Explicit, Implicit, and Scattered: Revisiting Event Extraction to Capture Complex Arguments

Omar Sharif, Joseph Gatto, Madhusudan Basak, Sarah M. Preum
Department of Computer Science, Dartmouth College
{omar.sharif.gr, sarah.masud.preum}@dartmouth.edu

Abstract

Prior works formulate the extraction of event-specific arguments as a span extraction problem, where event arguments are **explicit** — i.e. assumed to be contiguous spans of text in a document. In this study, we revisit this definition of Event Extraction (EE) by introducing two key argument types that cannot be modeled by existing EE frameworks. First, **implicit arguments** are event arguments which are *not* explicitly mentioned in the text, but can be inferred through context. Second, **scattered arguments** are event arguments that are composed of information scattered throughout the text. These two argument types are crucial to elicit the full breadth of information required for proper event modeling.

To support the extraction of explicit, implicit, and scattered arguments, we develop a novel dataset, **DiscourseEE**, which includes 7,464 argument annotations from online health discourse. Notably, 51.2% of the arguments are implicit, and 17.4% are scattered, making DiscourseEE a unique corpus for complex event extraction. Additionally, we formulate argument extraction as a *text generation problem* to facilitate the extraction of complex argument types. We provide a comprehensive evaluation of state-of-the-art models and highlight critical open challenges in generative event extraction. Our data and codebase are available at https://omar-sharif03.github.io/DiscourseEE.

1 Introduction

Event Extraction (EE) is a challenging yet crucial NLP task required for event-centric information extraction. EE is the composition of two tasks: (i) Event Detection (ED), identifying *if* an event occurs in a text and (ii) Event Argument Extraction (EAE), extracting event-specific details or event arguments according to a pre-defined event ontology. Existing works in EE have two key limitations.

First, most prior works are focused on event extraction from formal texts (e.g., news or Wikipedia articles) (Doddington et al., 2004; Tong et al., 2022; Du and Cardie, 2020). This makes existing EE systems limited in their capacity to model other text sources such as social media or other colloquial text (Karimiziarani, 2022; Lei et al., 2024). Insufficient data for EE on social media limits the ability of EE to facilitate downstream tasks such as mining online discourse (Jain et al., 2016), tracking dynamic events, knowledge base construction, rumor, and misinformation detection (Wu et al., 2022), moral value understanding (Zhang et al., 2024c), or characterization of user behaviors (Rosa et al., 2020).

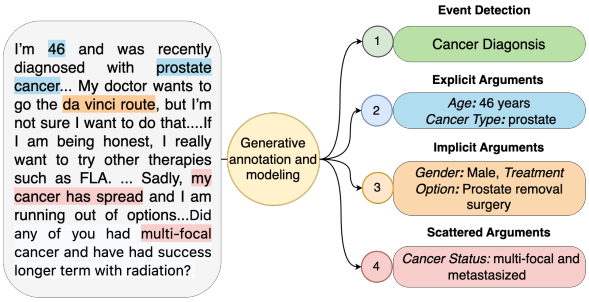

Figure 1: An example demonstrating complex event arguments that are prevalent in online discourse. This Reddit post is narrated by a newly diagnosed prostate cancer patient who is seeking treatment information from online peers on the r/ProstateCancer subreddit. In addition to explicit arguments, it contains implicit and scattered arguments that cannot be extracted using one contiguous span of text.

Second, existing works in EAE extract an argument as *a span in the input text* (Huang et al., 2024). This is extremely limiting as many event arguments are implicit and only discernible from subtext. In Figure 1, we illustrate the need for complex argument types through the lens of a Reddit post written by a newly diagnosed cancer patient. We find that only a small fraction of crucial event-specific details can be tied to contiguous spans of text in the post — demanding a more complex EE solution.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12061–12081
November 12-16, 2024

For example, the patient's consideration of prostate removal surgery is only discernible through a mention of the "da vinci route", which is in reference to the Da Vinci Surgical Robot. Such implicit information is crucial to providing in-depth event details and will improve the accuracy of large-scale event-centric information aggregation efforts.

In this paper, we address these two limitations by reformulating EE annotation as text generation and introducing a novel dataset, **DiscourseEE** — which contains EE annotation for health-related discourse on Reddit. By focusing on health discourse, we can provide a nuanced understanding of healthcare needs which has significant implications for public health research (Parekh et al., 2024; Guzman-Nateras et al., 2022; Romano et al., 2024; De Choudhury et al., 2013). DiscourseEE introduces a novel EE annotation strategy that facilitates the extraction of the following three argument types: (i) ***Explicit Arguments:*** event details that are found directly in the document. (ii) ***Implicit Arguments:*** event details which are *not* directly mentioned in the document but can be inferred using context. (iii) ***Scattered Arguments:*** event details which are the composition of multiple pieces of information scattered throughout the document.

While there is significant prior work on the extraction of explicit arguments, DiscourseEE is the first to introduce implicit and scattered argument extraction. When compared to prior work, our EE formulation adds significant depth to the amount of event information that can be extracted, making DiscourseEE well-suited to improve a range of downstream tasks such as question-answering (Jiang and Kavuluru, 2024), or rumor (Li et al., 2019), conflicting information (Preum et al., 2017b,a; Gatto et al., 2023), and misinformation (Wu et al., 2022) detection from complex, online discourse. Additionally, by changing the EAE paradigm from span extraction to text generation, we better align EE with the abilities of Large Language Models (LLMs), which have been shown to have limited capacity on extractive EE tasks in prior work (Huang et al., 2024; Gao et al., 2023). Our major contributions are as follows.

- We introduce DiscourseEE, a dataset for characterizing event-level information in social media discourse. In addition to *explicit* arguments, we introduce two prevalent yet overlooked argument types: *implicit* and *scattered*, broadening the scope of accessible knowledge in EE. DiscourseEE uses a novel event ontology, with 7,464 event-argument annotations leveraging relevant data from a health-related subreddit, i.e., a topic-specific community on Reddit. 51.2% of arguments in DiscourseEE are implicit, and 17.4% are scattered. To the best of our knowledge, this is the first large-scale, annotated social media dataset on event extraction with annotations for explicit, implicit, and scattered arguments.
- We reformulate EE annotation as a text generation problem to enable the extraction of non-explicit event information. We benchmark a diverse set of state-of-the-art event-extraction models on DiscourseEE, including both extractive models and several relevant LLMs. We identify limitations of existing models on DiscourseEE, motivating future works in EE.

2 Event Extraction via Text Generation

Generative Event Argument Extraction (EAE): Prior works on EE have exclusively focused on extracting arguments, which are *continuous spans* that can be found directly in the text (Du and Cardie, 2020; Tong et al., 2022). In a real-world setting, this problem formulation limits one's ability to extract complex arguments such as those which are the composition of *scattered information* throughout a text, or *implicit information* with no direct mention in a text. In this study, we argue that implicit and scattered arguments are *crucial* to understanding an event and that classic approaches to EE can not capture these arguments. To address this, we reformulate argument extraction as text generation rather than span extraction tasks and annotate arguments as natural texts.

Trigger-Free Event Detection (ED): Many prior works perform Event Detection (ED) by identifying event triggers — where the trigger is a word or phrase that best indicates the occurrence of an event (Du and Cardie, 2020; Lu et al., 2023). Recently, various studies used trigger-free ED, where texts are simply classified as containing an event without specific grounding to a trigger phrase (Tong et al., 2022; Liu et al., 2019a). We also adopt a trigger-free ED formulation as DiscourseEE events can be deeply implicit or the result of phrases scattered throughout a document, making it difficult to tie an event to a single trigger phrase.

Evaluating Generative Event Extraction (EE) Outputs: A core challenge of implementing EE as

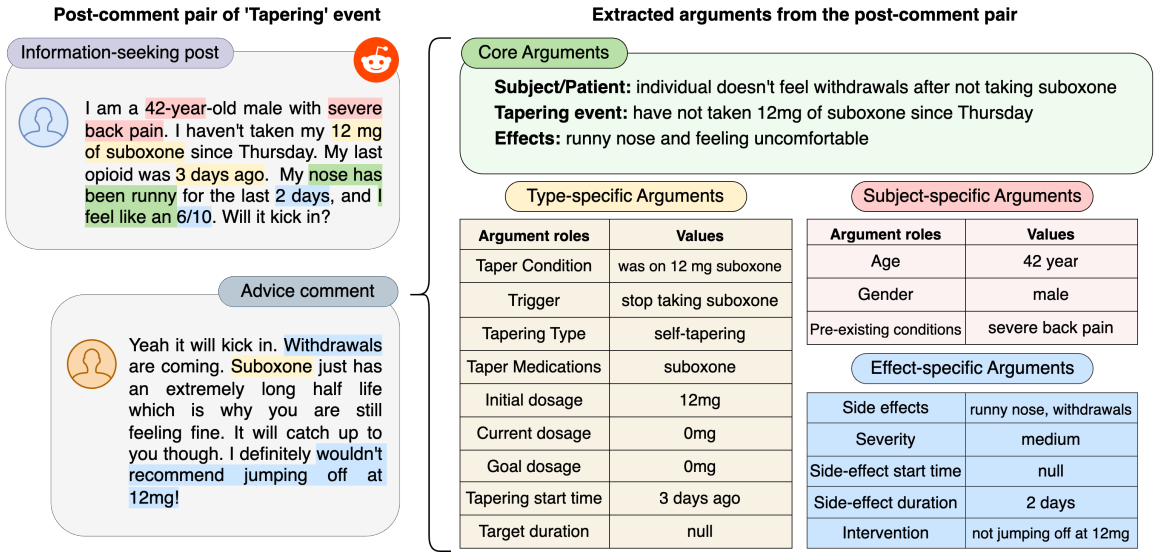

Figure 2: Example annotation in DiscourseEE. Core arguments capture the key aspects of the advice, while type-specific, subject-specific, and effect-specific arguments capture the fine-grained details. An argument can be explicit, implicit, or scattered throughout the document, e.g., as the individual is tapering suboxone, the *goal dosage* is '0mg.' which is not directly mentioned in the text. We separately annotate the arguments from posts and comments. However, due to label sparsity, we merge them during model evaluation. The argument value is set to *'null'* if absent, and multiple values for a role are comma-separated.

