International Journal of Computational Intelligence Systems (2024) 17:296
https://doi.org/10.1007/s44196-024-00697-0

RESEARCH ARTICLE

Fine-Grained Meetup Events Extraction Through Context-Aware Event Argument Positioning and Recognition

Yuan-Hao Lin[1] · Chia-Hui Chang[1] · Hsiu-Min Chuang[2]

Received: 11 June 2024 / Accepted: 4 November 2024

Abstract
Extracting meetup events from social network posts or webpage announcements is the core technology to build event search services on the Web. While event extraction in English achieves good performance in sentence-level evaluation [1], the quality of auto-labeled training data via distant supervision is not good enough for word-level event extraction due to long event titles [2]. Additionally, meetup event titles are more complex and diverse than trigger-word-based event extraction. Therefore, the performance of event title extraction is usually worse than that of traditional named entity recognition (NER). In this paper, we propose a context-aware meetup event extraction (CAMEE) framework that incorporates a sentence-level event argument positioning model to locate event fields (i.e., title, venue, dates, etc.) within a message and then perform word-level event title, venue, and date extraction. Experimental results show that adding sentence-level event argument positioning as a filtering step improves the word-level event field extraction performance from 0.726 to 0.743 macro-F1, outperforming large language models like GPT-4-turbo (with 0.549 F1) and SOTA NER model SoftLexicon (with 0.733 F1). Furthermore, when evaluating the main event extraction task, the proposed model achieves 0.784 macro-F1.

Keywords Meetup event extraction · Context-aware event extraction · Event argument positioning · Event argument recognition

1 Introduction

Looking for local events to attend is a common need for most people when traveling or exploring a city. Therefore, extracting meetup events from the Web is crucial to building a meetup event search service. A meetup event is featured by event title, hosting organization, date, location, target participants, and registration fee, etc. This study aims to design an efficient model for extracting four key elements of meetup events: title, location, start date, and end date. As shown in Fig. 1, we highlighted title in blue, start and end dates in orange and purple, and venue in green within the Facebook fan page post.

Since a webpage or a post on social networks may mention multiple meetup events, we define a target event to be an event that includes a title and at least one other piece of event information, such as the event location, start date, or end date. Only events that meet this criterion will be labeled as target events for extraction. In Fig. 1, the underlined text shows another entity of event name, but it is not a target event that we aim to extract. When multiple target events co-locate in one page, associating recognized locations and dates with event title is challenging. Therefore, we focus on the extraction of target events.

Existing meetup event extraction methods primarily focus on event extraction at the webpage level. For example, Foley et al. [3] uses distant supervision to automatically label ClueWeb12 corpus based on 211K unique events leveraged from http://schema.org/event records. They adopt a bottom-up approach to recognize three basic event elements, including "When," "Where," and "What" through a LIBLINEAR sentence classifier based on 13 features. However, the

✉ Chia-Hui Chang
chiahui@g.ncu.edu.tw

Yuan-Hao Lin
luff543@gmail.com

Hsiu-Min Chuang
showmin@cycu.edu.tw

[1] Department of Computer Science and Information Engineering, National Central University, No. 300, Zhongda Rd., Zhongli Dist., Taoyuan City 320317, Taiwan

[2] Information and Computer Engineering, Chung Yuan Christian University, No. 200, Zhongbei Rd., Zhongli Dist., Taoyuan 320314, Taiwan

Published online: 26 November 2024

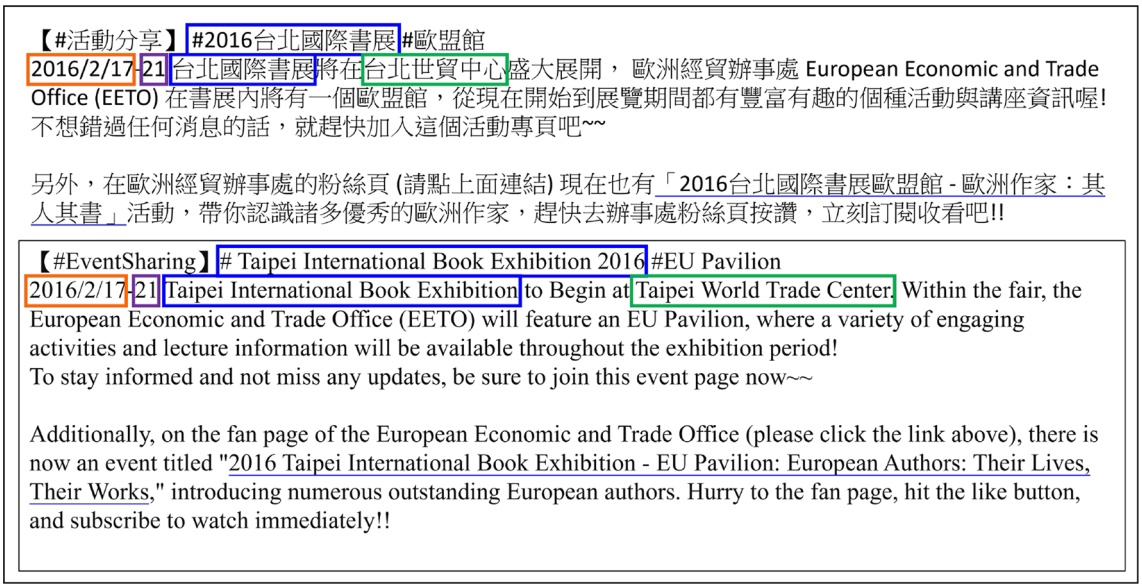


Fig. 1 An event post with the title boxed in blue, start/end dates in orange/purple, venue in green, and non-target entity underlined (Top) In Chinese (Bottom) In English

precision of event field extraction for these three elements are only 0.36, 0.32, and 0.66 for “What,” “When,” and “Where", respectively.

To improve the performance of event extraction, Wang et al. [1] proposed an event extraction pipeline divided into two phases. The first phase consists of three modules which predict if a web page contains any event information, decide whether a page contains a single event or multiple events, and extract the event title by classifying a text node of more than 20 words. The event date and location are extracted through a joint extraction method. The second phase includes multiple event extraction through repeated patterns, event consolidation, and wrapper induction, which are designed to use the raw event extractions as input to generate events with high confidence.

On the other hand, Lin et al. [2] targeted word-level event extraction and proposed the task of extracting local events from Facebook Fanpage posts. They modeled event extraction as a sequence labeling problem similar to named entity recognition (NER) through distant supervision with 109 K seed events. However, word-level event extraction is hindered because the automatic tagging of event titles cannot be precise for long titles. Although model-based auto-labeling is exploited to improve the quality of the training data, the performance of word-level event title extraction only achieves 0.573 F1 for event title recognition using BERT-based sequence labeling.

The performance gap between the extraction of sentence-level events (coarse-grained) in English and the extraction of word-level events (fine-grained) in Chinese can be attributed to several reasons: First, precise matching of lengthy titles frequently results in numerous false negatives, while approximate matching may yield excessive false positives. This issue is magnified in word-level event extraction. Second, non-English meetup event sources for distant supervision are typically scarce and lack popularity on social networks, rendering distant supervision unfeasible.

Despite the availability of manually annotated data, extracting events at the word-level remains a formidable challenge. Even the most advanced large language models, such as GPT-4, achieve only an F1 score of 0.549. This challenge arises because not all entities mentioned in the text pertain to our task of extracting information about meetup events ; that is, our targets are event arguments described in the text, rather than all mentioned entities. Moreover, word-level extraction of meetup events suffers from the data sparsity problem. This implies that the percentage of sentences labeled with event titles, locations, and start/end dates is incredibly low.

Furthermore, unlike traditional event extraction, in which event arguments are usually mentioned in the same sentence, meetup event arguments (especially event titles) are often scattered in multiple sentences. They may be mixed with descriptions of other meetup events. Therefore, performance is hampered when contextual information is disregarded. All these issues pose substantial hurdles, making meetup event extraction an arduous endeavor.

