appl. Sci.* **2023**, *13*, 9257. <https://doi.org/10.3390/app13169257>

Academic Editors: Fei Xiong, Hongshu Chen, Xuzhen Zhu and Shirui Pan

Received: 11 July 2023

Revised: 4 August 2023

Accepted: 10 August 2023

Published: 15 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Event Extraction (EE) aims to extract structured event details from unstructured text [1], which is regarded as a core research area of Natural Language Processing (NLP). Generally, event details include event types, trigger words, and the various arguments that play different roles within events. The structured representation of events facilitates many downstream tasks, such as recommendation systems [2,3], knowledge graph construction [4,5], and intelligent question–answering systems [6,7]. Particularly in food and cosmetics sentiment monitoring, researchers have identified immense potential in utilizing news and social media data [8] for event extraction in sentiment analysis. Moreover, previous studies have indicated that harnessing multiple data sources [9] or incorporating different features [10] can facilitate the harmful events from extensive text data. Such capabilities offer valuable insights and guidance for sentiment management and decision-making processes.

Existing methods [11–15] of EE are mainly focused on sentence-level event extraction (SEE), which involves extracting events and arguments from individual sentences. For example, ACE-2005 [16], one of the most frequently utilized datasets for evaluation, only annotate event arguments within the scope of a sentence. However, in real-world scenarios, event arguments often span multiple sentences, causing incomplete results when extracting from individual sentences. In reality, a document can contain multiple event records, an entity can play multiple argument roles, and event records may have missing event arguments. Figure 1 illustrates an example from a financial document containing two event records, an Equity Pledge (EP) event and an Equity Underweight (EU) event. In the EP event, the Kangping Company plays the role of the Pledger, which refers to a corporate entity, and it fills this entity as an event argument for the Pledger role in the EP event. At

the same time, the Kangping Company also plays the argument role of Equity Holder in the EU event. However, the event record for the EU event lacks event arguments for the argument role of Later Holding Shares and Average Price.



Figure 1. The example document from ChFinAnn. Due to space constraints, only three sentences (S5, S9, S10) are shown. Color-coded words correspond to event arguments under different event types.

Therefore, Document-level Event Extraction (DEE) is necessary in such cases. The primary objective of the DEE task is to extract events that encompass specific argument roles from an entire document, potentially spanning multiple sentences. There are two main areas for improvement in DEE tasks. One is failing to capture long-distance dependencies between sentences and needing a comprehensive perspective when capturing the correlation between arguments across sentences. Secondly, the extraction order of argument roles is often predefined when performing an asynchronous extraction. This rigid extraction strategy may make it hard to handle the diverse occurrence order and importance of different arguments within each event.

Hence, we propose a novel model of Graph Neural Networks and Dynamic Role Sorting (GNNDRS) for argument extraction in documents. When dealing with scattered arguments across sentences, we establish interactions between documents, sentences, and entities. Specifically, we construct a heterogeneous graph interaction network, including document nodes, sentence nodes, and entity nodes. We model their interactions using five types of edges: sentence-node edges, sentence-entity edges, intra-sentence entity-node edges, cross-sentence entity-node edges, and document-sentence edges. Regarding argument role detection, we dynamically adjust the detection order based on the number of arguments. We prioritize roles with fewer arguments and gradually transition to roles with more arguments. The results demonstrate the effectiveness of our proposed approach compared to the baseline models.

Our contributions can be summarized as follows:

1. We propose the GNNDRS model for addressing the two mentioned challenges. GNNDRS constructs a heterogeneous graph interaction network, which can better capture the connections among the different pieces of information within the document.
2. The GNNDRS model dynamically adjusts the detection order of argument roles, prioritizing the roles with fewer arguments. This approach enhances the accuracy of extracting each event and its associated arguments.
3. We experimentally validate the effectiveness of the GNNDRS model on the datasets, demonstrating its superior performance.

The rest of the paper is organized as follows: Section 2 provides an overview of the related work in event extraction. Section 3 presents our proposed model, which includes entity extraction, the construction of a heterogeneous graph, event type detection, and argument extraction. Section 4 describes the dataset and experimental setup and compares our results with a state-of-the-art method. Finally, in Section 5, we conclude the paper and discuss future directions for research.

2. Related Work

2.1. Sentence-Level Event Extraction (SEE)

The existing approaches for SEE can be broadly categorized into pattern-based and statistical learning methods. Pattern-based methods involve mining contextual argument association features for different event types and designing corresponding pattern-matching templates. However, these methods need better transferability and require extensive manual operations. On the other hand, statistical learning methods focus on extracting appropriate sentence features and selecting suitable classifiers for event type detection and argument extraction. Initially, these methods heavily relied on manually engineered features. With advancements in deep learning techniques, researchers began leveraging the feature representation capabilities of deep neural networks to extract text features automatically. Chen et al. [17] creatively proposed a dynamic multi-pooling convolutional neural network, which divides sentences into segments and applies different convolution pooling operations to each part, augmented by distance encoding for event extraction. Liu et al. [18] simultaneously extracted multiple events from sentences by combining graph structures with attention mechanisms. Zhang et al. [19] introduced imitation learning in event extraction, enhancing the learning effectiveness of infrequent but important events through reward allocation mechanisms.

Furthermore, some researchers have leveraged syntactic features to extract events. Nguyen et al. [20] introduced syntactic dependency relations in the network input and utilized memory matrix encoding to capture relationships between trigger words and arguments. Similarly, Sha et al. [21] incorporated syntactic dependencies into the RNN network. With the advancement of pre-trained language models, researchers have begun exploring the application of BERT in event extraction tasks. Yang et al. [22] enhanced the semantic representation capabilities of their event extraction model by incorporating BERT during the text input stage. In the SEE task, the same word can represent different events in different contexts, requiring accurate classification and attribution based on the context. Therefore, integrating more semantic and contextual features is necessary to improve the extraction of scattered arguments and multiple event information in the future.