a text generation problem is evaluating the quality of generated arguments. In prior EE formulations, all arguments correspond to start/end indices in a text — thus, one can simply evaluate if the model has produced the 'exact match' argument, i.e., identified the correct span (Huang et al., 2024). Unfortunately, it is well established that 'exact match' evaluation is not well-suited for models that generate human-like responses, such as LLMs (Wadhwa et al., 2023). Thus, if we attempt to translate the exact match evaluation to the generative setting, performance is severely underestimated as correct outputs can vary from ground truth. For example, if the ground truth argument for the role *side effect* is "`runny nose`", a generative model may correctly output one of `[runny nose, drippy nose, sniffly, nasal discharge]` yet only one answer would be accepted by exact match.

To solve this problem, we employ a relaxed match approach based on semantic similarity to achieve a more accurate evaluation. To account for variations in text, we consider the generated output and the ground-truth label for an argument to be similar if the semantic similarity score exceeds 0.75. We compute BERT-based semantic similarity (Reimers and Gurevych, 2019). This threshold of 0.75 was determined through manual observation of the outputs. We acknowledge that changes in the threshold will affect the model's relaxed match score and suggest setting the threshold according to the downstream task. We also calculate the 'exact-match' score for comparability with previous evaluations. We consider two sentences to be exactly matched when their semantic similarity is 1.0. For evaluation, we calculate the F1 score based on both relaxed and exact matches and denote them as **RM_F1** and **EM_F1**, respectively.

3 Event Ontology Design

Event Types: Analyzing online discourse through an event-argument framework can inform data mining and knowledge discovery for several impactful domains including but not limited to healthcare, politics, public policy, finance, and law. As demonstrated in the motivating example in Figure 1, millions of patients across the world seek informational and emotional support in online peer communities (e.g., Reddit, Facebook) on different conditions, e.g., mental health, pregnancy, recovery from substance use disorder, cancer, and other chronic diseases. While our proposed method can be applied to any of these use cases, we found only one labeled, large-scale event dataset for online discourse / social media discussion, namely TREAT-ISE (Sharif et al., 2024). This dataset covers health discourse regarding medications for opioid use disorder (MOUD), a critical yet stigma-

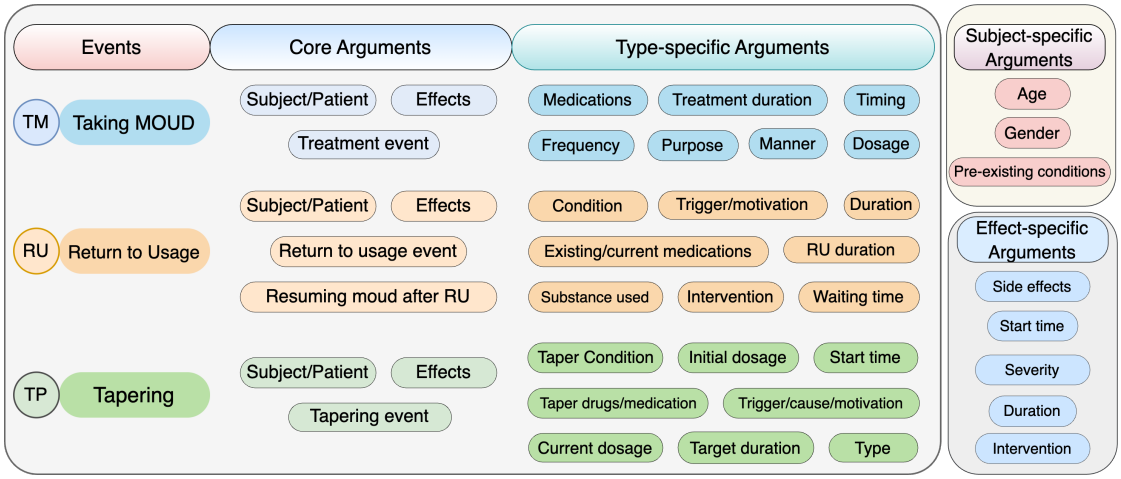

Figure 3: Event ontology of DiscourseEE dataset. Details of arguments provided in Table 11.

tized public health topic. OUD remains a leading cause of mortality in the US, incurring a massive economic toll, estimated at 1.02 trillion dollars annually (Florence et al., 2021). There exists a lot of misperception and knowledge gaps regarding OUD treatment that impact treatment initiation and adherence. Thousands of affected individuals seek treatment information online among peers and make critical treatment decisions due to stigma, distrust on traditional healthcare, and lack of access to care. Event-driven analysis of such online discourse can inform MOUD-related public policy, patient communication and education. The dataset introduced by Sharif et al. (2024) comprises five information-seeking events on medications for opioid use disorder (MOUD) from Reddit. These events include *Accessing MOUD, Taking MOUD, Psychophysical effects, Relapse, Tapering*. We select three of these five event types as part of our event ontology: Taking MOUD (TM), Relapse (RL), and Tapering (TP). We exclude Accessing MOUD as they lack relevance to health advice. Additionally, we consider psychophysical effects as an event role instead of an event due to it's prevalence in all classes. In this work, we refer to 'Relapse' as *Return to Usage* (RU) in the rest of the paper as the former term can be stigmatizing[1]. Detailed descriptions of the relevant event types are provided in Appendix A.

Capture Complex Arguments We have a total of four types of arguments. **(1) Core Arguments:** longer texts containing the details of the advice event (e.g., subject receiving advice, advised treatment, outcomes or side effects of the treatment). The goal of annotating the core arguments is to get a high-level summary of the advice, which is difficult to infer from traditional annotations of short, discontinuous text spans. **(2) Type-specific Arguments:** words or phrases highlighting an advice event's specifics (e.g., advised treatment duration, medications). These arguments help to get a nuanced understanding of the specific event type. **(3) Subject-specific Arguments:** To understand advice, it is crucial to know to whom advice is given. These arguments are words or phrases providing details about the subject/patient (e.g., age, gender, prior medical history, or other social determinants of health). **(4) Effect-specific Arguments:** are words or phrases providing details about psychophysical effects (e.g., severity, duration) individuals experience or can experience that are related to an event, e.g., taking MOUD.

We defined **10 core, 23 type-specific, 3 subject-specific, and 5 effect-specific arguments** across three event types. Figure 3 shows the event ontology we followed to develop DiscourseEE. To our knowledge, our dataset is the first discourse-level event extraction dataset enriched with fine-grained argument annotation capturing the real-world complexity of health advice on social media, including scattered spans and implicit arguments. Figure 2 shows a sample annotation of arguments for the 'tapering' event. Short description of argument roles for each event type provided in Table 11.

4 DiscourseEE Dataset Curation

4.1 Data Collection and Filtering

DiscourseEE expands the *TREAT-ISE* dataset (Sharif et al., 2024; Basak et al., 2024) which contains 5,412 information seeking Reddit posts on

[1]https://tinyurl.com/axtrbwrd

Step	#Posts	#Comments
Original dataset	5,412	39,300
Filtering target events	3,713	25,769
Filtering *information-scarce* samples	932	6,214

Table 1: Dataset summary at different stages of filtering

	TM	**RU**	**TP**
Avg. sample length (#words)	112.80	117.28	114.33
#Arguments (without null arguments)	1492	1213	1140
Avg. #arguments per sample	8.38	10.27	11.4
# Explicit, implicit, scattered arguments			
– explicit	462	449	295
– implicit	756	546	668
– scattered	274	218	177

Table 2: DiscourseEE statistics across three event types. Here, TM, RU, and TP indicate 'Taking MOUD', 'Return to Usage', and 'Tapering' events, respectively. Additional statistics of the dataset are shown in Table 7.

recovery treatment. We expand TREAT-ISE to include both *information seeking* and *information sharing* data by sourcing advice-centric comments on information seeking Reddit posts. We collect the 39,300 comments associated with TREAT-ISE dataset. We keep the posts/comments of our selected event types and discard the rest. We obtained 3,713 post threads comprising 25,769 comments and applied a two-step filtering process to discard information-scarce post-comment discussions. (1) **Sufficient Discourse Filtering:** We include only threads with at least 4 comments, ensuring a certain level of peer interaction. After excluding threads failing to meet this criterion, we have 2,432 posts with 23,101 comments. (2) **Discourse Length Filtering:** We removed threads where the initial posts (title and body) contained fewer than 10 words. We also removed comments with fewer than 10 words. We chose 10 as the filtering threshold here to ensure argument annotation quality as short samples do not have sufficient arguments. After this filtering process, we obtained 932 posts with 6,214 comments. These selected samples were then utilized for advice and event argument annotation. Table 1 illustrates data statistics on different filtering stages.