To address these challenges, we propose a context-aware meetup-event extraction (CAMEE) framework. Our framework is designed to handle the specific difficulties presented by meetup events, as event titles are often lengthy and full of details, with relevant information frequently scattered across multiple sentences, making key information extraction more

difficult. The CAMEE framework operates in two phases: event positioning and event argument extraction. In the first phase, we utilize a sentence-level model to identify whether a given sentence contains relevant event information, such as the event's title, date, or location. This reduces the impact of irrelevant content and helps focus on key information. In the second phase, we apply a word-level model to jointly identify the boundaries and types of event arguments using the Joint Boundary and Type Recognition (JBTR) approach. This approach accurately determines the boundaries of event arguments and classifies them into their respective types. Specifically, JBTR addresses challenges of boundary ambiguity and type overlap by simultaneously learning boundaries and types, enabling precise event argument identification and differentiation of overlapping types. Through this dual-layered approach, CAMEE effectively filters out irrelevant entities, enhancing the accuracy and completeness of event extraction. The contribution of the work can be summarized in three parts.

- We introduce a context-aware event argument positioning model based on the BERT architecture for locating event arguments at the sentence level. Experimental results show that the proposed model, Context-Aware Multi-Label Classifier (CAMLC), outperforms other sentence-level detection models (e.g., BERT-CLS [4], BERT-Att-BiLSTM-RC [4, 5]), and H-BERT-MLP [6]) and achieves the highest performance, with a Macro-F1 score of 0.780, for the event argument positioning task at the sentence level.
- For word-level event field recognition, we adopted a multitask learning model by decoupling the event field extraction problem into two subproblems: boundary identification and argument type categorization. The proposed Joint Boundary and Type Recognition (JBTR) model outperforms BERT-QA [7] and ERNIE [8] by more than 14 to 28% and is comparable to the lexically enhanced BERT-based SoftLexicon model [9] (0.726 vs. 0.733 macro-F1).
- Through the CAMEE framework, the two-stage model achieves an overall performance of 0.743 macro-F1, outperforming BERT-based SoftLexicon [9] by 1% macro-F1 (0.743 vs. 0.733). In terms of the top 1 extraction, i.e., P/R/F@1, the two-stage method achieves 0.784 macro-F1.

The following section compares related work on meetup events and traditional event extraction. Section 3 introduces the architecture of the system to build meetup events. We present the performance of different event extraction models in Sect. 4. Finally, Sect. 5 concludes the article and suggests future work.

2 Related Work

Finding local events in a new city can be a search service for users to explore the city. Like GIS and geo-social search, event search or recommendation is also a location-based service that aims to meet users' information needs due to mobility. An event search system may recommend a kid-friendly event at a children's amusement park to users. The motivation is similar to the research on contextual suggestions by Dean-Hall et al. [10], which aims to provide users with a better search experience. Constructing an event database can provide a comprehensive event search service that depicts what people do in a city and, to some extent, reflects the community culture. Therefore, Google researchers [1, 3] have focused on extracting events from semistructured web pages or plain text messages posted on social networks.

2.1 Meetup-Event Extraction from the Web

Wang et al. [1] used HTML tag paths as landmarks for event extraction rules. They handled single and multiple event pages differently. For pages with multiple events, they exploited repeated patterns, labeling event titles from schema.org as positive examples to train a neural network for title extraction. For single-event pages from the same domain, they clustered similar HTML structures to generate XPath template rules for field extraction. For single-event pages without similar templates, they grouped extracted titles/arguments across sources, removing duplicates to consolidate the best event title, date, and location representations from each cluster. This multi-source consolidation improved extraction quality even when individual page extractions were imprecise.

Unlike above, Foley et al. [3] do not expect the target data to align perfectly with any structure. Since web pages often contain multiple areas, such as navigation links, banners, main content, advertisements, and footers, they model the task as a scoring problem following a bottom-up approach (from field-level, region-level, to document-level) and greedily grouped extracted fields into disjoint event records with the assumption that predicted events should not overlap. The field set scoring function included the field-level scoring for each field (i.e., text span) and the joint scoring of field occurrences in a region. However, Foley et al. use simple SVM methods to get the score for each event field. Thus, the performance does not achieve the precision needed for real-world applications.

Essentially, Foley et al. [3] and Wang et al. [1] handled the problem based on text nodes parsed from HTML DOM trees, i.e., the output is a coarse-grained text string output, which may cover other irrelevant information. Foley et al. conducted a multiclass classification on each text field divided by HTML

tags. Since a text node may contain more than one field and other information, using multiclass classification does not seem reasonable. Thus, the performance for event title extraction is not high (36% precision). To improve performance, Wang et al. ignore text nodes with less than 20 words to reduce the number of negative examples for event title classification and improve performance to 84% precision. As for event date and address extraction, they trained an independent binary classifier using pattern-based approaches. Since there could be multiple titles, venues, and start and end dates in a webpage, Foley et al. [3] ranked predicted nodes by scores and greedily output the highest-scoring event on a page.

On the contrary, Lin et al. [2] focused on word-level event extraction from posts on social media networks to obtain a more fine-grained output. They used seed-based distant supervision to prepare two training corpora (based on Google search snippets and posts in the Facebook Fan group) using Facebook Event and Citytalk. Because exact matches could be used to annotate only a small percentage of posts relating to events, they introduced longest common subsequence (LCS)-based matching and the filtering of core words to overcome noise in approximate matching. They achieved 0.565 F1 in extracting the title of the events. They used this model to label 604K sentences from fan groups on Facebook, and their F1 score improved from 0.565 to 0.573 F1.

2.2 Traditional (Closed Domain) Event Extraction from Plain Texts

Unlike the extraction of meetup-type events that users can add to their calendars, traditional event extraction dates back to the message understanding conference (MUC) series, which aimed to develop systems capable of recognizing named entities (e.g., people, organizations, and locations) and extracting events (such as military conflicts, changes in corporate management, and joint ventures). Following the MUC series, the Automatic Content Extraction (ACE) project [11][1] was launched in the early 2000s as part of DARPA's Translingual Information Detection, Extraction, and Summarization (TIDES) program.

According to the task definition of event extraction provided by ACE 2005, event mentions are triggered by single verbs or nouns and are associated with other entities referred to as arguments, describing changes in event states, including who, what, when, where, and how [12, 13]. In the ACE 2005 event corpus, eight event types and 33 event subtypes are predefined. Typical closed-domain event extraction methods involve four subtasks: trigger word identification, event type classification, argument identification, and argument role classification, which can be executed sequentially or simultaneously.

While classical approaches such as Dynamic Multi-pooling Convolutional Neural Network (DMCNN) [14] and Joint Recurrent Neural Network (JRNN) [15] apply event type classification or argument role labeling for each word, others [16] utilize sequence labeling models to identify both triggers and argument roles.

One benefit of sequence labeling is that one can utilize lexical information to recognize trigger words as well as argument roles to improve model performance. For example, the BERT-based SoftLexicon [9] introduces a simple yet effective method incorporating lexical information into character representations. Compatible with BERT, it further enhances performance by effectively leveraging lexicons containing proper nouns and common entity names, significantly boosting model precision and recall.

Meanwhile, large pre-trained language models like BERT [4], GPT [17, 18], and T5 [19] have revolutionized model building in deep learning. BERT-like models, founded upon Auto-encoding Language Models, exhibit outstanding performance in singular-task Natural Language Understanding (NLU) tasks [6, 7, 9]. Of particular note is their excellent performance on the ACE 2005 event retrieval task, surpassing the category T5 model proposed in 2019 [20].

Furthermore, traditional event extraction methods typically focus on identifying event arguments from explicit mentions in the text, relying heavily on clearly stated event triggers and arguments. However, in real-world scenarios, event arguments are not always explicitly mentioned, making implicit event argument extraction increasingly important. In the field of implicit event argument extraction, prior work mainly focused on capturing direct relationships between arguments and event triggers. However, the FEAE framework [21] introduces a novel reasoning method based on event frames, using related arguments as clues to guide the extraction of implicit arguments. This approach integrates a curriculum knowledge distillation strategy and achieves state-of-the-art performance on the RAMS dataset. Similarly, the AREA framework [22] proposes a model for implicit event argument extraction using argument-relationship reasoning. By incorporating knowledge distillation and curriculum learning, it achieves strong performance on the RAMS and Wikievents datasets.