2.2. Document-Level Event Extraction (DEE)

In recent years, an increasing number of researchers have started to focus their attention on DEE tasks. Researchers have put forth numerous deep learning-based approaches to address the challenge of scattered arguments in DEE. The mainstream methods can be categorized into two types: trigger-inclusive methods and trigger-free methods. In trigger-inclusive ways, DCFEE [23] introduced an event extraction model that utilizes a neural network sequence labeling model and a parameter-filling strategy. However, the lack of a comprehensive system for coherent reasoning without context fails to effectively address the challenge of scattered sense. Yang et al. [24] and Chen et al. [25] independently performed the joint extraction of events and entities within a document context. Yang et al. utilized an external knowledge base, while Chen et al. employed a Transformer model. Furthermore, the joint extraction approach should consider incorporating multi-task joint learning, effectively leveraging task interactions, and allowing tasks to benefit from the mutual learning process. MSAL [26] focused on identifying all arguments and themes in the text, establishing their relationships to capture a more extensive context. Not long ago, DEEDP [27] combined dependency paths with event role labels to identify events in financial documents. However, in the abovementioned models, the distance between

event arguments and triggers increases within a document, making it more challenging for models to extract event elements [28,29]. Additionally, some events may not have explicit triggers, or acquiring trigger information may be difficult. As a result, some researchers have attempted trigger-free DEE tasks. For instance, Doc2EDAG [30] transformed event tables into entity-based directed acyclic graphs, enabling complex event-filling tasks to convert into multiple entity-based sequential path expansion subtasks, which are often easier to handle. However, this model's process of learning chain structures is computationally complex and not parallelizable, requiring substantial hardware power and time. DEPPN [31] introduced document-level encoders and multi-granularity decoders to generate event-aware representations parallelly with the document, enabling the simultaneous extraction of all events. GIT [32] proposed a Tracker module that continuously tracks extracted event records using a global memory module to capture interdependencies between events. MMR [33] built a multi-round multi-granularity reader designed to read multiple documents simultaneously at different granularities, such as words, sentences, and paragraphs. RAAT [34] utilized the association matrix and entity representation, which allowed it to get the relational dependency information.

While some approaches have attained notable success, existing methods for DEE tasks rely on predefined event role orders for argument detection without considering the correlations between event roles or overlooking the overall information of the document. As a result, there is still a need for improvement in effectively capturing cross-sentence event relationships.

3. Methodology

Before introducing our proposed event extraction model, we would like to provide a more comprehensive explanation of the terms mentioned in this paper. The input document $D = \{S_1 \dots S_i \dots S_n\}$ consists of n sentences, where S_i represents the i -th sentence. The goal of the DEE task is to extract m event records from document D , where each event record consists of an event type j , multiple event arguments a , and the corresponding event roles k , where $a \in E$, $j \in J$, $k \in K$. E represents the candidate entity set, while J and K denote the predefined sets of event types and argument roles, respectively.

3.1. Model Architecture

Our model, as shown in Figure 2, consists of four modules. In the first module, we conduct entity identification using a Conditional Random Field (CRF) [35]. In the second module, we create a heterogeneous graph to model the interactions among the document, sentences, and entities, utilizing GCN to learn document-level representation. In the final two modules, we detect event types and extract their corresponding arguments, employing the Tracker module for continuous global tracking and memorizing candidate entities.

3.2. Entity Extraction

We treat entity recognition as a sequence labeling task. Given a document D composed of multiple sentences $\{S_1 \dots S_i \dots S_n\}$, we use the BERT pre-trained model [36] to encode each sentence, mapping each token in the input sequence to a d -dimensional vector space. We obtain the representation vector E_{emb} for each token by adding two parts of embedding: E_{token} (word embedding) and E_{position} (position embedding). We utilize CRF to better capture the dependencies among labels and achieve a more accurate classification of label sequences. We employ the traditional BIO (Begin, Inside, Other) sequence labeling scheme. To train the model, we use the negative log-likelihood loss function for entity extraction:

$$L_{ner} = - \sum_{s \in D} \log P(y_s | s), \quad (1)$$

where y_s is the gold label sequence for s , and P is the score of the gold label sequence. We use the Viterbi algorithm to decode the label sequence based on the maximum probability for inference.

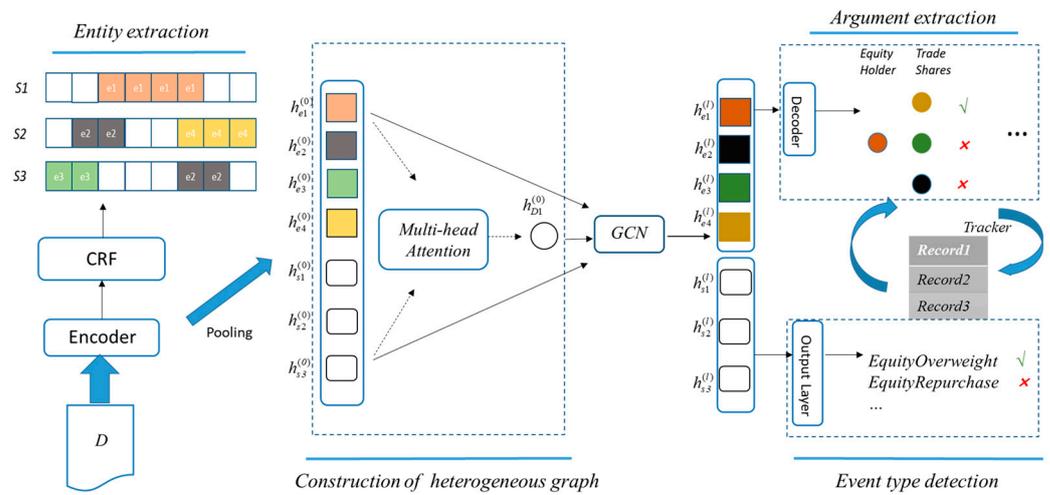


Figure 2. The overview of the proposed GNNDRS model, which can be mainly divide into four modules: entity extraction (3.2), construction of heterogeneous graph (3.3), event type detection (3.4), and argument extraction (3.5).