4.2 Identifying Information Sharing Content

DiscourseEE aims to model event-centric information from social discourse, which contains information-seeking posts and information-sharing comments. Prior work has established how to source the former, but in this work, we introduce a method of automatically sourcing information sharing comments, i.e., containing advice. In other words, we aim to identify comments that provide *answers* to information-seeking content.

We follow the model in the loop (Chakrabarty et al., 2022) annotation protocol for labeling post-comment pairs as advice followed by human-level verification. We apply different state-of-the-art open-source (Mistral) and close-sourced (GPT-4, GPT-3.5) LLMs to identify the advice in a comment. Note that since the context of the comment heavily depends on the post, we allow the model to view post content when determining if a comment contains advice. The GPT-4 model achieved the highest precision of 0.94, and we employ it to identify advice samples. Out of 6,214 post-comment pairs, the model categorized 2,934 as advice. This question-advice (post-comment) pair is utilized in the subsequent event argument annotation step. We manually validate the advice labeling accuracy of GPT-4 as discussed in Appendix B.

4.3 Argument Annotation

In DiscourseEE, 51.2% arguments are implicit, and 17.4% are scattered arguments. We focused on four types of arguments (*core*, *type-specific*, *subject-specific*, *effect-specific*) for each event. Figure 2 illustrates a sample annotation with associated argument values. Annotating such arguments takes more effort and time than classification tasks or span-based annotation. For the sake of feasibility, we randomly selected 396 post-comment pairs for argument annotation. Previously, researchers highlighted limitations of crowd-sourced annotation for complex tasks (Zhang et al., 2024a), including event detection from online discourse (Parekh et al., 2024; Sharif et al., 2024). Generative argument annotation in such data is even more challenging. It requires domain knowledge and interactive, progressive training sessions to ensure annotators understand the nuances. We decided to engage both domain experts and paid annotators recruited and trained locally at the authors' institution.

Annotators have to write the values for each argument. For each sample, we annotated the comment and the corresponding post. The separated post and comment annotations are sparse. Therefore we merge the annotations and inspect both posts and comments as a pair. Each sample (i.e., post-comment pair) has 19 arguments on average.

Annotation process: To complete the annotation, we formed a diverse group of 8 annotators: 4 graduate and 4 undergraduate students at the authors' institution. Each annotator underwent an extensive, four-week training period involving trial annotations to ensure proficiency in identifying event arguments and understanding the annotation guidelines. For each sample, at least two annotators wrote the *core, type-specific, subject-specific,* and *effect-specific* arguments. Note that all the arguments are annotated separately for each post-comment pair. The major challenges of such manual annotation include domain-specific terms and context, and identifying implicit and scattered arguments. These complexities are discussed in detail in Appendix C. A third annotator reviewed each sample to correct potential errors and resolve any disagreement to ensure the reliability of the annotation process.

Inter-annotator Agreement: Traditional Cohen's kappa is not suitable for our annotations due to the unknown number of disagreements. Following previous works (Sun et al., 2022; Thompson et al., 2018), we choose the F1 score to measure inter-annotator agreement (IAA). Specifically, we used relaxed match F1-score (RM_F1) due to the generative nature of our annotation, which is discussed in Section 2. We have two sets of annotations for each sample. The F1 score is computed by selecting one annotation set as a 'reference' to another. Through progressive training and interactive discussions, we achieve quality annotations (a 0.811 mean IAA score, which indicates substantial agreement). Finally, we have **DiscourseEE**, a novel discourse-level event extraction dataset comprising 7,464 event argument annotations.

4.4 DiscourseEE Summary Statistics

Table 2, shows the statistics of DiscourseEE. Our dataset differs from other general domain (Doddington et al., 2004) and clinical/pharmacovigilance (Ma et al., 2023; Sun et al., 2022) EE datasets because of the higher average length ($\approx$ 115 words) and higher density of arguments per document ($\approx$ 10 words). Thus DiscourseEE is a reasonably sized EE dataset with fairly dense event arguments. We also annotated implicit and scattered arguments written as natural text, providing a novel resource for such argument (68.6% of the total arguments) extraction. More dataset statistics are presented in Appendix D.

5 Benchmarking EE Models

In this section, we detail our experimental setup for benchmarking a wide range of models on ED and EAE for DiscourseEE.

5.1 Event Detection (ED) Models

We formulate ED as a multilabel classification task, as each sample can provide information about multiple events. We employ three transformer-based models (BERT, RoBERTa, and MPNet), two instruction-finetuned models (FLAN-T5-base and FLAN-T5-large), and five large language models (Gemma-7B, Mixtral-8x7B, Llama3-8B, Llama3-70B, and GPT-4) for event detection. Our objective is to assess the feasibility of ED using large models and to examine the effects of scaling. Consequently, we experimented with models ranging from 7B to 70B parameters, including the closed-source GPT-4. These models demonstrated SOTA performance across various event extraction (Zhang et al., 2024b; Wang et al., 2023) and information extraction (Wadhwa et al., 2023) tasks. Detailed descriptions of these models are provided in Appendix E.

5.2 Argument Extraction Models

We perform comprehensive experiments in three distinct settings: Extractive-QA, Generative-QA, and LLM-based generation with varying prompt types. In this set of experiments, we assume knowledge of the ground truth event type. We focus on a question-answering (QA) based approach as previous works achieved SOTA results with this model type (Du and Cardie, 2020; Hsu et al., 2022).

Extractive-QA: Following (Du and Cardie, 2020), we implement a span extraction EAE baseline using question-answering. Specifically, the model is trained to map `(Question, Input Text)` $\rightarrow$ `(Argument)`, where each question is a function of the role we wish to extract[2]. Note that since DiscourseEE is formulated as a generative task, we do not have span-level annotations. We thus use a BERT model pre-trained on general question-answering data (Rajpurkar et al., 2016) to extract argument spans.

Generative-QA: We employ the instruction fine-tuning approach (Zhou et al., 2023) to develop the generative-QA models. The instruction set is created from the training data and the format is shown in Figure 7. To examine impact of model size

[2]Questions used for this baseline can be found here: https://tinyurl.com/44e8u5fx

	Taking MOUD			Return to Usage			Tapering			
Model	C-A	TS-A	SE-A	C-A	TS-A	SE-A	C-A	TS-A	SE-A	Mean (RM_F1)
Extractive-QA	4.98	19.26	16.08	14.95	29.15	15.39	19.98	18.65	15.74	17.13
Generative-QA										
– FLAN-T5 (Base)	34.99	37.22	11.40	25.37	20.31	11.37	38.78	40.94	17.53	26.44
– FLAN-T5 (Large)	41.70	45.61	23.92	38.44	27.41	19.79	44.04	51.92	26.93	35.53
LLMs with Zero-Shot Description-guided Prompt										
Gemma (7B)	26.84	39.11	31.83	20.48	31.75	28.66	31.24	28.64	30.26	29.87
Mixtral (8x7B)	34.19	30.70	33.94	33.77	33.73	31.80	40.55	41.51	39.47	35.52
Llama-3 (8B)	32.88	48.45	32.48	33.43	37.75	27.49	41.33	42.88	35.54	36.91
Llama-3 (70B)	41.35	39.39	25.28	35.20	38.80	30.78	41.54	40.57	28.83	35.75
GPT-4	37.88	46.34	30.50	43.56	41.94	39.68	42.90	38.43	43.15	40.49
LLMs with Zero-Shot Question-guided Prompt										
Gemma (7B)	25.52	46.46	29.00	15.66	33.51	24.94	27.63	36.39	32.38	30.16
Mixtral (8x7B)	36.89	27.88	31.31	32.53	33.06	20.44	33.67	32.95	42.69	32.38
Llama-3 (8B)	34.88	42.19	27.64	33.31	32.38	31.48	25.82	45.89	41.09	34.96
Llama-3 (70B)	37.28	42.31	25.02	35.94	41.42	33.49	36.13	40.28	34.90	36.31
GPT-4	35.77	47.38	38.83	39.75	40.41	49.35	40.69	44.41	41.26	**41.98**

Table 3: Performance (avg. of 3 runs) of the models for event argument extraction across all argument types in relaxed match F1-score (RM_F1). C-A, TS-A, and SE-A denote core, type-specific, and subject-effect arguments. The mean RM_F1 is calculated by averaging the scores across all argument types for all three classes. The best score in each column is underlined. Model superiority is determined based on the mean RM_F1 score. Models performance based on exact match F1-score (EM_F1) presented in Table 8.

	P	R	F1
Transformer-based Models			
BERT	48.89	52.90	50.48
MPNet	47.91	65.88	54.95
RoBERTa	51.74	59.59	55.26
Instruction-tuned Models			
FLAN-T5 (base)	54.12	53.48	51.26
FLAN-T5 (large)	57.61	54.78	55.63
LLMs with Zero-Shot Prompt			
Gemma-7B	54.90	56.25	50.12
Mixtral-8x7B	55.09	54.31	51.75
Llama3-8B	60.62	59.29	55.42
Llama3-70B	61.21	62.38	59.84
GPT-4	62.36	64.62	**61.40**

Table 4: Performance comparison (avg. of 3 runs) of the models for event detection (ED). P, R, and F1 indicate precision, recall, and macro-F1 scores, respectively.

on performance, we fine-tune two smaller models: FLAN-T5-base and FLAN-T5-large (Chung et al., 2024), as they contain less than 1 billion parameters.