2.3 Open Domain Event Extraction

With the popularity of social networks, the task of event extraction has shifted to user-generated content on social networks or blogs. The new event extraction goal is extracting open-domain events (such as epidemiology spreads, natural disaster damage reports, riots, etc.) for information understanding. For example, Ritter et al. propose TwiCal [23] to

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace.

extract, summarize, and classify important events. Events in TwiCal are represented by 4-tuples, including named entities, event phrases, calendar dates, and event types. The authors apply a named entity tagger trained on in-domain Twitter data, train a CRF-based sequence model to extract event phrases using 1000 manually annotated tweets, and use TempEx [24] to parse explicit calendar-referenced time expressions.

In contrast to supervised training for event phrase extraction, Chen et al. [25] proposed distant supervision to automatically generate labeled data for large-scale event extraction by jointly using world knowledge (Freebase) and linguistic knowledge (FrameNet). The labeled data preparation involves four steps: (i) key argument detection via Freebase; (ii) trigger word detection by labeling sentences in Wikipedia; (iii) trigger word filtering and expansion by FrameNet; and (iv) automatic labeled data generation by the Soft Distant Supervision. The experimental result showed that the model trained with large-scale auto-labeled data is competitive with models trained with human-annotated data from the ACE corpus [14].

However, earlier Open Information Extraction (OIE) systems often faced difficulties in accuracy due to their inability to model dependencies among extractions. In contrast, sequence generation-based methods have shown improved performance but are limited by autoregressive strategies that may reduce efficiency. Therefore, OIE methods have gradually evolved from traditional sequence labeling approaches to detection-based models. For example, Wei et al. [26] utilized parallelism for tuple extraction and highlighted the challenges related to label assignment in these models, proposing a novel IoU-aware Optimal Transport method to enhance label assignment in OIE.

In addition to the development of open domain event extraction techniques, the problem of event detection, description and linking from heterogeneous social media has become a new research topic. For example, Abebe et al. [27] proposed a framework SEDDaL, which uses a Metadata Representation Space Model (MRSM) with temporal, spatial, and semantic dimensions to uniformly represent diverse social media events. The framework then applies similarity evaluation, event detection, and relationship identification to create a knowledge graph of interconnected events, demonstrating a new application of open domain event extraction.

3 Proposed Methods

The focus of this study is to design an effective model to extract meetup events. Specifically, we want to extract the title, location, and start/end date of the main event mentioned in posts on fan groups on Facebook, as shown in Fig. 1. Similar to relation extraction tasks, which involve NER and relation classification between entities, event extraction methods typically follow a pipeline framework first to identify trigger words and arguments followed by event type and argument role classification, as shown in Fig. 2a.

However, extracting meetup events poses unique challenges compared to traditional event extraction: (1) meetup event titles are often verbose and difficult to identify via single trigger words. The vocabulary must accommodate community jargon (e.g., Dining In The Dark) and event-related details (such as dates, person names, etc.) beyond simple triggers. (2) Event titles, dates, and locations for meetups are scattered across multiple sentences within posts while existing methods operate at the single-sentence level. (3) Meetup extraction focuses specifically on extracting event arguments like titles, times, and venues—a restrained scope compared to all entity mentions captured in named entity recognition. Therefore, event-irrelevant entities need to be excluded to avoid incorrect recognition of the event relation.

Based on the analysis, we propose the context-aware meetup event extraction (CAMEE) framework (see Fig. 2b) that consists of a sentence-level model for locating event arguments (i.e. predicting whether each sentence contains the title, location, start date, and end date of an event based on four binary outputs) and a multi-task argument recognition model (that extracts the boundary of each event field and predicts its type).

3.1 Sentence-Level Model for Event Argument Positioning

As mentioned, traditional event extraction models are limited to processing single-sentence inputs, while existing meetup event extraction models only extract sentence-level event fields. To address this limitation, we propose the event argument positioning approach at the sentence level to capture contextual information from adjacent sentences to predict if any event arguments are present in a sentence. Meanwhile, because a sentence may contain the event title, event venue, and start or end date simultaneously, we frame the problem as a multi-label classification task.

The sentence-level model for locating event arguments is shown in Fig. 3. Let $L = \{Title, Venue, StartDate, EndDate\}$ be the set of argument roles for event extraction and $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\mathbf{s}|})$ denotes a message, where each $\mathbf{s}_i$ is a sentence in $\mathbf{s}$, represented by $\mathbf{s}_i = (w_{i1}, \ldots, w_{ik}), k = |\mathbf{s}_i|$. Each sentence is associated with a subset of L, represented by a vector $\mathbf{y}_i = [y_{i1}, y_{i2}, y_{i3}, y_{i4}]$ where $y_{ij} = 1$ if the sentence $\mathbf{s}_i$ contains an event argument L_j and 0 otherwise.

To represent a sentence, we adopt a pre-trained BERT-based encoding module to produce sentence-level representations. A special token [CLS] is added in front of every sentence, i.e., $w_{i0} = [CLS]$ for all i. The output of BERT's bidirectional transformer on the [CLS] token,

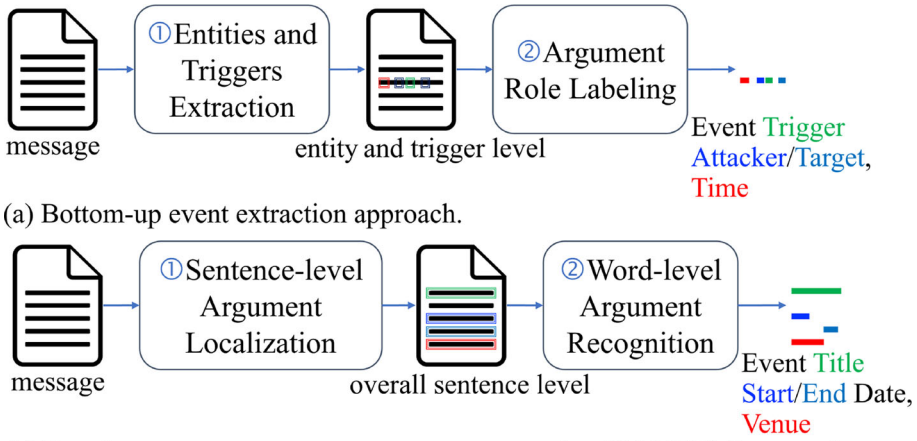


Fig. 2 Comparison of bottom-up traditional event extraction approach and top-down CAMEE framework

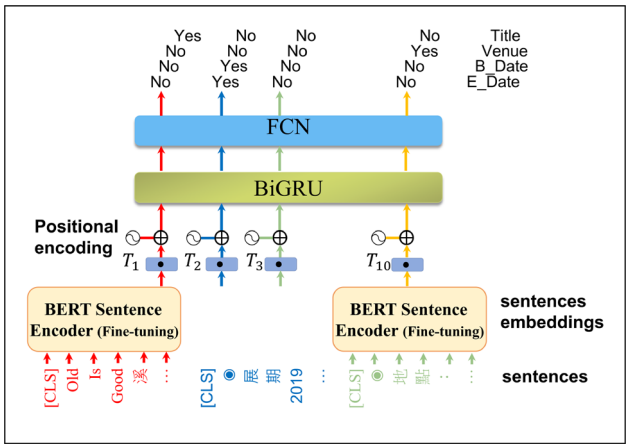


Fig. 3 Sentence-level event argument positioning by context-aware multi-label classifier (CAMLC)

$T_i = BERT(\mathbf{s}_i) \in R^d$, is then used as the sentence representation for sentence $\mathbf{s}_i$.