3.3. Construction of Heterogeneous Graph

We construct a heterogeneous graph G to capture the interactions between sentences and entities, which includes entity, sentence, and document nodes. For a given graph $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. In graph G , the interactions among entities, entities and sentences, and documents and sentences, are all modeled and represented, as shown in Figure 3.

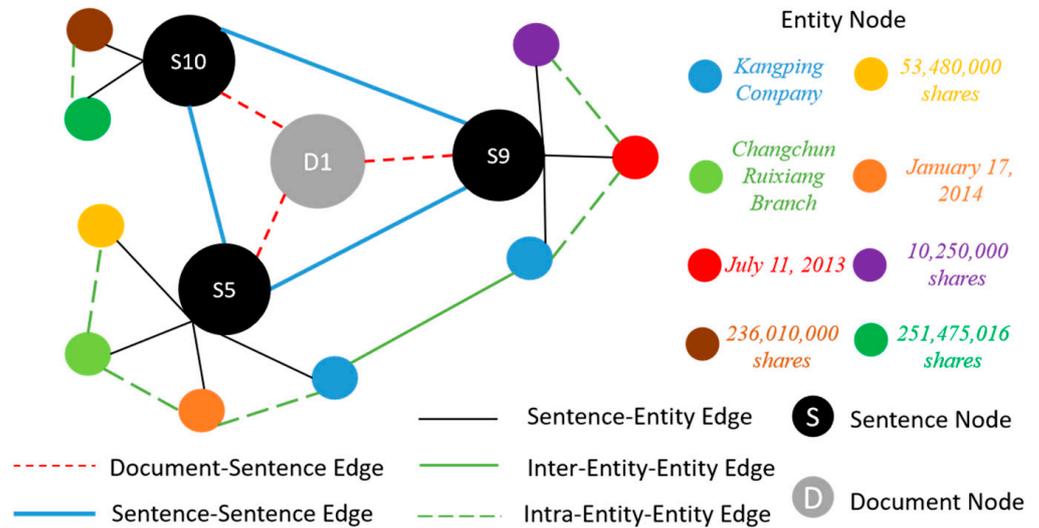


Figure 3. The heterogeneous graph we proposed.

For an entity node e , its initial embedding $h_e^{(0)}$ is the average pooling value of the word vectors within the entity, specifically represented as $h_e^{(0)} = \text{Mean}\{g_j\}_{j \in e}$, where g_j represents the vector for word j , and the mean denotes the average pooling operation. For a sentence node s , its initial embedding $h_s^{(0)}$ is the maximum pooling value of all the word vectors in the sentence plus the position embedding value of the sentence, specifically represented as $h_s^{(0)} = \text{Max}\{g_j\}_{j \in s} + \text{SentPos}(s)$, where Max denotes the maximum pooling operation, and $\text{SentPos}(s)$ represents the position embedding of the sentence.

The document node is jointly represented by the initial embedding vectors of the sentences and entities in the document. The operation involves applying a multi-head attention

mechanism to transform the input sentence embedding vectors and entity embedding vectors into query matrix Q , key matrix K , and value matrix V through a linear mapping layer. Next, the Q , K , and V tensors split into m attention heads, where m represents the number of heads. Matrix multiplication and scaling operations are applied to the query and key matrices of each head, resulting in new attention tensors. Then, the softmax function is utilized to weigh the attention tensor and allocate it to the value matrix, generating the final attention-based output tensor. Finally, the output results of the m attention heads are concatenated and passed through a fully connected layer for mapping, generating the embedded features based on attention computation. The multi-head attention mechanism model can be represented by the following formula:

$$\text{MultiHeadAttention}(x) = FC([\text{Attention}(Q, K, V)]_1, \dots, [\text{Attention}(Q, K, V)]_m), \quad (2)$$

where FC represents the fully connected layer, m denotes the number of attention heads, and Attention denotes the attentional mechanism as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (3)$$

where d_k represents the dimensionality of the query vectors.

After performing multi-head attention calculations on the initial embedding vectors of sentences, a weighted pooling is applied along the document dimension to obtain the overall sentence-level document vector. As for the initial embedding vectors of entities, the vectors of each word are averaged to create a matrix of equally sized word embedding. Then, multi-head attention calculations are performed on this matrix, and the results are weighted and summed along the document dimension to gain word-level document vectors. Finally, by concatenating these vectors, we get the feature representation of the entire document node. The feature vector representation of the document node is as follows:

$$\text{DocEmbedding}(X_{sent}, X_{token}) = [\text{Attention}(X_{sent}), \text{Attention}(X_{token})], \quad (4)$$

where X_{sent} consists of all sentence embedding vectors in a document, and matrix X_{token} consists of all entity embedding vectors in the document. $\text{Attention}()$ refer to the multi-head attention operations, and $[\]$ represents the concatenation between the vectors.