LLM-based Generation: We conduct extensive experiments using both open-source and closed-source LLMs of various parameter sizes, including Gemma (7B), Mixtral (8x7B), Llama-3 (8B and 70B), and GPT-4. Models are evaluated in a zero-shot setting with two types of prompts: description-guided and question-guided. In the *description-guided prompts*, role descriptions guide the models to extract arguments. In contrast, in the *question-guided prompts*, questions are used to extract arguments similar to generative-QA and extractive-QA approaches. Since each event in DiscourseEE has an average of 19 arguments, It will require a large number of inferences if we extract arguments for each role separately. On the other hand, extracting all arguments from a sample with only one inference (a) results in noisy outputs that are difficult to parse and (b) reduces accuracy. So, to manage parsing complexity and inference costs, we employ a divide-and-conquer strategy. We note that *subject-specific* arguments are rare in DiscourseEE. To manage experimental complexity we merge *subject-specific* and *effect-specific* arguments roles during evaluation. For the rest of the paper, we call it *subject-effect* arguments. We extract *core*, *type-specific*, and *subject-effect* arguments separately for each sample and then merge them. The generic prompt template is illustrated in Figure 8, with further prompt details in Table 10. Details about each model, instruction prompt, fine-tuning, and hyperparameters are presented in Appendix E, F.

5.3 Experimental Setup

Data Splits: DiscourseEE is partitioned into three mutually exclusive subsets: train (246 samples), validation (50 samples), and test set (100 samples). The same test set is used across all models for both tasks to ensure unbiased evaluation. All the

training and fine-tuning experiments were done on the GPU-accelerated Google Colab platform.

Prompt Setting: Different LLMs require prompts and in-context examples optimized specifically for each model (Ziems et al., 2024). In practice, users adopt a trial-and-error approach to find the optimal prompt for a model (Zamfirescu-Pereira et al., 2023). We evaluate a wide range of LLMs in a zero-shot setting to mitigate biases and reduce computational costs associated with finding the optimal prompts in few-shot setting. We use the same prompt across all models to (a) eliminate the confounding factor of prompt variation and (b) ensure a fair comparison of the models.

Performance Metrics: We use macro F1-score to evaluate ED performance. For EAE, we employ the relaxed-match F1-score (RM_F1) to identify the best models and also compute the exact match F1-score (EM_F1). Scores are computed following prior work (Peng et al., 2023). The details of RM_F1 and EM_F1 are discussed in Section 2. Additionally, for EAE, we report the overall F1 score as well as the per-event type and per-argument type F1 in Table 3.

6 Results and Discussion

Event Detection: Table 4 illustrates the ED results, where GPT-4 achieved the highest F1 score of 61.40. Among the open-source LLMs, Llama-3 (70B) achieved the maximal score of 59.84. We notice a linear relation between model size and performance, with zero-shot performance improving as the model size increases. Interestingly, instruction-fine tuning enabled smaller FLAN-T5 models to achieve comparable performance (55.63). Conversely, the transformer models performed poorly, potentially indicating the complex nature of online discourse and this task.

Event Argument Extraction: Table 3 shows model performance for argument extraction. GPT-4 with question-guided prompting attained the highest mean RM_F1 score of 41.98. The instruction fine-tuned FLAN-T5 (large) model obtained a 35.53 score, outperforming several larger models. This indicates that instruction-tuned models achieve comparable performance when compute resources are limited. The extractive model performed poorly, achieving only a 17.13 RM_F1 score, highlighting their limited scope in capturing implicit and scattered arguments.

Explicit, implicit, and scattered arguments: Table 5 compares the performance of various models in extracting explicit, implicit, and scattered arguments. All models achieved low scores in implicit argument identification, with the best-performing GPT-4 model reaching only 36.53. The extractive model performed poorly, identifying only 9.40% of implicit and 13.44% of scattered arguments. This weak performance is due to implicit arguments lacking direct mentions and scattered arguments consisting of discontinuous spans.

	Relaxed Match (Recall)		
Model	Explicit	Implicit	Scattered
Extractive-QA	32.40	9.40	13.44
Generative-QA			
– FLAN-T5 (Base)	35.49	23.72	33.33
– FLAN-T5 (Large)	45.98	34.15	43.54
LLMs with Zero-Shot Description-guided Prompt			
Gemma (7B)	37.96	24.13	28.31
Mixtral (8x7B)	52.26	27.53	48.20
Llama-3 (8B)	46.09	31.76	38.88
Llama-3 (70B)	52.67	26.99	43.36
GPT-4	53.39	33.46	**54.12**
LLMs with Zero-Shot Question-guided Prompt			
Gemma (7B)	40.02	25.01	27.77
Mixtral (8x7B)	47.83	26.85	50.53
Llama-3 (8B)	41.35	30.67	36.91
Llama-3 (70B)	53.49	28.35	41.21
GPT-4	**55.14**	**36.53**	49.82

Table 5: Performance comparison of explicit, implicit, and scattered argument extraction. Model performance (avg. of 3 runs) is reported based on recall, showing how many explicit, implicit, and scattered arguments are extracted correctly.

Impact of exact-match evaluation: We also evaluate models' performances using the 'exact-match' (EM_F1) metric. Due to space constraints, results are presented in the appendix (see Table 8, 9). These results show a significant performance drop in extracting core and subject-effect arguments, which are often implicit or scattered. To investigate further, we conducted a qualitative human evaluation on a subset of the best-performing GPT-4 outputs. The evaluation revealed that although the outputs are semantically similar, the exact-match evaluation frequently marked them as incorrect, underestimating the performance of the models. Addressing these evaluation issues is a crucial future direction for generative EE research.

7 Related Work

Event extraction with generative models: Prior studies have approached EE as a token-level classi-

fication or extractive task (Nguyen et al., 2016; Du and Cardie, 2020; Wang et al., 2021). Recently, EE has been formulated as a text generation task using pre-trained language models, where the model is prompted to fill in natural language templates (Hsu et al., 2022; Lu et al., 2021). With the advent of LLMs, generation-based EE gained more traction (Wang et al., 2023; Gao et al., 2023). However, most of these generative models are evaluated using span-based annotated datasets, which can underestimate their performance (Huang et al., 2024). This is because the models' predictions may differ from the exact ground-truth spans yet still be accurate. This work addresses this gap by proposing a relaxed-match evaluation metric, and presenting a new dataset and benchmarks to facilitate generative event extraction.

Event extraction from social media: Existing research on social media primarily addresses event detection and often overlooks argument extraction, a gap DiscourseEE addresses. For instance, Parekh et al. (2024) detected epidemic-related events from tweets, while Guzman-Nateras et al. (2022) identified suicide-related events on Reddit. Arguments in social media data vary in span and are often ambiguous, implicit, and scattered. Our approach to implicit and scattered argument formulation differs from existing works. We define implicit arguments as those not directly mentioned in the document but can be inferred from the context and scattered arguments composed of information throughout the text. In contrast, existing works define implicit arguments as those that are *explicitly mentioned but outside the fixed sentence window* of the event's trigger (Ebner et al., 2020; Zhang et al., 2020; Liu et al., 2021). They overlook arguments that are entirely implicit or scattered. The event argument aggregation task done by (Kar et al., 2022) is aligned with our work, but they do not model implicit and scattered arguments. Moreover, they only focus on six discrete argument roles (Time, Place, Casualties, After Effects, Reason, and Participant) without grounding them with specific events. In contrast, our EE formulation adds significant depth to the amount of event information that can be extracted. We annotate 41 arguments from three events and characterize four types of complex arguments: core, type-specific, subject-specific, and effect-specific. Our formulation better captures the complexity and nuance of real-world EE applications.

8 Conclusion

This paper presents DiscourseEE, a discourse-level EE dataset with fine-grained annotations of complex event arguments and develops a novel pipeline for extracting these arguments. DiscourseEE provides a new resource for studying implicit and scattered arguments within complex social discourse, which has not been previously explored. The best performing GPT-4 model achieved only a 41.98% overall F1 score, highlighting the challenges of extracting such intricate arguments. Specifically, the models accurately extracted only 36.53% of implicit arguments and 54.12% of scattered arguments, underscoring that effective argument extraction remains an open challenge. We believe DiscourseEE fills a critical gap in EE research and provides a valuable, timely dataset and benchmark for generative event extraction.

9 Limitations

One limitation of our benchmarking effort is the reliance on relaxed matching (RM_F1) to assess model performance. While we attempted to select an appropriate threshold by comparing model outputs with ground truth, some model outputs may still be inaccurate. Thus, we also report performance using the exact match approach (EM_F1). However, EM_F1 significantly underestimates model performance, highlighting the need for a more robust evaluation metric in future generative EE research.