To highlight the position of the sentence, we incorporate positional embeddings employing a sinusoidal curve encoding method [28] as BERT.

$$T_i = T_i + PositionEmbedding(i) \tag{1}$$

Next, all sentence representations $\mathbf{T} = [T_1, T_2, \ldots, T_{|\mathbf{s}|}]$ will pass through a Gated Recurrent Unit (GRU) layer to capture contextual information from adjacent sentences. We stack two GRU networks to get information from backward and forward states simultaneously. Formally,

$$\begin{aligned} \overrightarrow{z_i} &= \overrightarrow{GRU}(\overrightarrow{z_{i-1}}, T_i) \\ \overleftarrow{z_i} &= \overleftarrow{GRU}(\overleftarrow{z_{i+1}}, T_i) \\ z_i &= \overrightarrow{z_i} \oplus \overleftarrow{z_i} \end{aligned} \tag{2}$$

where $\overrightarrow{z_i}$ and $\overleftarrow{z_i}$ (with d_r hidden neurons) denote the hidden states of the forward and backward GRU at the i-th time step. Let θ_r denote the parameters of Bi-GRU, we define the output of Bi-GRU function $G_r(\mathbf{T}; \theta_r) = [z_1, z_2, \ldots, z_{|\mathbf{s}|}] \in R^{|\mathbf{s}| \times 2d_r}$. Finally, two fully connected layers with parameters W_l and $\mathbf{b_l}$, $l = 1, 2$ are added to predict whether each sentence $\mathbf{s}_i$ is associated with label $\mathbf{y}_i$ by

$$\hat{\mathbf{y}} = \sigma(G_r(\mathbf{T}; \theta_r) * W_1 + \mathbf{b_1}) * W_2 + \mathbf{b_2}) \tag{3}$$

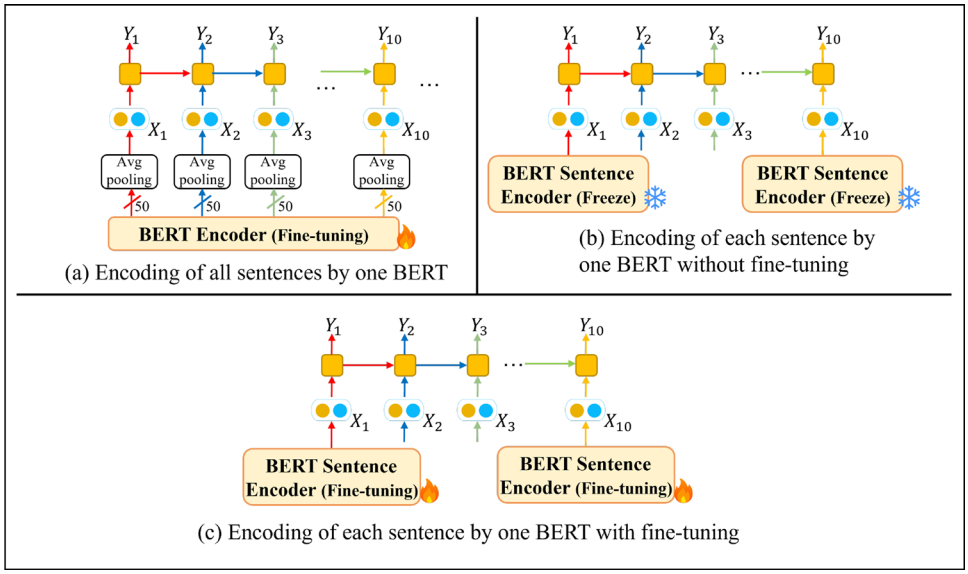
where $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|s|}\} \in R^{|\mathbf{s}|*4}$ and σ denotes the sigmoid function. For layer one: $W_1 \in R^{2d_r \times d_f}$, and $\mathbf{b_1} \in R^{d_f}$. For layer two: $W_2 \in R^{d_f \times 4}$, and $\mathbf{b_2} \in R^4$. We use the cross-entropy to evaluate the loss for each message s as our loss function:

$$L(\mathbf{s}) = -\frac{1}{4 * |\mathbf{s}|} \sum_{i=1}^{|\mathbf{s}|} \sum_{j=1}^{4} y_{ij} log(\hat{y}_{ij}) + (1 - y_{ij}) log(1 - \hat{y}_{ij}) \tag{4}$$

Implementations

There are two possible implementations of BERT sentence-level models. One can encode all sentences by one BERT and fine-tune the model during training (see Fig. 4a) [6] by utilizing average pooling with 50 window size to extract sentence vectors for each sentence. However, simply using average pooling may lead to the loss of some crucial semantic information within sentences and may not accurately encode each sentence independently, thus limiting the model's grasp of the internal details of each sentence. Another approach is to encode each sentence by one BERT model and fine-tune the model with the pre-trained BERT frozen during training (see Fig. 4b) [29]. In other words, this approach employs BERT as feature extraction such that the pre-trained BERT does not participate in the downstream tasks training (no gradient feedback).

On the contrary, our implementation (see Fig. 4c) records both the forward propagation and vector values of sentences



(a) Encoding of all sentences by one BERT

(b) Encoding of each sentence by one BERT without fine-tuning

(c) Encoding of each sentence by one BERT with fine-tuning

Fig. 4 Three possible implementations of sentence-level sequential output: **a** encoding of all sentences by one BERT [6] **b** encoding of each sentence by one BERT without fine-tuning [29] **c** encoding of each sentence by one BERT with fine-tuning

each time BERT is used as a sentence encoder. We use TensorFlow's while_loop function to retain tensor values across iterations during the forward pass, ensuring that gradient updates can be calculated accurately during backpropagation. This enables the calculation of gradient changes during the backward process, allowing BERT to utilize the gradient information from each sentence to update its weights and better capture semantic relationships across sentences. By harnessing this gradient information, BERT's weights are adjusted to enhance further its ability to capture semantic information. Specifically, we adopt a sequence-to-sequence GRU structure, which has the advantage that the loss includes the output of the GRU at each time step. This means that the error gradient of the model will flow backward from the output of each time step, allowing multiple fine-tuning of BERT.

3.2 Word-Level Argument Recognition Model

Following the previous section, we will perform argument boundary recognition if the event argument positioning indicates a sentence is marked as "yes" for any event field. Otherwise, we will skip it to avoid false positives. Note that argument recognition is similar to named entity recognition. However, this task can involve nested structures and mutually exclusive categories. Nested structures mean some labels may be contained within others in the sequence labeling task. Mutually exclusive categories mean each input can only belong to one class—the classes are mutually exclusive. For instance, date can be categorized as either "start date" or "end date", but it cannot simultaneously belong to both.

To handle both nested structures and mutually exclusive categories effectively, we propose combining conditional random fields (CRF) and softmax. First, we use CRF to identify the boundaries of entities or event arguments, separating the sequence into chunks. Then, we use softmax to vote on each chunk and classify it as a particular entity or event argument type.

Specifically, for each sentence $\mathbf{x} = (w_1, \ldots, w_{|x|})$ with $|x|$ words, the output is divided into two parts, the boundary of the event arguments $\mathbf{y^B} = \{y_1^B, y_2^B, \ldots, y_{|x|}^B\}$ and the type of the event arguments $\mathbf{y^T} = \{y_1^T, y_2^T, \ldots, y_{|x|}^T\}$. The former is implemented with L_B={B, I, E, S, O} tags denoting $Begin$, $Inside$, End, $Single$, and $Outside$, i.e., $y_i^B \in L_B$; while the latter is formulated as a five-label classification problem, i.e., $y_i^T \in \{0, 1\}^5$ ($|\mathbf{y_i^T}| = \mathbf{1}$), where each dimension denotes one of the argument role $L_T = \{Title, Venue, StartDate, EndDate\}$ or None.

The loss function is divided into two parts (Eq. 5): the boundary loss $L_{boundary}$ calculates the loss value using negative log-likelihood (Eq. 6) and the type loss uses cross-entropy to estimate the loss value for each token label and predicted probability (Eq. 7):

$$L(\mathbf{x}) = L_{boundary}(\mathbf{x}) + L_{type}(\mathbf{x}) \tag{5}$$

$$L_{boundary}(\mathbf{x}) = -logp(\mathbf{y^B}|\mathbf{x}) \tag{6}$$

$$L_{type}(\mathbf{x}) = -\sum_{i=1}^{|\mathbf{x}|}\sum_{j=1}^{5} y_{ij}^T \log p(y_{ij}^T|\mathbf{x}) \tag{7}$$

where the $p(\mathbf{y^B}|\mathbf{x})$ is calculated by a conditional random field (CRF) layer for boundary identification with a transition matrix T and an omission matrix O (see Eqs. 8 and 9), and $p(\mathbf{y^T}|\mathbf{x})$ is calculated by a fully connected layer with softmax output.