Our model includes five types of edges: Sentence–sentence edges (S–S), sentence–entity edges (S–E), intra-entity edges within the same sentence (E–E intra), inter-entity edges for the same entity across different sentences (E–E inter), and document–sentence edges (doc–s). Sentence nodes are connected through S–S edges to capture long-range dependencies between individual sentences in the document. The S–E edges connect sentences with all entities within them, modeling the context of entities in the sentences. The E–E intra edges connect different entities within the same sentence, indicating that these entities may be related to the same event. The E–E inter edges connect the same entity across all sentences, allowing the tracking of the entity’s occurrences at different positions.

Document–Sentence Edge (doc–s): By connecting the document node with the sentence nodes, we have achieved an interaction between the document and sentences. As a result, the document node can attend to information from all other nodes, facilitating the fusion of textual information from different levels and better modeling long-distance dependency relationships between sentences. Our heterogeneous graph enables the simulation of the interaction between sentences and entities from a global perspective and strengthens the connections between documents and sentences while better capturing event information within the document.

We apply multi-layer graph convolutional networks to model global interactions. For each node i with its feature representation, the node representation at the l -th layer can be computed using the following formula:

$$h_i^l = \text{Relu} \left(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^l h_j^{l-1} + b_r^l \right), \quad (5)$$

where R represents all edge relation types, and N_i^r represents the set of all neighboring nodes connected to node i via relation type r . $c_{i,r}$ is a normalization constant used for normalization. W_r^l denotes the weight matrix corresponding to each edge relation type r , and Relu represents the activation function. We then derive the final hidden state of node i by combining the output features $h_i^{(l)}$ of node i in each layer of GCN along the column direction. The combined column vector is linearly transformed using the learnable weight matrix W_a to learn the final hidden state h_i of node i .

$$h_i = W_a [h_i^{(0)}; h_i^{(1)}; \dots; h_i^{(L)}], \quad (6)$$

where $h_i^{(0)}$ is the initial embedding representation of node i , and L is the number of GCN layers. Finally, we get sentence embedding vectors and entity embedding vectors. In this way, sentences and entities interact in a context-aware manner for representation.

3.4. Event Type Detection

Since documents can express events of different types, and various event arguments within the same event type may be scattered across multiple sentences during event extraction, we formulate event detection as a multi-label classification task. First, we concatenate the sentence embedding vector list of the document to obtain the sentence feature matrix S . Then, we use a *Logsoftmax* classifier to get the probabilities of the occurrence or non-occurrence of each event type, further capturing the contextual information between different sentences:

$$Att = \text{MultiHead}(Q, S, S) \in \mathbb{R}^{d \times J}, \quad (7)$$

$$R = \text{Logsoftmax}(Att^T W_j) \in \mathbb{R}^J, \quad (8)$$

where *MultiHead* refers to the standard multi-head attention mechanism with $Q/K/V$. $Q \in \mathbb{R}^{d \times J}$ and $W_j \in \mathbb{R}^J$ are trainable parameters, where J represents the number of possible event types. Finally, we derive the loss for event type detection using the gold labels:

$$L_{detect} = - \sum_{j=1}^J \prod (\widehat{R}_j = 1) \log P(R_j|D) + \prod (\widehat{R}_j = 0) \log(1 - P(R_j|D)), \quad (9)$$

where $\widehat{R}_j \in \mathbb{R}^J$.

3.5. Argument Extraction

We adopted an ordered expansion tree [32] to decode documents containing multiple event records and extract specific types of event records. After detecting the event types, we performed argument role detection by dynamically adjusting the detection order based on the obtained vector representations and labels of all candidate entities in the text. We first detected roles with fewer arguments and gradually transitioned to argument roles with more arguments. The process of filling event records, as shown in Figure 4, analyses five argument roles. The Company Name, Highest Trading Price, and Lowest Trading Price have only one event argument, while the Repurchased Shares and Closing Date argument roles have two event arguments. When filling in event records, prioritize identifying

argument roles with fewer associated arguments and gradually shift towards roles with more event arguments. Therefore, these two argument roles will be detected last.

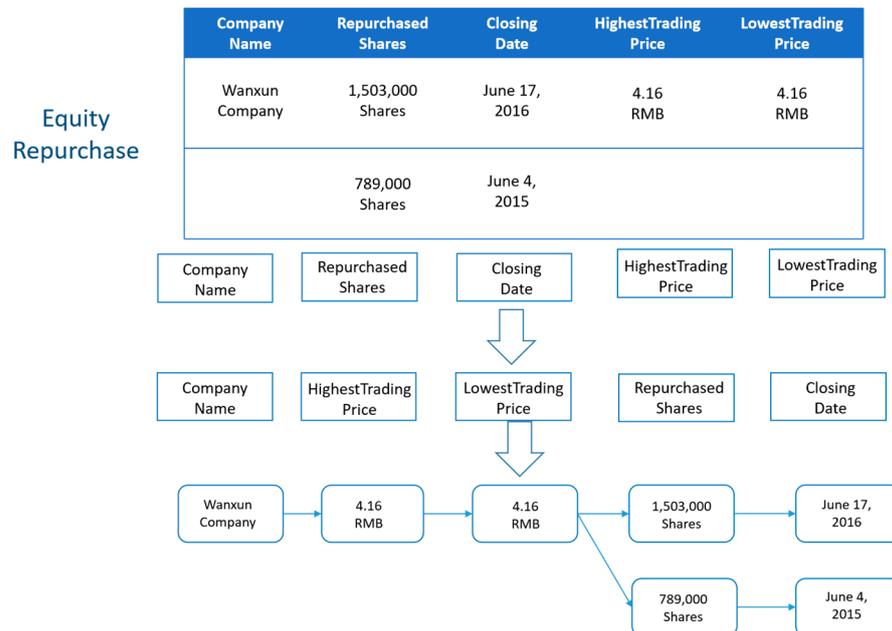


Figure 4. The reordering process of argument extraction.