DiscourseEE contains 396 annotated pairs with 7,464 arguments, which might seem small. However, this size is comparable to similar works with $\approx$ 8,000 arguments (Doddington et al., 2004; Ma et al., 2023). Unlike previous works (Tong et al., 2022), our annotations require manually writing values for each argument instead of selecting spans, making scaling more difficult. We focused the use of our annotation budget on having high annotation quality, as this work lays the groundwork for future research in the domain. We engaged with domain experts, recruited students, and trained them for the annotation. Each sample is annotated by two annotators and reviewed by an expert. This rigorous process makes the scaling challenging and annotation expensive. In future work, we plan to extend the dataset size as more resources become available.

Finally, we evaluate all models in a zero-shot setting without tailoring prompts specifically for each

one. While model-agnostic prompt optimization or alternative prompting techniques could improve performance, we did not pursue these experiments due to the high computational cost. Our focus is on benchmarking a broad range of models rather than optimizing a single model's output. Future research can explore few-shot learning, chain-of-thought prompting, and other techniques to increase model performance.

Ethical Considerations

This research was approved by the author's institution's Institutional Review Board (IRB).

User Privacy: All data samples were collected and annotated in accordance with the terms and conditions of their respective sources. No identifying personal information that could violate user privacy was collected or shared.

Biases: Any biases in the dataset and model are unintentional. Data annotation was performed by experts and a diverse group of annotators following comprehensive guidelines, and all annotations were reviewed to mitigate potential biases. The developed dataset and model can only be used to detect events and identify arguments we discussed in the paper. The scope of using these resources for malicious reasons is minimal.

Intended Use: We intend to make our dataset and models accessible to encourage further research on generative event extraction.

Annotation: Annotation was conducted by experts and trained student annotators. Key characteristics of our annotators include: (a) ages 22-30, (b) a mix of native and non-native English speakers, and (c) 1-5 years of research experience. We provided detailed annotation guidelines, including background knowledge of health advice, event extraction, and the type of information we wanted to extract to mitigate potential biases. All annotators were compensated as per the standard paying rate of the author's institution.

Reproducibility The model, parameter, and implementation details are presented in Appendix E, F. Our code, evaluation and the dataset are available at https://omar-sharif03.github.io/DiscourseEE.

References

Madhusudan Basak, Omar Sharif, Sarah E. Lord, Jacob T. Borodovsky, Lisa A. Marsch, Sandra A. Springer, Edward Nunes, Charlie D. Brackett, Luke J. ArchiBald, and Sarah M. Preum. 2024. A thematic framework for analyzing large-scale self-reported social media data on opioid use disorder treatment using buprenorphine product.

Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022. FLUTE: Figurative language understanding through textual explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7139–7159, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, and et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In *Proceedings of the international AAAI conference on web and social media*, volume 7, pages 128–137.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Curtis Florence, Feijun Luo, and Ketra Rice. 2021. The economic burden of opioid use disorder and fatal opioid overdose in the united states, 2017. *Drug and Alcohol Dependence*, 218:108350.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction.

Joseph Gatto, Madhusudan Basak, and Sarah Masud Preum. 2023. Scope of pre-trained language models for detecting conflicting health information. *Proceedings of the International AAAI Conference on Web and Social Media*, 17(1):221–232.

Gemini-Team. 2024. Gemini: A family of highly capable multimodal models.

Gemma-Team. 2024. Gemma: Open models based on gemini research and technology.

Luis Guzman-Nateras, Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Nguyen. 2022. Event detection for suicide understanding. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1952–1961, Seattle, United States. Association for Computational Linguistics.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12804–12825, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ajit Jain, Girish Kasiviswanathan, and Ruihong Huang. 2016. Towards accurate event detection in social media: A weakly supervised approach for learning implicit event indicators. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 70–77, Osaka, Japan. The COLING 2016 Organizing Committee.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, and et al. 2024. Mixtral of experts.

Yuhang Jiang and Ramakanth Kavuluru. 2024. Covid-19 event extraction from twitter via extractive question answering with continuous prompts. *Studies in health technology and informatics*, 310:674.

Debanjana Kar, Sudeshna Sarkar, and Pawan Goyal. 2022. ArgGen: Prompting text generation models for document-level event-argument aggregation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 399–404, Online only. Association for Computational Linguistics.

Mohammadsepehr Karimiziarani. 2022. A tutorial on event detection using social media data analysis: Applications, challenges, and open problems.

Yuanyuan Lei, Md Messal Monem Miah, Ayesha Qamar, Sai Ramana Reddy, Jonathan Tong, Haotian Xu, and Ruihong Huang. 2024. EMONA: Event-level moral opinions in news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5239–5251, Mexico City, Mexico. Association for Computational Linguistics.

Quanzhi Li, Qiong Zhang, Luo Si, and Yingchi Liu. 2019. Rumor detection on social media: Datasets, methods and opportunities. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 66–75, Hong Kong, China. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019a. Event detection without triggers. 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)*, pages 735–744, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2023. Event extraction as question generation and answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1666–1688, Toronto, Canada. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023. DICE: Data-efficient clinical event extraction with generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15898–15917, Toronto, Canada. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. 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*, pages 300–309, San Diego, California. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, and Sam Altman et al. 2024. Gpt-4 technical report.

Tanmay Parekh, Anh Mac, Jiarui Yu, Yuxuan Dong, Syed Shahriar, Bonnie Liu, Eric Yang, Kuan-Hao Huang, Wei Wang, Nanyun Peng, and Kai-Wei Chang. 2024. Event detection from social media for epidemic prediction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5758–5783, Mexico City, Mexico. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

Sarah Masud Preum, Abu Sayeed Mondol, Meiyi Ma, Hongning Wang, and John A Stankovic. 2017a. Preclude: Conflict detection in textual health advice. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 286–296. IEEE.

Sarah Masud Preum, Abu Sayeed Mondol, Meiyi Ma, Hongning Wang, and John A Stankovic. 2017b. Preclude2: Personalized conflict detection in heterogeneous health applications. *Pervasive and Mobile Computing*, 42:226–247.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

William Romano, Omar Sharif, Madhusudan Basak, Joseph Gatto, and Sarah Preum. 2024. Theme-driven keyphrase extraction from social media on opioid recovery. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Renata Lopes Rosa, Marielle Jordane De Silva, Douglas Henrique Silva, Muhammad Shoaib Ayub, Dick Carrillo, Pedro HJ Nardelli, and Demostenes Zegarra Rodriguez. 2020. Event detection system based on user behavior changes in online social networks: Case of the covid-19 pandemic. *Ieee Access*, 8:158806–158825.

Omar Sharif, Madhusudan Basak, Tanzia Parvin, Ava Scharfstein, Alphonso Bradham, Jacob T Borodovsky, Sarah E Lord, and Sarah M Preum. 2024. Characterizing information seeking events in health-related social discourse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22350–22358.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Paul Thompson, Sophia Daikou, Kenju Ueno, Riza Batista-Navarro, Jun'ichi Tsujii, and Sophia Ananiadou. 2018. Annotation and detection of drug effects in text for pharmacovigilance. *Journal of cheminformatics*, 10:1–33.

MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Xueqing Wu, Kung-Hsiang Huang, Yi Fung, and Heng Ji. 2022. Cross-document misinformation detection based on event graph reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 543–558, Seattle, United States. Association for Computational Linguistics.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Hansong Zhang, Shikun Li, Dan Zeng, Chenggang Yan, and Shiming Ge. 2024a. Coupled confusion correction: Learning from crowds with sparse annotations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16732–16740.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024b. ULTRA: Unleash LLMs' potential for event argument extraction through hierarchical modeling and pairwise self-refinement. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8172–8185, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xinliang Frederick Zhang, Winston Wu, Nicholas Beauchamp, and Lu Wang. 2024c. MOKA: Moral knowledge augmentation for moral event extraction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4481–4502, Mexico City, Mexico. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

Appendix

A Details of Event Ontology

We select three event types in DiscourseEE, and the definitions of selected event types are as follows.

Taking MOUD (TM): Events related to MOUD regimen details, e.g., advice about timing, dosage, frequency of taking a MOUD, suggestions about splitting and missing a dose. Analyzing advice from this event can surface potential misconceptions and concerns about MOUD administration that negatively impact treatment adherence.

Return to Usage (RU): Events related to relapsing or using other substances during recovery. Such substance use can be attributed to recreational purposes or for self-medication (e.g., marijuana for sleep). Advice in this event class can help unearth specific information individuals provide concerning recreational and medical usage of substances.

Tapering (TP): Events related to reducing the dose or frequency of MOUD and eventually quitting MOUD. Although the current standard of care recommends consulting healthcare providers for tapering MOUD, individuals often resort to self-managed tapering strategies. Analyzing advice from this class can inform addiction researchers and clinicians about the context of self-managed tapering strategies (e.g., why and when people self-taper) and their effectiveness (what works for whom).

B Advice Annotation

Figure 5 illustrates the DiscourseEE development pipeline. Recent works have indicated that LLMs, such as GPT and LLaMA, can be effective zero-shot data annotation tools (He et al., 2024). LLMs have demonstrated the ability to reliably classify texts in various domains without supervision (Ziems et al., 2024). Therefore, we apply different state-of-the-art open-source (Mistral) and close-sourced (GPT-4, GPT-3.5) LLMs to identify potential advice. We developed an evaluation set of 100 samples (post-comment pairs) annotated by human experts as *advice* and *not-advice* to select the best model for our task. As our primary focus is on identifying advice, the **best model is selected based on the precision in the *'advice'* class**. Table 6 shows the results of advice classification for various LLMs.