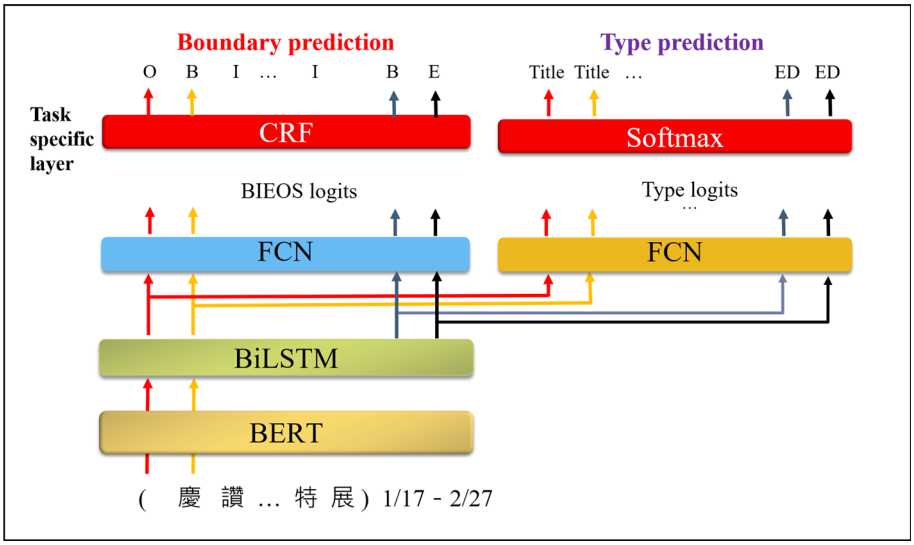


Fig. 5 Word-level event argument recognition by joint boundary and type recognition (JBTR)

$$p(\mathbf{y}^{\mathbf{B}}|\mathbf{x}) = \frac{e^{s(\mathbf{x},\mathbf{y}^{\mathbf{B}})}}{\sum_{\mathbf{y}} e^{s(\mathbf{x},\mathbf{y})}}, \tag{8}$$

$$s(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} T_{y_{i-1},y_i} + \sum_{i=1}^{|\mathbf{x}|} O_{w_i,y_i} \tag{9}$$

By combining the output of these two parts, we can extract the event title, event location, and start and end date in the message. More specifically, given the multi-task predictions, we first follow the boundary prediction to decide the string to be output and use the majority vote to determine the event field based on the type prediction of each token in the boundary. For example, as illustrated in Fig. 6, consider the sentence "1/13 – 31 書畫義賣聯展" (which means "Painting and Calligraphy Charity Exhibition from January 13th to 31st"). The CRF boundary identification (the second row in Fig. 6) results in the entire sentence as a single boundary. However, softmax classification assigns the first three tokens "1/13" to the start date, the following three tokens " – " to other, "31" to the end date, and the remaining 6 tokens " 書畫義賣聯展" to the event title. Using majority voting, the entire sentence is ultimately classified as the event title.

4 Experiments

In this section, we first detail the preparation of the training dataset (Sect. 4.1) used in our experiments and then show why distant supervision with auto-labeled data could not achieve manual-labeled performance in data Sect. 4.2. Following this, we report the performance on (sentence-level) event argument positioning in Sect. 4.3 and word-level event argument extraction in Sect. 4.4. Finally, we show that the two-stage CAMEE framework outperforms other state-of-the-art methods for event extraction in Sect. 4.5 and report the performance of main event extraction in Sect. 4.7.

4.1 Preparation of Training Data and Evaluation Metric

We used both automatic and manual labeling to prepare the training data. For automatic labeling, we used events from Accupass,[2] a ticket-selling website, as seeds to automatically label event fields in the corresponding posts regarding events to prepare the training data. However, since only 11% and 32% of the posts in the dataset mentioned event title and location, respectively, in order to avoid insufficient automatic labeling, we also manually labeled 2065 posts[3] (1274 for training, 791 for validation) from the Facebook fanpages. The number of sentences containing event titles/venues/dates and the number of respective event fields are shown in Table 1.

To evaluate the proposed method's performance in extracting the event field, we used the same test data as Lin et al. [2], which consists of 1300 posts (33,991 sentences) from event announcements on fanpages on Facebook. Note that half of these posts contain only one event title, while the other half might contain more than one event title. Therefore, the number of sentences that contain event titles (2010) and event venues (1563) is higher than 1300. It is worth noting that most event posts had a main event, which is consistent with the nature of event posts on Facebook fanpages.

For sentence-level event argument positioning, we calculated the ratio of the number of correct predicted sentences to the total number of predicted sentences ($\sum \hat{y}_i^c$) and the

[2] https://www.accupass.com/.

[3] The dataset and code are available at https://github.com/luff543/CA-BERT-MLP/.

Token	1	/	13		-		31	書	畫	義	賣	聯	展
Boundary	B	I	I	I	I	I	I	I	I	I	I	I	E
Type	SD	SD	SD	O	O	O	ED	Title	Title	Title	Title	Title	Title
Combine output	B-SD	I-SD	I-SD	I-O	I-O	I-O	I-ED	I-Title	I-Title	I-Title	I-Title	I-Title	E-Title
After voting	B-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	I-Title	E-Title

Fig. 6 Example of combining boundary prediction and type prediction

Table 1 Datasets from which event fields were extracted

Label method	Auto-label		Manual label					
			Training		Development		Testing	
# Event posts	26,314		1,274		791		1,300	
# Sentences	1,324,020		62,474		24,067		33,991	
Event field	# sent	# ent	# sent	# ent	# sent	# ent	# sent	# ent
Title	9090	8943	1117	1096	1087	1071	1985	2010
Venue	14,680	15,614	811	930	812	1005	1270	1563
Start date	14,625	15,123	892	874	832	832	1347	1348
End date	2561	2706	563	525	471	470	874	874

labeled sentences ($\sum y_i^c$) to obtain the precision and recall, respectively:

$$P^c = \frac{\sum_{i=1}^{N} y_i^c \hat{y}_i{}^c}{\sum_{i=1}^{N} \hat{y}_i^c}, \quad R^c = \frac{\sum_{i=1}^{N} y_i^c \hat{y}_i^c}{\sum_{i=1}^{N} y_i^c},$$
$$F^c = \frac{2 \times P^c \times R^c}{P^c + R^c} \tag{10}$$

where N is number of sentences, y_i^c and $\hat{y}_i^c \in \{0, 1\}$ are the true label and the predicted label of sentence i, respectively. We then calculated the harmonic mean of P^c and R^c as the F1 measure. To access the overall performance, we used macro-averaging to summarize its performance on C event fields.

$$P_{macro} = \frac{1}{C}\sum_{c=1}^{C} P^c, \quad R_{macro} = \frac{1}{C}\sum_{c=1}^{C} R^c,$$
$$F_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}} \tag{11}$$

For word-level evaluation of event argument extraction, we used partial precision/recall for each case in the extracted fields E and true answer fields A. Formally, for each extracted field e that overlapped with the true answer field a, we defined the partial scores $P\_score(e, a)$ and $R\_score(e, a)$, as shown in Eq. (12). The precision and recall were then averaged over all the extracted fields E and the true answer fields A, respectively, as shown in Eq. (13).

$$P\_score(e, a) = \frac{P(e \cap a)}{|e|}, \quad R\_score(e, a) = \frac{P(e \cap a)}{|a|} \tag{12}$$

$$\text{Precision} = \frac{\sum_{e \in E} P\_score(e, a)}{|E|},$$
$$\text{Recall} = \frac{\sum_{a \in A} R\_score(e, a)}{|A|} \tag{13}$$

For the following experiments, we train the models using Chinese $BERT_{BASE}$[4] [4] for subsequent experiments. We use the Adam weight optimizer with an initial learning rate of 1e– 5. For the sentence-level Model, the maximum learning epochs are set to 60, while for the word-level Model, the maximum learning epochs are set to 30. All experimental results were averaged over three trials. We utilized the validation set to select the epoch with the best Macro F1 score and report the models' performance.

4.2 Comparison of Auto-Labeled and Manually Labeled Data

To see why we chose manual-labeled data instead of auto-labeled data, we evaluated the performance of the proposed method CAMLC and JBTR in terms of sentence level and word level, respectively. For sentence-level event argument positioning, we utilize a positive-to-negative ratio of 1:9 for both auto-labeled and manual-labeled data. As shown in Table 2, the performance of the sentence-level model trained on manual-labeled data surpasses that of models trained on auto-labeled data (0.780 vs. 0.493 macro F1).

As for word-level event extraction, we employed a positive-to-negative ratio of 1:5 and 1:2 for manual-labeled

[4] Google Research BERT: https://github.com/google-research/bert.