Specifically, when extracting arguments, we start from a virtual root node and expand based on the reordered sequence of argument roles. We introduce the Tracker module [32], and the path from the root node to the leaf node is an event record. For the i -th record path represented by an entity sequence $U_i = [E_{i1}, E_{i2}, \dots]$, the Tracker utilizes LSTM to encode this vector and adds the event type embedding. The compressed information is then stored in the global memory G_i for sharing across different event types, as shown in Figure 5. It represents the decoding process, wherein records are extracted in a revised sequence following the adjustment of argument role detection. Two event records have already been retrieved from the document. By harnessing the contained information within the real-time Tracker and its globally tracked memory, accurate forecasts can be made regarding the argument roles associated with each event. In this prediction, Entity B is assigned to Role1, and the subsequent argument roles are forecasted accordingly, commencing from this particular child node until a comprehensive event record is extracted.

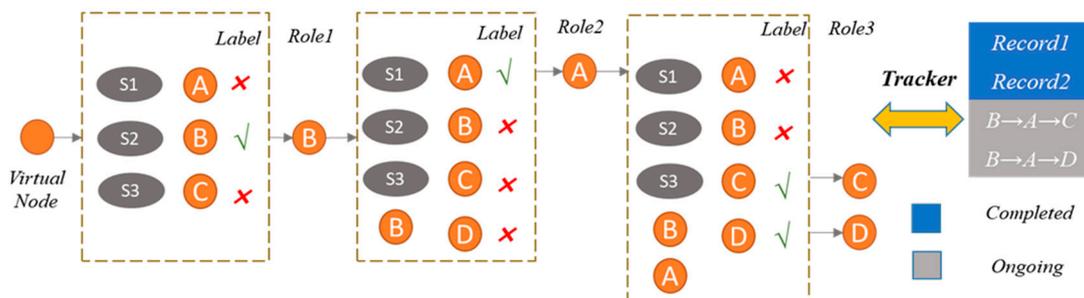


Figure 5. The decoding process. Capitalized letters (A–D) embody all potential entities, whereas S1–S3 denote the distinctive features of the sentence.

During the inference process, we predict the k -th role by incorporating the feature of the argument role into the entity representation.

$$\bar{E} = E + Role_k, \tag{10}$$

where $Role_k$ is the embedding of the k -th role, E is the feature matrix of the entities at the previous time, and \bar{E} refers to the feature matrix at the current time.

Next, the Tracker concatenates the entity feature matrix \bar{E} , sentence feature matrix S , current record path U_i , and global memory G_i . It then utilizes the Transformer to update the feature information. The updated features include specific role information for all candidate entities globally.

$$[\tilde{E}, \tilde{S}, \tilde{U}_i, \tilde{G}] = \text{Transformer}([\bar{E}; S; U_i; G]), \quad (11)$$

L_{record} is represented as:

$$L_{record} = - \sum_{n=1}^N \sum_{i=1} \log P(y^{i,n} | n), \quad (12)$$

where N is the node in the event record tree, and $y^{i,n}$ refers to its golden label sequences. If the i -th entity is the next event argument of node n , then $y^{i,n} = 1$, otherwise $y^{i,n} = 0$.

3.6. Training

To train the above three losses, we employ a multi-task learning approach [37] that integrates the three corresponding loss functions, as shown below:

$$L_{all} = \lambda_1 L_{ner} + \lambda_2 L_{detect} + \lambda_3 L_{record}, \quad (13)$$

where λ_i is present to balance the weights between the three components. L_{ner} is the loss for entity recognition, L_{detect} is the loss for event type detection, and L_{record} is the loss for event record extraction. Then, by selecting an optimizer and setting a reasonable learning rate, the model is trained to optimize the entire system and accomplish the document-level event extraction task. We employ a scheduled sampling strategy to improve the model's performance and stability. In the initial stages, the model only utilizes the actual labels as input and disregards any outputs generated by the model. Subsequently, the model gradually increases its reliance on the model-generated outcomes while decreasing its dependence on genuine labels. During the inference process, we first identify all the entities for a given document, then detect the event types, and finally combine the extracted event arguments to form individual event records. This process ensures that all relevant information is considered when generating event records, leading to enhanced model performance and higher-quality generated event records.

4. Experiments

In this section, we present the experimental results of our model, demonstrating the effectiveness of GNNDRS. Additionally, we conduct ablation studies on each newly proposed architecture in this paper.

4.1. Datasets and Evaluation Metrics

We compare our model with the baselines on the two datasets. The ChFinAnn dataset was constructed by Doc2EDAG [30] and collected from the company announcements published on the Chinese financial portal East Money from 2008 to 2018. The dataset includes five event types: Equity Freeze (EF), Equity Repurchase (ER), Equity Underweight (EU), Equity Overweight (EO), and Equity Pledge (EP). We have processed two datasets and conducted two experiments.

The first dataset focuses on Equity Repurchase (ER) documents. This dataset comprises 1862 training documents, 677 validation documents, and 1138 test documents, covering six argument roles.

The second dataset utilizes a sample training set of 2500 provided by Doc2EDAG [30]. The validation set consists of 3204 documents, and the test set consists of 3204 documents, covering 35 argument roles.