	Advice			Not-Advice		
Model	P	R	F1	P	R	F1
GPT-4 (gpt-4-1106-preview)	**0.94**	0.62	0.75	0.45	0.88	0.60
GPT-4 (gpt-4-0613)	0.86	0.93	0.90	0.75	0.58	0.65
GPT-3.5 (gpt-3.5-turbo-1106)	0.83	0.07	0.13	0.27	0.96	0.42
Mistral (7B-Instruct)	0.85	0.64	0.73	0.40	0.69	0.51

Table 6: Zero-shot advice classification of performance of different LLMs.

We evaluate the zero-shot performance and do not tailor prompts specifically for each model. Instead, we write a simple prompt and use it for all the models. Figure 4 shows the structure of our prompt. The GPT-4 model achieved the highest precision of 0.94. While recognizing the potential for enhancing other models' performance through prompt optimization or the utilization of alternative prompting techniques (e.g., chain-of-thought, few-shot), we refrain from exploring these avenues due to the already high agreement observed between the human annotator and the GPT-4 model in the 'advice' class. This aspect could be the focus of a separate study.

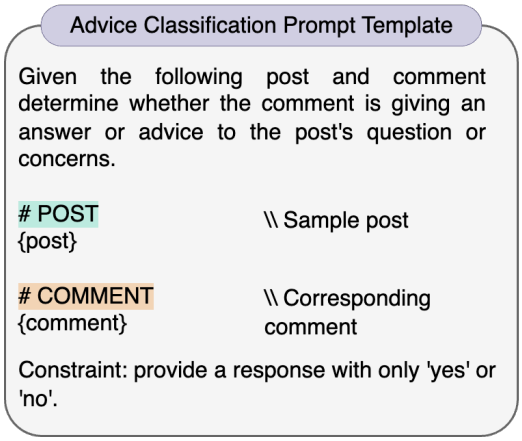

Figure 4: Advice classification prompt template.

Final Annotation: As the false positive is very low for *advice* class, we employed the GPT-4 model to identify advice samples. Out of 6,214 post-comment pairs, GPT-4 categorized 2,934 as advice. This advice set is utilized in the subsequent event argument annotation step.

Human Verification: To validate the labeling accuracy of GPT-4, we conducted a second-level verification. A human annotator manually reviewed 50 randomly selected samples labeled as 'advice' by the model. The human annotator confirmed advice labels for 49 of the 50 samples, resulting in a 98% precision for GPT-4 advice labeling.

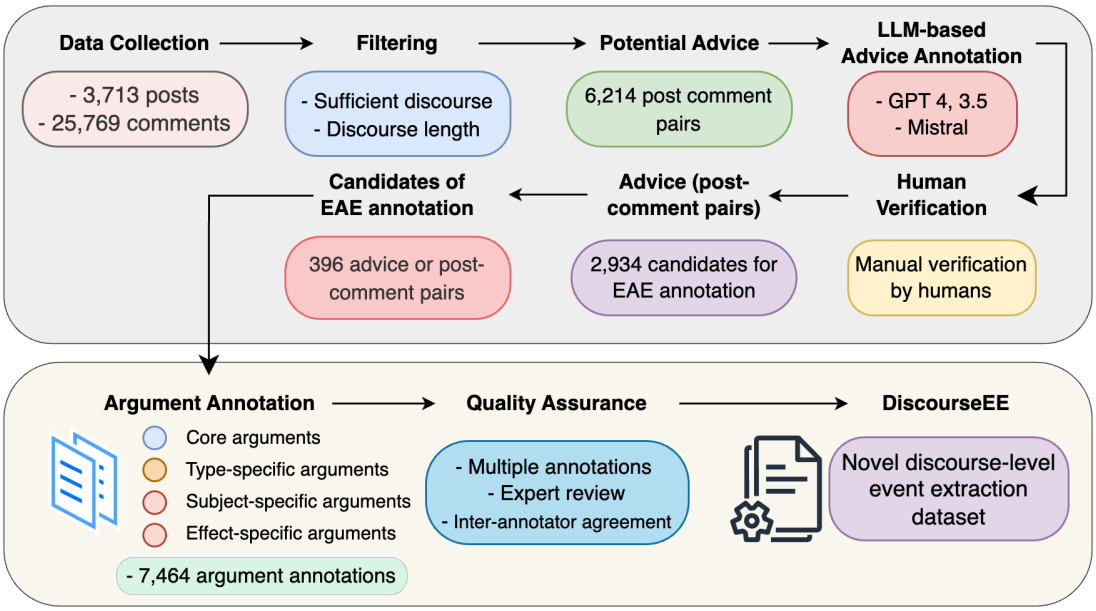

Figure 5: DiscourseEE development pipeline

C Annotation complexity

To achieve quality annotations, we held biweekly meetings with the annotators to address challenges and complexities. Major challenges annotators face are,

- **Misleading context:** Commenters often share long stories of their treatment journey when giving health advice. Finding the correct arguments from such descriptions can be challenging. For instance, multiple mentions of *'treatment dosage'* may make it difficult to differentiate between what is part of the current advice and what is simply mentioned about past conditions.
- **Domain-specific terms and noisy shorthands:** Annotators struggle with understanding domain-specific terms like *'Jumping off'* or *'cold turkey'* which refer to quitting or the intention of not taking medications. Additionally, the presence of shorthand makes annotation challenging, such as *'PWD'* for precipitated withdrawals and abbreviations like *'percocet'*, *'meth'*, *'oxy'* representing different substances.
- **Inconsistency in finding scattered arguments:** Arguments can be scattered throughout the document. For instance, for the core argument of *'tapering event'*, components like *'taper trigger,' 'current dosage,' 'taper start time'* may be dispersed all over. The chance of missing some parts of the arguments by annotators increased due to this dispersion.
- **Implicit arguments:** Annotators face difficulties identifying implicit arguments as they require a deeper understanding of the context. Such as the severity of an individual's experience can be *high, mild, low* or none. Understanding all psychophysical effects the individual is experiencing is crucial for selecting the severity when it is not explicitly mentioned.

We address these challenges to minimize annotation disagreements through interactive and iterative sessions, as well as multiple rounds of reviews.

D Dataset Statistics

Table 7 shows the number of explicit, implicit, and scattered arguments across core, type-specific, subject-effect types. 68.6% of the arguments are implicit or scattered, with only 31.4% being explicit. The core arguments are predominantly implicit or scattered. Implicit arguments are more common in subject-effect categories, while type-specific arguments have a higher proportion of explicit ones.

Figure 6 illustrates the distribution of type-specific and subject-effect argument annotations. For the 'taking moud' event, mentions of *medications, dosage*, and *manner* are frequent. For 'return to usage', people often mention their *substance use, current-medications, condition* and *trigger* of the

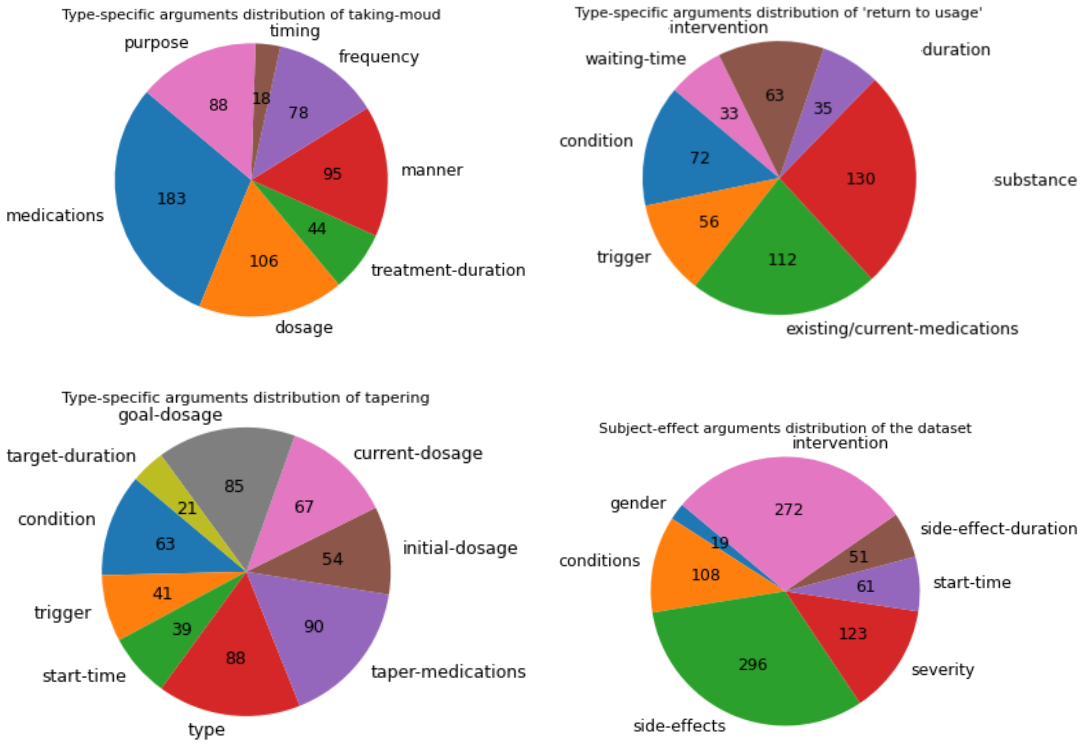

Figure 6: Distribution of type-specific and subject-effect arguments in the dataset.