Table 2 Sentence-level event argument positioning by CAMLC model (Fig. 3)

Training data	AutoLabel			Manual label		
	P	R	F	P	R	F
Title	0.580	0.526	0.551	0.706	0.727	**0.715**
Venue	0.534	0.525	0.526	0.751	0.810	**0.779**
Start date	0.796	0.325	0.461	0.743	0.881	**0.806**
End date	0.812	0.173	0.285	0.791	0.849	**0.819**
Macro	0.681	0.387	0.493	0.748	0.817	**0.780**

Bold symbols indicate the best performance among all compared models

Table 3 Word-level event argument extraction using JTBR model in Fig. 5

Training data	AutoLabel			Manual label		
	P	R	F	P	R	F
Title	0.736	0.272	0.396	0.607	0.608	**0.608**
Venue	0.578	0.262	0.360	0.678	0.791	**0.730**
Start date	0.718	0.416	0.527	0.655	0.904	**0.760**
End date	0.823	0.193	0.311	0.736	0.874	**0.799**
Macro	0.714	0.286	0.408	0.669	0.795	**0.726**

Bold symbols indicate the best performance among all compared models

and auto-labeled data, respectively to obtain the best performance. One explanation is that auto-labeled data might contain more noise than manual-labeled data. Note that the performance gap in sentence-level and word-level evaluation urges us to boost JBTR by CAMLC. As shown in Table 3, since it is more difficult for auto-labeling to achieve accurate labeling at the word level (especially by "exact match"), the performance of the model trained on manual-labeled data is much higher than that trained on auto-labeled data (0.726 vs 0.408 macro F1).

4.3 Effect of Event Arguments Positioning

We compared the proposed CAMLC model with three sentence-level event argument positioning methods: BERT-CLS [4], BERT-Att-BiLSTM-RC [5], and H-BERT-MLP [6]. The BERT-CLS and BERT-Att-BiLSTM-RC accepted a single-sentence input, while the H-BERT-MLP and CAMLC accepted multiple-sentence inputs. BERT-CLS is based on the fine-tuned BERT and uses the [CLS] token to represent the entire sentence [4]. BERT-Att-BiLSTM-RC uses the BiLSTM and an attention mechanism to model sentence-level relation extraction [4, 5]. H-BERT-MLP [6] receives a multiple-sentence input separated for context-aware representation and uses a shared dense layer to predict the output. Since we model the problem as a multi-label classification problem, we equip the output layer with four sigmoid functions and train the model with the same loss function as Eq. 4.

Table 4 shows that of the two single-sentence input models considered, BERT-Att-BiLSTM-RC (0.764 F1 score) outperformed BERT-CLS (0.674 F1 score), although the latter had the highest recall (0.879). Of the models that accepted multi-sentence input, the proposed CAMLC model had a higher precision than H-BERT-MLP by almost 10% (0.748 vs 0.655) and a slightly lower recall (0.817 vs. 0.836). Therefore, it also had a better macro-F1 score (0.780 vs. 0.735).

We conducted an ablation study on the proposed CAMLC model. As shown in Table 5, the performance drops to 0.575, 0.759, and 0.754 without fine-tuning, downsampling mechanism, and position encoding, respectively. Thus, it is vital that the BERT model is fine-tuned during training.

4.4 Performance of Word-Level Event Field Extraction

Next, we compared the proposed word-level event argument recognition model with *BERT-QA*, *ERNIE*, *BERT-based SoftLexicon*, and *BERT-BiLSTM-CRF*. *BERT-QA* [7] formulates event field extraction as a question-answering (QA) task by asking questions, such as event triggers and argument roles. *ERNIE* [8],[5] a Baidu-released pretraining model learns prior knowledge about phrases and entities during training through knowledge masking. The *BERT-based SoftLexicon* [9] is a state-of-the-art (SOTA) entity recognition model that integrates character embedding (Char + bichar), CTB (Chinese Treebank) 6.0 embedding, BERT embedding, and lexical information to label sequences of sentences in Chinese. This model enhances character representations by incorporating lexical information, which helps to capture richer semantic and syntactic features. Lexical information is derived from external lexical resources and is used to augment the character embeddings, providing more context and improving the model's ability to recognize entities accurately. Finally, the *BERT-BiLSTM-CRF* model serves as the baseline for sequence-labeling tasks. This model leverages contextual information captured by BERT encoder [4] and sequential dependencies modeled by Bi-LSTM [30], enhancing its ability to accurately assign labels to sequential data, making it a robust starting point for various sequence-labeling problems.

As shown in Table 6, the proposed JBTR model with Chinese base BERT outperforms (in terms of macro F1 score) BERT-QA, ERNIE and BERT-BiLSTM-CRF and is comparable (0.726) to the BERT-based SoftLexicon model (0.733). However, our proposed model is much simpler than BERT-based SoftLexicon and does not rely on additional lexical information.

To reason about the low performance of BERT-QA, we compare the ACE dataset with our meetup event corpus as

[5] https://github.com/PaddlePaddle/ERNIE/tree/ernie-kit-open-v1.0/applications/tasks/sequence_labeling.

Table 4 Comparison of sentence-level models for event argument positioning

Sentence-level model		BERT-CLS [4]	BERT-Att-Bi-LSTM-RC [4, 5]	H-BERT-MLP [6]	CAMLC
Title	P	0.505	0.683	0.591	**0.706**
	R	**0.855**	0.717	0.728	0.727
	F1	0.635	0.699	0.652	**0.715**
Venue	P	0.550	0.669	0.653	**0.751**
	R	**0.876**	0.840	0.817	0.810
	F1	0.675	0.744	0.725	**0.779**
Start date	P	0.566	0.702	0.652	**0.743**
	R	0.876	0.899	**0.930**	0.881
	F1	0.688	0.788	0.766	**0.806**
End date	P	0.564	0.771	0.723	**0.791**
	R	**0.911**	0.875	0.871	0.849
	F1	0.697	**0.820**	0.790	0.819
Macro	P	0.546	0.706	0.655	**0.748**
	R	**0.879**	0.833	0.836	0.817
	F1	0.674	0.764	0.735	**0.780**

Bold symbols indicate the best performance among all compared models

Table 5 Sentence-level model ablation study

Methods	Macro precision	Macro recall	Macro F1
CAMLC (this paper)	0.748	0.817	0.780
w/o fine-tuning	0.505	0.666	0.575
w/o downsampling	0.795	0.727	0.759
w/o position encoding	0.723	0.788	0.754

Table 6 Comparison of word-level event argument extraction models

Word-level model		Title	Venue	Start date	End date	Macro
BERT-QA [7]	P	0.375	0.470	0.512	0.407	0.441
	R	0.354	0.332	0.768	0.338	0.448
	F1	0.363	0.388	0.614	0.368	0.444
ERNIE [8]	P	**0.793**	**0.840**	**0.834**	**0.866**	**0.833**
	R	0.197	0.558	0.533	0.504	0.448
	F1	0.315	0.671	0.650	0.637	0.583
BERT-based SoftLexicon [9]	P	0.742	0.785	0.717	0.756	0.750
	R	0.503	0.712	0.823	0.834	0.718
	F1	0.586	**0.747**	**0.765**	0.793	**0.733**
BERT-BiLSTM-CRF [4, 30]	P	0.599	0.652	0.633	0.703	0.647
	R	0.599	**0.792**	**0.910**	**0.887**	**0.797**
	F1	0.598	0.714	0.746	0.784	0.714
BERT-based JBTR	P	0.607	0.678	0.655	0.736	0.669
	R	**0.608**	0.791	0.904	0.874	0.794
	F1	**0.608**	0.730	0.760	**0.799**	0.726

Bold symbols indicate the best performance among all compared models

shown in Table 7. The ratio of OOV trigger words in the ACE 2005 dataset (58/212) was much lower than that in the FB event dataset (1888/1889). The ratio of sentences containing mentions of events in ACE 2005 (18% = 3136/17,172) was much higher than that of the FB event dataset (1.79% = 1117/62,474). Finally, event triggers in the ACE 2005 dataset were defined as single words, while those in the FB event dataset were defined as the event titles, usually phases.