We evaluated the results using commonly used metrics, such as precision (P), recall (R), and the F1 score [30]. The formula for accuracy is as follows:

$$P = \frac{TP}{TP + FP'} \quad (14)$$

where TP represents valid positive samples predicted as positive by the model, and FP' represents false positive samples predicted as positive by the model. The formula for recall is:

$$R = \frac{TP}{TP + FN'} \quad (15)$$

where TP has the same meaning as in the aforementioned formula, and FN' represents false negative samples predicted as unfavorable by the model. The formula for $F1$ score is as follows:

$$F1 = \frac{P * R * 2}{P + R}. \quad (16)$$

Both accuracy and recall can be calculated using a confusion matrix, which separates true positive (TP), true negative (TN), false positive (FP), and false negative (FN) based on actual and predicted classifications. The $F1$ score amalgamates precision and recall, imparting a more comprehensive reflection of the accuracy and correctness of the model.

4.2. Experiments Setting

For each document to be processed, we set the minimum number of sentences to 64 and the maximum length of sentences to 128. We use 8 layers for the encoding module and 4 layers for the decoding module, with 4 attention heads per layer. The hidden layer dimension is set to 768, and the feed-forward layer dimension is set to 1024. The dropout is set to 0.1. During the training process, we set the learning rate to 1×10^{-4} , the number of epochs to 40, $\lambda_1 = 0.05$, and $\lambda_2 = \lambda_3 = 1$. The batch size is set to 8, and we use Adam as the optimizer [30].

4.3. Results and Analysis

We have chosen the current state-of-the-art DEE methods as baselines to enable comprehensive comparisons using the evaluation metrics introduced in Section 4.1. These baselines are listed below:

DEPPN [31] employed a multi-granularity decoder to extract events simultaneously and utilized a matching loss function in training end-to-end models, leading to improved global optimization.

GIT [32] introduced a tracking module to monitor and trace the extracted events, capturing the interdependencies between events.

RAAT [34] proposed an attention-based transformer model that effectively captures the relationships between events in the document by incorporating a mechanism to enhance the representation of event relations.

Tables 1 and 2 present the results of GNNDRS and the baselines on the two datasets, respectively.

Table 1. Precision (P), recall (R), and F1 scores of ER event extraction on the first dataset.

Model	P (%)	R (%)	F1 (%)
DEPPN [31]	70.9	57.5	63.5
GIT [32]	71.3	86.3	78.1
RAAT [34]	70.7	64.5	67.4
GNNDRS (ours)	88.2	82.2	85.1

Table 2. Precision (P), recall (R), and F1 scores on the second dataset.

Model	EF			ER			EU			EO			EP			Average			Total		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
DEPPN [31]	61.7	38.4	47.3	72.1	57.3	63.9	54.0	47.8	50.7	40.3	49.5	44.4	65.3	44.0	52.5	58.7	47.4	51.8	62.9	47.5	54.1
GIT [32]	69.2	33.2	44.9	69.1	61.3	65.0	61.4	52.8	56.7	70.0	55.5	62.0	68.2	59.0	63.3	67.6	52.4	58.4	68.2	57.4	62.3
RAAT [34]	70.0	36.3	47.8	55.0	50.1	52.4	57.7	43.4	49.6	51.5	49.7	50.6	62.3	57.7	59.9	59.3	47.5	52.1	59.7	53.2	56.3
GNNDRS	77.9	37.2	50.4	77.9	65.6	71.2	63.7	49.7	55.8	63.4	57.5	60.3	73.5	56.3	63.8	71.3	53.3	60.3	73.3	57.1	64.2

Overall, our model significantly outperforms the baseline models. In Table 1, GNNDRS showed an impressive 7% increase in the F1 score compared to others. Similarly, in Table 2, GNNDRS achieved a 1.9% improvement in the F1 score. The improvement in the F1 score can be attributed to the implementation of two strategic approaches. First and foremost, we integrate localized and overarching semantic information by including the heterogeneous graph consisting of documents, sentences, and entity nodes. The local semantic data is captured by modeling distinct feature representations for each sentence and entity, and the document nodes provide global semantic insight. By combining them, our model can capture event information at a cross-sentence level. In addition, argument roles with fewer argument quantities may carry more critical semantic information. To ensure the accurate extraction of this crucial information, we adjust the detection order of argument roles, prioritizing these roles. Subsequently, the model progressively detects argument roles with more arguments.

We split the datasets into two subsets to demonstrate the performance of our model in handling different types of documents. The first includes single-event documents (S.), which exclusively contain documents with a single event record. The second consists of multi-event documents (M.), which contain documents with multiple event records. This division allows us to assess how effectively our model captures and processes information from documents with varying complexities and event densities. Tables 3 and 4 compare GNNDRS and the baselines on the two datasets.

Table 3. Comparison of single (S.) and multiple (M.) event extraction on the first dataset.

Model	S. (%)	M. (%)
DEPPN [31]	64.1	58.2
GIT [32]	80.2	61.1
RAAT [34]	69.2	54.5
GNNDRS (ours)	87.7	60.1

Table 4. Comparison between single (S.) and multiple (M.) event extraction on the second dataset.