	TM	RU	TP
Avg. #Words per argument			
– Core	4.46	4.64	4.23
– Type-specific	1.70	1.95	1.93
– Subject-effect	2.28	2.09	2.21
#Sample	178	118	100
– train	117	67	62
– dev	20	16	14
– test	41	35	24
Explicit, implicit, and scattered arguments			
#Explicit arguments			
– Core	86	69	53
– Type-specific	252	266	151
– Subject-effect	124	114	91
#Implicit arguments			
– Core	242	209	177
– Type-specific	321	190	346
– Subject-effect	193	147	145
#Scattered arguments			
– Core	183	139	92
– Type-specific	39	45	51
– Subject-effect	52	34	34

Table 7: Explicit, implicit, and scattered arguments distribution in DiscourseEE.

return to usage. In 'tapering', *taper-medications, intial-dosage, current-dosage,* and *type* of tapering are mostly mentioned. From subject-effect argument distribution, it is evident that people share more side-effect information (*side-effects, intervention, severity*) than personal information (*age, gender*). All these arguments provide crucial information for understanding or comparing health advice within social discourse.

E Models

We perform comprehensive experiments encompassing various methodologies, including transformer-based, instruction-tuned, and large language models. The details of each model are described in the following.

Transformer Models: We employ three transformer-based models to benchmark the event detection task. These include Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), robust BERT architecture trained with more training data for longer period (RoBERTa) (Liu et al., 2019b), and the model with permuted pre-training (MPNet) (Song et al., 2020).

Instruction-tuned Models: We choose FLAN-T5 (Chung et al., 2024) as the backbone model for instruction fine-tuning. It follows the standard T5 (Raffel et al., 2020) architecture and treats each task as a sequence-sequence problem. The model is fine-tuned on diverse task mixtures with

instruction-following objectives and is available in a wide range of sizes, from small (80M parameters) to UL2 (20B parameters). To explore the feasibility of our task with smaller language models and reduce computational costs, we experiment with the base (250M parameters) and large (780M parameters) models.

Large Language Models: We used five LLMs for the experimentation.

- **Gemma (7B)** (Gemma-Team, 2024) is an open-source language model trained on 6T tokens, following the architecture and training recipe of Gemini models (Gemini-Team, 2024). It outperformed similarly sized open-source models on 11 out of 18 text-based tasks. Gemma comes in two sizes, 2B and 7B parameters. We utilize the 7B version for our experiments.
- **Mixtral (8x7B)** (Jiang et al., 2024) is a sparse mixture of expert language models with the similar architecture of Mistral 7B (Jiang et al., 2023). In mixtral, each layer comprises 8 feedforward blocks or experts, enabling each token to access 47B parameters while using only 13B active parameters. Due to this architectural change, mixtral outperforms models with higher parameters (e.g., Llama-2, GPT-3.5) across several benchmarks.
- **Llama-3** is a state-of-the-art open-source language model pretrained and instruction-fined with 8B and 70B parameters. It builds upon the Llama-2 model (Touvron et al., 2023), incorporating significant changes such as grouped query attention (GQA) and an expanded tokenizer vocabulary. The 8B and 70B models outperform others within a similar parameter range.
- **GPT-4** (OpenAI et al., 2024) is the best-performing multimodal model that achieved state-of-the-art performance on various professional and academic benchmarks.

We employed the instruction-tuned version of all the models as it aligns better with our task. The HuggingFace inference strings for the open-source LLMs are Gemma (google/gemma-1.1-7b-it), Mixtral (mistralai/Mixtral-8x7B-Instruct-v0.1), and Llama-3 (8B) (meta-llama/Meta-Llama-3-8B-Instruct), Llama-3 (70B) (meta-llama/Meta-Llama-3-70B-Instruct). Additionally, we investigate the performance of the GPT-4 model via API (version gpt-4o-2024-05-13) calls.

F Implementation Details and Hyperparameters

Experimental details of our models are discussed in the following subsections.

F.1 Event Detection Models

The transformer-based models (i.e., BERT, RoBERTa, MPNet) used for event detection are sourced from the HuggingFace library. All the models are fine-tuned on our training set with batch size 8 and learning rate $2e^{-05}$. For each model, the version that achieves the best performance on the validation set is saved for final predictions on the test set.

Similar to transformer models, we also sourced the FLAN-T5 models from Hugginface. We use simple instructions for event detection. The input text template is `'Classify the following post-comment pair. [Post] [Comment]'` and the target text consists of corresponding event labels `'[Event labels]'`. Instructions sets are created from the training set. We fine-tuned base and large models for 3 epochs with a learning rate of $3e^{-4}$ and batch size 4. The max input length is set to 512 tokens, and the output length to 20 tokens.

We conduct open-source LLM experiments using LangChain[3] and HuggingFace. To generate event types with LLMs, we employ the template `'<instruction> <class details> <post> <comment>'`. Table 10 provides a sample prompt for event detection. This prompt template is utilized across all open and closed-source models.

F.2 Argument Extraction Models

For argument extraction, we experimented with three suites of models.

Extractive-QA: We implement the extractive-QA model using the HuggingFace pipeline with the BERT-base model (110M parameters) fine-tuned on the SQuAD dataset. The model is fine-tuned with a learning rate of $2e^{-05}$, a batch size of 8, for 3 epochs. We extract the argument for each role separately. During inference, we pass a question about the role along with the post-comment pair as the context. The input is formatted as `[CLS] Question [SEP] Post-comment pair [SEP]`.

[3] https://python.langchain.com/v0.1/docs/modules/model_io/prompts/

	Taking MOUD			Return to Usage			Tapering			
Model	C-A	TS-A	SE-A	C-A	TS-A	SE-A	C-A	TS-A	SE-A	Mean (EM_F1)
Extractive-QA	1.25	7.83	1.50	3.36	14.85	6.47	5.11	4.20	4.00	5.40
Generative-QA										
– FLAN-T5 (Base)	4.89	25.24	2.85	3.75	13.31	11.37	7.40	21.93	6.67	10.82
– FLAN-T5 (Large)	5.10	32.33	7.87	7.09	18.57	14.66	10.06	33.58	10.81	15.56
LLMs with Zero-Shot Description-guided Prompt										
Gemma (7B)	2.35	20.88	14.21	1.11	16.33	12.65	8.45	17.76	12.57	11.81
Mixtral (8x7B)	7.25	18.34	11.41	3.80	13.12	12.17	8.79	19.01	11.68	11.73
Llama-3 (8B)	2.34	28.88	9.79	4.42	19.05	13.86	8.19	25.88	15.24	14.18
Llama-3 (70B)	6.75	25.39	9.70	5.63	21.04	13.17	11.09	22.14	15.30	14.47
GPT-4	2.36	27.30	14.79	7.39	23.04	21.04	8.42	20.35	16.77	15.72
LLMs with Zero-Shot Question-guided Prompt										
Gemma (7B)	3.47	26.85	9.48	0.70	19.74	14.04	1.74	21.52	15.66	12.58
Mixtral (8x7B)	6.17	15.94	9.59	1.56	12.74	10.86	7.71	17.39	16.46	10.93
Llama-3 (8B)	1.85	21.72	6.99	4.92	18.88	11.77	5.81	25.75	14.31	12.45
Llama-3 (70B)	3.33	25.84	10.21	5.82	23.33	14.62	12.11	23.02	18.56	15.20
GPT-4	4.95	29.06	19.35	6.22	21.19	26.10	8.72	21.06	14.86	16.84

Table 8: Performance (avg. of 3 runs) of the models for event argument extraction across all argument types in exact match F1-score (EM_F1).

	Exact Match (Recall)		
Model	Explicit	Implicit	Scattered
Extractive-QA	20.67	0.40	0.0
Generative-QA			
– FLAN-T5 (Base)	26.54	10.02	1.61
– FLAN-T5 (Large)	33.33	14.72	5.91
LLMs with Zero-Shot Description-guided Prompt			
Gemma (7B)	26.74	7.43	0.89
Mixtral (8x7B)	34.77	6.54	2.86
Llama-3 (8B)	33.33	10.22	1.07
Llama-3 (70B)	40.02	6.95	3.94
GPT-4	37.96	10.77	3.22
LLMs with Zero-Shot Question-guided Prompt			
Gemma (7B)	30.86	8.17	1.25
Mixtral (8x7B)	31.06	7.56	2.68
Llama-3 (8B)	30.65	8.58	1.61
Llama-3 (70B)	41.15	8.45	1.97
GPT-4	39.81	11.86	3.04

Table 9: Performance (avg. of 3 runs) comparison of explicit, implicit, and scattered argument extraction in exact-match setting. Performance drop is significant in scattered arguments as these arguments are longer. Models generate different outputs with slight variation, which is not considered under exact match.

The output span is then decoded as the argument for the specific role.