In summary, although BERT-QA achieved the SOTA performance in the ACE 2005 Trigger Identification + Classification task, its performance drops significantly for the meetup event extraction, which requires deeper semantic or contextual understanding. As for ERNIE, it had the highest precision on all tasks of event field extraction. This verifies its success in minimizing bias for false-negative instances through knowledge-masking. However, its low recall rate also caused the model to lose generalizability. We believe that the bias caused by knowledge masking during the pretraining of ERNIE led to this phenomenon because the title and venue to be identified were OOV.

4.5 Evaluation of the Two-Stage CAMEE Framework

Tables 4 and 6 manifest our intuition that coarse-grain event extraction typically performs better than fine-grain event extraction. Therefore, we exploited the sentence-level model to filter out sentences that did not contain event arguments and boost the performance of the word-level model as shown in Fig. 2b. Table 8 compares the performance of the four sentence-level event argument positioning models combined with the proposed word-level JBTR model. The macro F1 of the word-level JBTR model is improved by the sentence-level event argument positioning model CAMLC to 0.743, outperforming the BERT-Att-BiLSTM-RC model (0.739) and the H-BERT-MLP model (0.738). In general, the performance of word-level event extraction (0.726) improves by 1.2 to 1.7%, except for BERT-CLS. Although BERT-CLS has the best recall (0.879) in predicting event argument positions at the sentence level, the inferior precision (0.546) worsens the performance of the JBTR model from 0.726 to 0.724 F1.

In addition to Table 8, we also examined the effect of the proposed CAMEE framework on the BERT-based Softlexicon and Roberta-large-based JBTR model. As shown in Table 9, when filtered through the sentence-level CAMLC model, both word-level event argument extraction models exhibited a 6.8 to 7.8% improvement in precision but a 5.6 to 10% decrease in recall performance. Since the BERT-based Softlexicon model itself showed significantly lower recall (0.718) compared to JBTR based on BERT and JBTR based on Roberta-large (0.795), its final Macro-F1 score is only 0.704, which is lower than the Macro-F1 scores of 0.743 and 0.750 for the BERT-based and Roberta-large-based JBTR models respectively.

4.6 Evaluation of Large Language Models on Meetup Event Extraction Tasks

To see how large language models perform in event meetup extraction tasks, we assessed GPT−3.5-turbo and GPT-4-turbo to extract meetup events on the test data set using zero-shot prompt as shown in Appendix A. Table 10 presents the precision (P), recall (R), and F1 scores of each model in the meetup event extraction task. Specifically, GPT−3.5-turbo and GPT-4-turbo had macro-F1 scores of 0.369 and 0.549, respectively.

These results may stem from several factors. Firstly, BERT is a model architecture specifically designed for natural language understanding and representation tasks. After fine-tuning, it can better adapt to the specific requirements of the task. Secondly, during the fine-tuning process, the BERT model can effectively capture subtle nuances in the meetup event extraction tasks, thereby enhancing its performance on this particular task. While GPT models possess powerful generative capabilities and versatility, their precision and recall in specific tasks may be compromised, particularly in scenarios requiring high-precision extraction.

4.7 Performance of Main Event Extraction

Finally, a message may contain more than one meetup event, but only the main event is our extraction target. Therefore, we also evaluated the precision and recall of the top-ranked extraction in each message to assess its message-level performance. We defined the precision at k ($P@k$) and the recall at k ($R@k$) to evaluate the performance of the proposed method in terms of extracting the main event from each post. Suppose that a system returns a ranked list $R_l = (r_1^l, \ldots, r_n^l)$ for the event field l, the $P@k$ is defined by matching it with the golden answer, i.e., human-annotated labels, A_l for event field l ($l \in L$) as shown in Eq. (14):

$$\mathrm{P@k} = \frac{1}{k}\sum_{j=1}^{k} b_j^l, \quad \mathrm{R@k} = \frac{1}{|A_l|}\sum_{j=1}^{k} b_j^l \tag{14}$$

where b_j^l is one if $r_j^l \in A_l$ and is zero otherwise.

Table 11 shows the overall performance and the P@1/R@1/F1@1 values of the proposed method for the two scenarios when the sentence-level model was and was not used. Filtering of the sentence-level extraction model improved its overall precision and F1 score from 0.669/0.726 to 0.747/0.743 at the cost of the recall value. The values of P@1/R@1/F1@1 of the proposed method in terms of extracting the main event improved from 0.736/0.673/0.703 to 0.837/0.738/0.784, an increase of 10.1%/6.5%/8.1%.

Table 7 Comparison of the ACE 2005 corpus and the meetup event corpus

Description	ACE 2005 EN			FB meetup event corpus		
	Train	Test	# OOV	Train	Test	# OOV
# distinct triggers	918	212	58	929	1889	1888
# distinct stemmed triggers†	577	163	28	817	1541	1532
Sentences	17,172	832	NA	62,474	33,991	NA
Positive instance	3136	284	NA	1117	1985	NA

†With JaroWinkler similarity greater than 0.8

Table 8 Performance boosting of JBTR word-level event argument extraction by four sentence-level event argument positioning models (except for BERT-CLS)

Word-level evaluation		BERT-CLS [4]	BERT-Att-BiLSTM-RC [4, 5]	H-BERT-MLP [6]	CAMLC
Title	P	0.630	0.682	0.696	**0.705**
	R	**0.587**	0.543	0.523	0.532
	F1	**0.608**	0.605	0.597	0.606
Venue	P	0.719	0.741	0.746	**0.755**
	R	**0.781**	0.761	0.758	0.756
	F1	0.748	0.751	0.752	**0.755**
Start date	P	0.657	0.705	0.701	**0.735**
	R	0.827	0.865	**0.878**	0.853
	F1	0.732	0.777	0.779	**0.790**
End date	P	0.772	0.780	0.777	**0.794**
	R	0.834	**0.837**	0.829	0.814
	F1	0.801	**0.807**	0.802	0.804
Macro	P	0.695	0.727	0.730	**0.747**
	R	**0.757**	0.752	0.747	0.739
	F1	0.724	0.739	0.738	**0.743**

Bold symbols indicate the best performance among all compared models

Table 9 Performance boosting of three word-level event argument recognition models by the CAMLC model

Model/framework	Single stage			Two-stage		
	P	R	F1	P	R	F1
BERT-based softlexicon	0.750	0.718	0.733	0.818	0.618	0.704
BERT-based JBTR	0.669	0.795	0.726	0.747	0.739	0.743
Roberta-large JBTR	0.683	0.795	0.734	0.754	0.746	**0.750**

Bold symbols indicate the best performance among all compared models

Table 10 Evaluating the performance of GPT large language models on meetup event extraction tasks

Model		Title	Venue	Start date	End date	Macro
GPT−3.5-turbo	P	0.527	0.296	0.359	0.435	0.404
	R	0.244	0.251	0.474	0.385	0.339
	F1	0.333	0.272	0.409	0.408	0.369
GPT-4-turbo	P	0.667	0.424	0.498	0.476	0.516
	R	0.400	0.481	0.749	0.713	0.586
	F1	0.500	0.451	0.598	0.571	0.549

Table 11 Performance of the main event extraction

Model/task		Title	Venue	Start date	End date	Macro
Single-stage	P@1	0.685	0.747	0.740	0.772	0.736
	R@1	0.505	0.697	0.719	0.769	0.673
	F1@1	0.581	0.721	0.729	0.771	0.703
Two-stage	P@1	0.815	0.819	0.866	0.848	0.837
	R@1	0.569	0.745	0.822	0.817	0.738
	F1@1	**0.670**	**0.780**	**0.843**	**0.832**	**0.784**

Bold symbols indicate the best performance among all compared models

5 Conclusion and Future Work

In this paper, we investigate the problem of extracting fine-grained Chinese meetup events from posts on social networks. We found that automatic labeling of training data by preexisting events was of rather limited quality, which can adversely affect performance. Therefore, manual labeling of data remains a crucial requirement. Additionally, compared to traditional named entity recognition or event extraction tasks, the frequency of meetup event sentences is relatively low. Consequently, traditional sequence labeling methods that operate solely on single sentences exhibit suboptimal performance in this scenario. To solve this problem, we propose a two-stage pipeline strategy to improve the word-level argument recognition task through sentence-level event argument positioning CAMLC model and a multi-task argument recognition JBTR model. Experimental results show that our proposed event extraction method can improve the extraction performance from 0.726 to 0.743 macro-F1 for all events and from 0.703 to 0.784 macro-F1 for main events. A demo website can be accessed at https://eventgo.widm.csie.ncu.edu.tw/.