Model	EF		ER		EU		EO		EP		Average		Total	
	S. (%)	M. (%)												
DEPPN [31]	56.9	37.7	65.3	51.6	55.4	43.6	46.0	41.4	60.2	47.4	56.8	44.3	60.1	46.3
GIT [32]	59.6	45.7	70.0	58.4	61.1	46.2	66.1	54.0	76.8	54.2	66.7	51.7	71.2	53.4
RAAT [34]	56.7	39.3	53.3	46.5	54.4	42.8	55.2	44.0	67.7	55.2	57.5	45.5	59.6	52.4
GNNDRS	62.2	47.4	66.1	53.2	63.9	42.0	66.9	48.6	77.4	56.0	67.3	49.4	70.6	54.0

In Table 3, GNNDRS performs better in the single-event extraction task and significantly outperforms the baseline model in the ER event type, with a 7.5% improvement in the F1 score for single events. In Table 4, GNNDRS outperforms the baseline model in the multiple extraction tasks, achieving a significant 0.6% improvement in terms of the F1 score. The results demonstrate that our proposed strategies effectively overcome the challenges of single-event extraction compared to existing baseline models. Firstly, we achieved global modeling of event information by introducing document nodes. Through

interactions between documents and sentences, the model can capture argument information across sentences and further understand the correlations between different pieces of information within an event. This global modeling approach helps improve the model's understanding of the overall semantics of events and enhances its ability to model relationships between different arguments. In addition, dynamically adapting the order of argument role detection endows the model to delve into contextual information within the document, enabling a more profound comprehension of the informational landscape. It also helps establish dependencies between multiple arguments, enhancing the semantic representation of events. These proposed strategies contribute to the superior performance of GNNDRS compared to the baselines.

4.4. Ablation Study

In this section, to verify the effectiveness of each strategy in GNNDRS, we conducted ablation studies. We specifically excluded the doc-s edges in the heterogeneous graph interaction network, as shown in Section 3.3, and the dynamic detection strategy for argument roles, as presented in Section 3.5. The experimental results are shown in Tables 5 and 6.

Table 5. Performance of GNNDRS on ablation study of the first dataset.

Model	P (%)	R (%)	F1 (%)
GNNDRS	88.2	82.2	85.1
w/o doc-s	72.7	83.6	77.8
w/o sorting	86.9	82.7	84.8

Table 6. Performance of GNNDRS on ablation study of the second dataset.

Model	P (%)	R (%)	F1 (%)
GNNDRS	73.3	57.1	64.2
w/o doc-s	70.5	57.3	63.3
w/o sorting	71.7	56.8	63.4

According to the results in Tables 5 and 6, we can draw the following conclusions: (1) Incorporating edges between documents and sentences leads to an increase of 7.3% and 0.9% in the F1 scores. By incorporating document node features, the heterogeneous graph interaction network adeptly captures the holistic context of the entire document, employing enhanced input features. This integration empowers the model to engage in comprehensive global reasoning while reinforcing semantic dependencies among sentences through interaction with sentence nodes. Ultimately, this cultivates an enriched understanding of the intricate relationship between arguments and events. (2) It is noteworthy that employing our proposed dynamic sorting strategy for argument role detection increases by 0.3% and 0.8% in the F1 scores. In our model, argument roles associated with fewer candidate arguments are prioritized for processing. These relatively fewer arguments are the most crucial components of an event. Consequently, prioritizing their handling enhances the accuracy of identifying and extracting vital information, thus improving the precision of the extraction outcomes. (3) The above two points demonstrate that introducing document nodes into the heterogeneous graph and modeling the interaction between documents and sentences contribute significantly to our model.

4.5. Discussion

According to the experimental results, we observed the following findings. In the task of single-event extraction, our model introduces document nodes to effectively integrate the entire document's information, resulting in a more accurate determination of the document's theme. Our model improved significantly compared to the state-of-the-art models [31,32,34] that focus on contextual information within sentences or relationships

between entities. When the dataset includes more event types, we adopted an asynchronous dynamic adjustment strategy for event argument extraction, prioritizing the processing of easier-to-handle argument roles. This strategy enhances the model's ability to handle complex documents. Therefore, our model improved the F1 scores by 10.1%, 1.9%, and 7.9% in the second dataset compared to the baseline models [31,32,34].

5. Conclusions

We proposed a GNNDRS model which introduces a heterogeneous graph with interaction networks to better capture semantic correlations between contexts. Additionally, our model utilized the dynamic arrangement of argument role detection based on the number of arguments, which further enhances complex argument roles' precision and contextual extraction. Experimental results demonstrated that GNNDRS outperforms the state-of-the-art model, achieving an impressive 7% and 1.9% improvement in the F1 scores. These results validated the effectiveness of our approach. In future work, we plan to evaluate the performance of GNNDRS in various domains and optimize the training speed of the model.

Author Contributions: Conceptualization, Q.Z. and W.D.; methodology, H.C. and Y.C.; software, H.C. and Y.C.; validation, Q.Z., W.D. and Y.C.; formal analysis, H.C.; investigation, Q.Z.; resources, W.D.; data curation, P.L.; writing—original draft preparation, H.C.; writing—review and editing, Q.Z.; visualization, Y.C.; supervision, W.D.; project administration, Q.Z. and Y.C.; funding acquisition, Q.Z., W.D. and Y.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Project of Cultivation for Young Top-notch Talents of Beijing Municipal Institutions (grant No. BPHR202203061), the R&D Program of Beijing Municipal Commission of Education (grant No. KM202010011011), the Humanity and Social Science Youth Foundation of Ministry of Education of China (grant No. 20YJCZH229), and the Innovation Research Special Project of the IFLYTEK for University Intelligent Teaching (grant No. 2022XF055).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: We thank all the anonymous reviewers for their thoughtful comments.