Generative-QA: We use the HuggingFace (Wolf et al., 2020) Transformers library to develop the instruction fine-tuned generative-QA models. An instruction prompt is input to the model, which is fine-tuned to generate the argument for a role. Figure 7 illustrates the format of the instruction

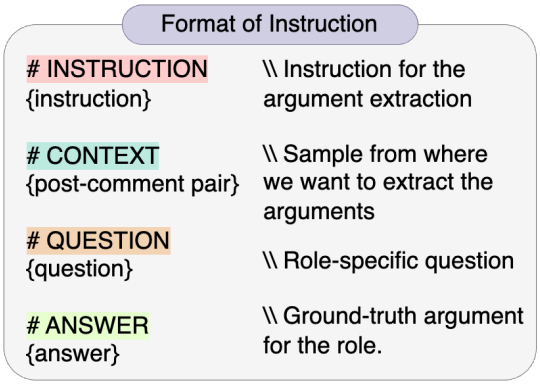

Figure 7: Instruction template for fine-tuning. See the sample instruction in table 10.

prompt. Here, post-comment pair and answers all the values are filled from the training set. The model is fine-tuned for 2 epochs with a learning rate of $3e^{-04}$ and a batch size of 8. The input length is fixed at 512 tokens while the output length is 128 tokens. Similar to the extractive-QA approach, the argument for each role is generated separately. We fine-tune both the FLAN-T5-base (250M) and large (780M) models with the same hyperparameters on Google Colab A100 GPU.

LLM-based Generation: Similar to event detection, we perform argument extraction experiments with LangChain and HuggingFace. We use instruction-tuned versions of the models as they align better with our task. To ensure the deterministic behavior of the model, we set the temperature value to 0.01 across all experiments with open-

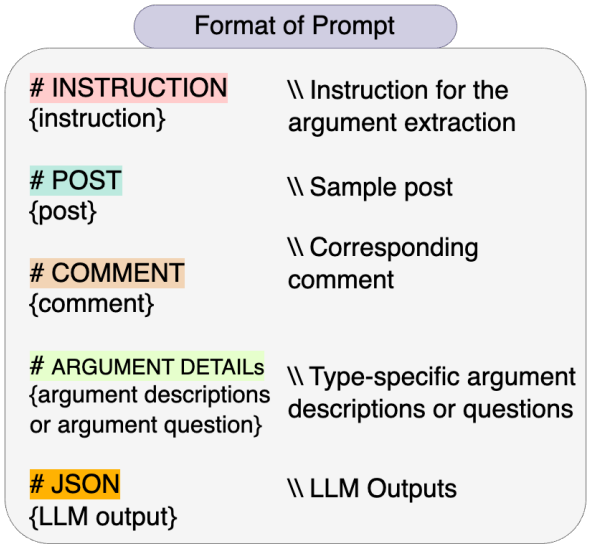

Figure 8: Generic argument extraction prompt template using LLMs.

source models since it only allows non-zero values. The temperature is set to 0.0 for GPT-4. We experimented with all models using both description-guided and question-guided prompts. The prompt template for argument extraction is illustrated in figure 8, and prompt examples are provided in table 10.

Prompt template for event detection	*#Instruction:* Classify the following post into 'taking-moud', 'return to usage', 'tapering' classes. *#Class Descriptions:* taking-moud: post related to medications for opioid use disorder (MOUD) regimen details. return to usage: post related to relapsing or using other substances during recovery. Such substance use can be attributed to recreational purposes or for self-medication (e.g., marijuana for sleep). tapering: post related to reducing the dose or frequency of MOUD and eventually quitting MOUD. *#Post*: <post> //post from the dataset. *#Comment*: <comment> //corresponding comment of the post. *#Output:* <LLM outupt> //LLM generated event types.
Instruction template for fine-tuning FLAN-T5 models for argument extraction.	*#Instruction:* Concisely extract the following argument from the post comment pair. Do not use more than 12 words to describe an argument. Return 'null' if any argument is not present. *#Context:* Post: I am a 42-year-old male with severe back pain. I haven't taken my 12 mg of suboxone since Thursday. My last opioid was 3 days ago. My nose has been runny for the last 2 days, and I feel like an 8/10. Will it kick in? Comment: Yeah it will kick in. Withdrawals are coming. Suboxone just has an extremely long half life which is why you are still feeling fine. It will catch up to you though. I definitely wouldn't recommend jumping off at 12mg! *#Question:* What are the tapering steps (drugs, start dosage, duration, goal dosage)? *#ANSWER:* have not taken 12mg of suboxone since Thursday.
Description-guided argument extraction prompt template for LLMs. This is for 'tapering' event 'type-specific' argument extraction.	*#Instruction:* Concisely extract the following argument from the post comment pair. Do not use more than 12 words to describe an argument. Return 'null' if any argument is not present. Return arguments in JSON format. *#Post:* I am a 42-year-old male with severe back pain. I haven't taken my 12 mg of suboxone since Thursday..... *#Comment:* Yeah it will kick in. Withdrawals are coming. Suboxone just has an extremely long half life which is why you are still feeling fine. ..... *#Arguments Descriptions:* condition: Describe the state or situations of the subject before tapering, trigger: Factors or events contribute to tapering, start-time: Start-time of tapering, type: Tapering type (self-tapering or prescribed tapering), taper-medications: Drugs/medications used during tapering, initial-dosage: Initial dosages of the drugs, current-dosage: Current dosage of the drugs, goal-dosage: Goal dosage the subject wants to achieve, target-duration: Duration to go from the start to the intended dosage or quit. *#JSON:*
Question-guided argument extraction prompt template for LLMs. This sample is for 'tapering' event 'core' type argument extraction.	*#INSTRUCTION:* Concisely extract the following argument from the post comment pair. Do not use more than 12 words to describe an argument. Return 'null' if any argument is not present. Return arguments in JSON format. *#Post:* I am a 42-year-old male with severe back pain. I haven't taken my 12 mg of suboxone since Thursday..... *#Comment:* Yeah it will kick in. Withdrawals are coming. Suboxone just has an extremely long half life which is why you are still feeling fine. ..... *#Arguments Questions:* subject/patient: How can you describe the individual or patient involved?, effects: What are the outcomes or side effects of the treatments?, tapering-event: What are the tapering steps (drugs, start dosage, duration, goal dosage). *#JSON:*

Table 10: Sample instruction and prompt used in the argument extraction experiments. To illustrate the difference in instruction, description-guided, and question-guided prompts, we used the same post-comment pair. For fine-tuned models, we extract arguments for each role separately. Thus, for each sample, we performed approximately 19 inferences (one for each role) based on the event type. To reduce inference costs for LLMs, we adopt a divide and conquer prompt approach (discussed in 5.2) . We extract all arguments of a specific type (i.e., core, type-specific, subject-effect) together, reducing the number of inferences from 19 to 3. Finally, we merge the outputs to obtain the predictions.

Type	Name	Description
Taking MOUD (TM)		
Core Arguments	Subject/Patient	Describe the individual or patient involved.
	Treatment	Describe the treatments prescribed or undergoing.
	Effects	Describe the outcomes or side effects of the treatments.
Type-specific Arguments	Medications	Drugs/medications used in the treatment.
	Dosage	Current or previous dosage of the medications.
	Treatment duration	Duration of taking the medication.
	Manner	Manner of taking medication orally/ sublingually/ as injections.
	Frequency	Frequency of taking medication (per day, week, month)
	Timing	Timing of taking medication (night, morning, etc.)
	Purpose	Purpose of taking this medication.
Return to Usage (RU)		
Core Arguments	Subject/Patient	Describe the individual or patient experiencing the return to usage.
	Return to usage event	Describe the occurrence of taking or using addictive substances.
	Resuming MOUD after RU	Describe the events the subject is doing or intends to follow after the last return to usage dose.
	Effects	Describe the outcomes or side effects of the return to usage.
Type-specific Arguments	Condition	Describe the substance use history/disorder from which the subject had previously recovered/ was in the process of recovery
	Trigger	Factors or events contribute to return to usage.
	Existing/Current medications	Medications used before the return to usage.
	Substance used in RU	Substance was used in the return to usage.
	RU duration	Duration of the return to usage.
	RU intervention	Measures are taken to address or prevent the return to usage
	Waiting time	Waiting time after the last dose of return to usage.
Tapering (TP)		
Core Arguments	Subject/Patient	Describe the individual or patient involved.
	Tapering Event	Describe the tapering steps (drugs, start dosage, duration, goal dosage).
	Effects	Describe the outcomes or side effects of the tapering.
Type-specific Arguments	Taper condition	Describe the state or situations of the subject before tapering.
	Trigger/motivation /cause/reason	Factors or events contribute to tapering
	Taper Type	Tapering type (self-tapering or prescribed tapering)
	Taper Drugs/ Medications	Drugs/medications used during tapering
	Initial dosage	Initial dosages of the drugs.
	Current dosage	Current dosage of the drugs.
	Goal dosage	Goal dosage the subject wants to achieve.
	Start time	Start-time of tapering
	Target Duration	Duration to go from the start to the intended dosage or quit.
Taking MOUD / Return to Usage / Tapering		
Subject-specific Arguments	Age	Age of the subject/patient
	Gender	Gender of the subject/patient
	Pre-existing or comorbid conditions	Pre-existing or co-morbid conditions of the subject/patient
Effect-specific Arguments	Side Effects	Side effects the subject is experiencing or expects to experience
	Severity	Severity of the side effects
	Start time	Start time of experiencing the side effects
	Duration	Duration of the side effects
	Intervention	Measures are taken to address or reduce side effects

Table 11: Details of the argument roles for each event type in the DiscourseEE dataset. Subject-specific and effect-specific arguments are the same across all event types.