A Prompts Used for GPT-3.5-turbo and GPT-4-turbo

GPT-3.5-turbo Prompt

標記輸入文章提到的活動的活動名稱、活動地點、活動開始時間和活動結束時間分別使用以下tag <activity_name_LxKyMA7eXL>、<activity_loc_LxKyMA7eXL>、<activity_date_start_LxKyMA7eXL>、<activity_date_end_LxKyMA7eXL>標註原始文本。
但請注意，輸出請不要添加額外非輸入資訊相關的內容。
Please mark the name of the activity, the location of the activity, the start time of the activity, and the end time of the activity mentioned in the input text using the following tags: <activity_name_LxKyMA7eXL>, <activity_loc_LxKyMA7eXL>, <activity_date_start_LxKyMA7eXL>, <activity_date_end_LxKyMA7eXL>.
However, please note that do not add any additional information unrelated to the input text.

GPT-4-turbo Prompt

標記輸入文章提到的活動的活動名稱、活動地點、活動開始時間和活動結束時間分別使用以下tag <activity_name_LxKyMA7eXL>、<activity_loc_LxKyMA7eXL>、<activity_date_start_LxKyMA7eXL>、<activity_date_end_LxKyMA7eXL>標註原始文本。
但請注意，輸出請不要添加額外非輸入資訊相關的內容，輸出也要包含輸入的原始文章。
Please mark the name of the activity, the location of the activity, the start time of the activity, and the end time of the activity mentioned in the input text using the following tags: <activity_name_LxKyMA7eXL>, <activity_loc_LxKyMA7eXL>, <activity_date_start_LxKyMA7eXL>, <activity_date_end_LxKyMA7eXL>.
However, please note that do not add any additional information unrelated to the input text, and the output should also include the original input text.

Author Contributions Y.-H. Lin conducted methodology, model training, programming, and writing. C.-H. Chang conducted review and editing and project administration. H.-M. Chuang conducted writing and reviewed. All authors have read and agreed to submit the manuscript.

Funding This work was supported by the National Science and Technology Council, Taiwan, under grant NSTC109-2221-E-008-060-MY3.

Data Availibility The Meetup Events Extraction dataset and codes used in this study have been publicly shared to promote transparency and reproducibility of the research findings. The data is available at the following URL: https://github.com/luff543/CA-BERT-MLP/.

Declarations

Conflict of interest All authors have no Conflict of interest to disclose regarding the publication of this study.

Ethical and Informed Consent for Data Used Our study adheres to ethical principles and has obtained appropriate informed consent for using the data involved. All data collection and processing procedures comply with international and local laws and regulations, as well as relevant research ethical standards. All personal data involved in the research process has undergone anonymization to safeguard the privacy and data security of participants. We are committed to ensuring transparency and legality in data usage, while also respecting the rights and interests of participants.

References

1. Wang, Q., Kanagal, B., Garg, V., Sivakumar, D.: Constructing a comprehensive events database from the web. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. CIKM '19, pp. 229–238. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3357986
2. Lin, Y.-H., Chang, C.-H., Chuang, H.-M.: Eventgo! mining events through semi-supervised event title recognition and pattern-based venue/date coupling. J. Inf. Sci. Eng. **39**(3), 655–670 (2023). https://doi.org/10.6688/JISE.20230339(2).0014
3. Foley, J., Bendersky, M., Josifovski, V.: Learning to extract local events from the web. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '15, pp. 423–432. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2766462.2767739
4. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423
5. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 207–212. Association for Computational Linguistics, Berlin, Germany (2016). https://doi.org/10.18653/v1/P16-2034. https://www.aclweb.org/anthology/P16-2034
6. Lei, J., Zhang, Q., Wang, J., Luo, H.: Bert based hierarchical sequence classification for context-aware microblog sentiment analysis. In: International Conference on Neural Information Processing, pp. 376–386. Springer, Sydney, NSW, Australia (2019). Springer
7. 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), pp. 671–683. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.49. https://aclanthology.org/2020.emnlp-main.49
8. Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y., Liu, W., Wu, Z., Gong, W., Liang, J., Shang, Z., Sun, P., Liu, W., Xuan, O., Yu, D., Tian, H., Wu, H., Wang, H.: Ernie 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. In: arXiv Preprint arXiv:2107.02137, vol. abs/2107.02137. arXiv, "Online" (2021). https://api.semanticscholar.org/CorpusID:235731579
9. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5951–5960. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.528. https://aclanthology.org/2020.acl-main.528
10. Dean-Hall, A., Clarke, C.L., Simone, N., Kamps, J., Thomas, P., Voorhees, E.: Overview of the TREC 2013 contextual suggestion track. In: Voorhees, E. (ed.) Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013. NIST Special Publication, vol. 500-302. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA (2013). http://trec.nist.gov/pubs/trec22/papers/CONTEXT.OVERVIEW.pdf
11. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: The automatic content extraction (ACE) program – tasks, data, and evaluation. In: Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R. (eds.) Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). European Language Resources Association (ELRA), Lisbon, Portugal (2004). http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf
12. Xiang, W., Wang, B.: A survey of event extraction from text. IEEE Access **7**, 173111–173137 (2019). https://doi.org/10.1109/ACCESS.2019.2956831
13. Li, Q., Li, J., Sheng, J., Cui, S., Wu, J., Hei, Y., Peng, H., Guo, S., Wang, L., Beheshti, A., Yu, P.S.: A survey on deep learning event extraction: approaches and applications. IEEE Trans. Neural Netw. Learn. Syst. **35**, 6301–6321 (2021)
14. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 167–176. Association for Computational Linguistics, Beijing, China (2015). https://doi.org/10.3115/v1/P15-1017. https://aclanthology.org/P15-1017
15. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 300–309. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/N16-1034. https://aclanthology.org/N16-1034
16. Tian, C., Zhao, Y., Ren, L.: A Chinese event relation extraction model based on bert. In: 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), pp. 271–276 (2019). IEEE
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
18. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
19. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
20. Lu, Y., Lin, H., Xu, J., Han, X., Tang, J., Li, A., Sun, L., Liao, M., Chen, S.: Text2event: controllable sequence-to-structure generation for end-to-end event extraction. arXiv preprint arXiv:2106.09232 (2021)
21. Wei, K., Sun, X., Zhang, Z., Zhang, J., Zhi, G., Jin, L.: Trigger is not sufficient: exploiting frame-aware knowledge for implicit event argument extraction. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Interna-

tional Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4672–4682. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-long.360. https://aclanthology.org/2021.acl-long.360
22. Wei, K., Sun, X., Zhang, Z., Jin, L., Zhang, J., Lv, J., Guo, Z.: Implicit event argument extraction with argument-argument relational knowledge. IEEE Trans. Knowl. Data Eng. **35**(9), 8865–8879 (2023). https://doi.org/10.1109/TKDE.2022.3218830
23. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '12, pp. 1104–1112. Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2339530.2339704. https://doi.org/10.1145/2339530.2339704
24. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 69–76. Association for Computational Linguistics, Hong Kong (2000). https://doi.org/10.3115/1075218.1075228
25. Chen, Y., Liu, S., Zhang, X., Liu, K., Zhao, J.: Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 409–419. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1038. https://aclanthology.org/P17-1038
26. Wei, K., Yang, Y., Jin, L., Sun, X., Zhang, Z., Zhang, J., Li, X., Zhang, L., Liu, J., Zhi, G.: Guide the many-to-one assignment: Open information extraction via IoU-aware optimal transport. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4971–4984. Association for Computational Linguistics, Toronto, Canada (2023). https://doi.org/10.18653/v1/2023.acl-long.272. https://aclanthology.org/2023.acl-long.272
27. Abebe, M.A., Tekli, J., Getahun, F., Chbeir, R., Tekli, G.: Generic metadata representation framework for social-based event detection, description, and linkage. Knowl.-Based Syst. **188**, 104817 (2020). https://doi.org/10.1016/j.knosys.2019.06.025
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Chang, C.-H., Liao, Y.-C., Yeh, T.: Event source page discovery via policy-based rl with multi-task neural sequence model. In: International Conference on Web Information Systems Engineering, pp. 597–606 (2022). Springer
30. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/N16-1030. https://aclanthology.org/N16-1030