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

References

1. Xiang, W.; Wang, B. A survey of event extraction from text. *IEEE Access* **2019**, *7*, 173111–173137. [[CrossRef](#)]
2. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A Survey on Session-based Recommender Systems. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [[CrossRef](#)]
3. Xiong, R.; Wang, J.; Zhang, N.; Ma, Y. Deep hybrid collaborative filtering for web service recommendation. *Expert Syst. Appl.* **2018**, *110*, 191–205. [[CrossRef](#)]
4. Zhou, Y.; Wang, W.; Yi, D.; Qiao, Z.; Xiao, M. A survey on the construction methods and applications of sci-tech big data knowledge graph. *Sci. Sin. Inf.* **2020**, *50*, 957. [[CrossRef](#)]
5. Wang, X.; Zhang, X.; Liu, F. Knowledge graph construction and application in geosciences: A review. *Comput. Geosci.* **2022**, *161*, 105082.
6. Qiu, B.; Chen, X.; Xu, J.; Sun, Y. A survey on neural machine reading comprehension. *Neurocomputing* **2019**, *338*, 28–41.
7. Qi, L.; Heng, J.; Liang, H. Joint event extraction via structured prediction with global features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–5 August 2013.
8. Jang, K.; Lee, K.; Jang, G.; Jung, S.; Myaeng, S. Food hazard event extraction based on news and social media: A preliminary work. In Proceedings of the 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 18–20 January 2016. [[CrossRef](#)]
9. Ihm, H.; Jang, H.; Lee, K.; Jang, G.; Seo, M.; Han, K.; Myaeng, S. Multi-source Food Hazard Event Extraction for Public Health. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Republic of Korea, 13–16 February 2017. [[CrossRef](#)]
10. Wang, X.; Zhang, Q. Food Safety Event Detection Based on Multi-Feature Fusion. *Symmetry* **2019**, *11*, 1222. [[CrossRef](#)]

11. 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.
12. Wang, X.; Wang, Z.; Han, X.; Liu, Z.; Li, J.; Li, P.; Sun, M.; Zhou, J.; Ren, X. HMEAE: Hierarchical modular event argument extraction. 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.
13. Jin, Y.; Jiang, W.; Yang, Y.; Mu, Y. Zero-Shot Video Event Detection with High-Order Semantic Concept Discovery and Matching. *IEEE Trans. Multimed.* **2022**, *24*, 1896–1908. [[CrossRef](#)]
14. Li, P.; Zhou, G. Joint Argument Inference in Chinese Event Extraction with Argument Consistency and Event Relevance. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 612–622. [[CrossRef](#)]
15. Kanetaki, Z.; Stergiou, C.; Bekas, G.; Troussas, C.; Sgouropoulou, C. Creating a Metamodel for Predicting Learners' Satisfaction by Utilizing an Educational Information System During COVID-19 Pandemic. In Proceedings of the International Conference on Novelties in Intelligent Digital Systems, Athens, Greece, 30 September–1 October 2021; pp. 127–136. [[CrossRef](#)]
16. Doddington, G.R.; Mitchell, A.; Przybocki, M.; Ramshaw, L.; Strassel, S.; Weischedel, R. The automatic content extraction (ACE) program-tasks, data, and evaluation. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal, 26–28 May 2004.
17. 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 (ACL-IJCNLP), Beijing, China, 26–31 July 2015.
18. Liu, X.; Luo, Z.; Huang, H. Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 30 October–1 November 2018.
19. Zhang, T.; Ji, H. Event extraction with generative adversarial imitation learning. *Data Intell.* **2018**, *1*, 99–120. [[CrossRef](#)]
20. Nguyen, T.; 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.
21. 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 Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
22. 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.
23. Yang, H.; Chen, Y.; Liu, K.; Xiao, Y.; Zhao, J. DCFEE: A Document-level Chinese Financial Event Extraction System based on Automatically Labeled Training Data. Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, 15–20 July 2018.
24. Yang, B.; Mitchell, T. Joint Extraction of Events and Entities within a Document Context. 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.
25. Chen, Y.; Chen, T.; Van Durme, B. Joint Modeling of Arguments for Event Understanding. In Proceedings of the First Workshop on Computational Approaches to Discourse of the Association for Computational Linguistics, Online, 10–11 November 2021; pp. 96–101.
26. Ebner, S.; Xia, P.; Culkin, R.; Rawlins, K.; Van Durme, B. Multi-sentence argument linking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
27. Li, H.; Zhao, X.; Yu, L.; Zhao, Y.; Zhang, J. DEEDP: Document-Level Event Extraction Model Incorporating Dependency Paths. *Appl. Sci.* **2023**, *13*, 2846. [[CrossRef](#)]
28. Li, S.; Ji, H.; Han, J. Document-level event argument extraction by conditional generation. 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; pp. 894–908.
29. Zhang, Z.; Kong, X.; Liu, Z.; Ma, X.; Eduard, H. A two-step approach for implicit event argument detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 6–8 July 2020; pp. 7479–7485.
30. Zheng, S.; Cao, W.; Xu, W.; Bian, J. Doc2EDAG: An End-to-End Document-Level Framework for Chinese Financial Event Extraction. 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.
31. Yang, H.; Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; Wang, T. Document-Level Event Extraction via Parallel Prediction Networks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021.
32. Xu, R.; Liu, T.; Li, L.; Chang, B. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021.
33. Yang, H.; Chen, Y.; Liu, K.; Zhao, J.; Zhao, Z.; Sun, W. Multi-Turn and Multi-Granularity Reader for Document-Level Event Extraction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**, *22*, 1–16. [[CrossRef](#)]

34. Liang, Y.; Jiang, Z.; Yin, D.; Ren, B. RAAT: Relation-Augmented Attention Transformer for Relation Modeling in Document-Level Event Extraction. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 10–15 July 2022; pp. 4985–4997.
35. Lafferty, J.; McCallum, A.; Fernando, C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, Williamstown, MA, USA, 28 June–1 July 2001. [[CrossRef](#)]
36. Devlin, J.; Chang, M.; 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, Minneapolis, MN, USA, 2–7 June 2019.
37. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 June 2008. [[CrossRef](#)]

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.