Generative Approaches to Event Extraction: Survey and Outlook

Étienne Simon[*1] Helene Bøsei Olsen[*1] Huiling You[1]
Samia Touileb[2] Lilja Øvrelid[1] Erik Velldal[1]
[1] University of Oslo, [2] University of Bergen

Abstract

This paper aims to map out the current landscape of generative approaches to the task of event extraction. In surveying the emerging literature on the topic, we identify the distinctive properties of existing studies and catalogue them to build a comprehensive view of the various techniques employed. Finally, looking ahead, we argue for a new generative formulation of event extraction, allowing for a better fit between methodology and task – a proposal that could also pertain to many other traditional NLP tasks currently based on annotations of text-spans.

1 Introduction

Event extraction is one of the core applications in Natural Language Processing (NLP), aiming to create structured representations of events described in unstructured text. The task revolves around the identification and categorisation of predefined types of events within texts. This is typically broken down into identifying and categorising so-called event triggers and their respective arguments, along with their relevant properties and relationships, such as time, location, and participants.

Recently, generative language models have seen widespread uptake across many subfields of NLP, and event extraction is no exception. Generative approaches to event extraction sometimes deviate from the traditional way of identifying and categorising events and their arguments, introducing new opportunities and challenges with respect to both training and evaluation.

This paper provides an overview of the current landscape of generative approaches to event extraction by focusing on a representative set of techniques across different dimensions. We survey how the task of event extraction is approached across the range of decoder-only and encoder–decoder models with regards to generating the extracted event fields as natural language – as opposed to the traditional sequence labelling or boundary identification approaches. For readers seeking a broader overview of event extraction approaches, the surveys by Xiang and Wang (2019), Liu et al. (2021) and Li et al. (2022) can be explored.

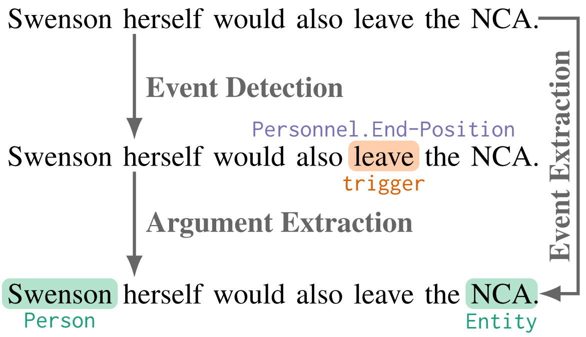
Swenson herself would also leave the NCA.
Event Detection
Personnel.End-Position
Swenson herself would also leave the NCA.
trigger
Argument Extraction
Swenson herself would also leave the NCA.
Person
Entity
Event Extraction

Figure 1: The two subtasks of event extraction on a sample from ACE. The event type is shown in blue over the trigger highlighted in orange. The event arguments are highlighted in green with their role specified under each argument.

Event extraction (EE) is traditionally approached as a sequence labelling problem. The annotations identify specific text spans that highlight event triggers with their associated arguments. This leads to the task being broken up into two parts as shown in Figure 1: (1) *event detection* (ED) where event triggers are identified and the event is categorised into a type. An event trigger typically corresponds to the word(s) in the text that most clearly describes an event. In the example of Figure 1, "leave" evokes an "`End-Position`"-type event. (2) *event argument extraction* (EAE) where event arguments are identified along with their role. The role is the semantic relationship of the argument to the event. In the example of Figure 1, "Swenson" is identified as relevant to the `End-Position` event as the `Person` leaving. When two distinct models are used to tackle each subtask, the approach is referred to as a

[*] Equal contribution.

Proceedings of the Workshop on the Future of Event Detection (FuturED), pages 73–86
November 15, 2024 ©2024 Association for Computational Linguistics

pipeline, while approaching both subtasks together is denoted a *joint* approach. Only a third of the models we survey perform ED. The two subtasks are also evaluated separately. However, the argument extraction scores are not always comparable as some models use a pipeline setup where the gold trigger is used in the evaluation of argument extraction, while other models only work in a joint setup where the arguments can only be extracted together with the trigger (Peng et al., 2023). We therefore elect to not include reported results as comparison can be misleading. Before we dive into the description of the different modelling approaches, we start by discussing some of the most prominent event extraction datasets.

2 Datasets

In this section, we discuss the event datasets most commonly used across the different generative approaches we assess in Section 3. While some of the datasets cover multiple languages, most primarily focus on English language sources.

The highly influential Automatic Content Extraction (ACE) program released manual event annotations for text spans at the sentence-level, also including rich information about entities, temporal expressions, and relations between entities. The event annotation in the ACE tradition has become a *de facto* standard for the evaluation of event extraction systems in the field of NLP. The 5th iteration of the dataset, ACE 2005 (Doddington et al., 2004), consists of broadcast transcripts in addition to newswire and newspaper texts. It provides manual annotation for entities, relations, and events for joint evaluation of multiple information extraction tasks in multiple languages (Arabic, Chinese, and English) at the sentence level. The ACE dataset is annotated for 8 general event types (e.g. `Life`, `Conflict`, `Transaction`), along with 33 subtypes (e.g. `Conflict.Attack`), and 22 argument roles (e.g. `Attacker`, `Agent`, and `Recipient`). The English version of the dataset comprises 599 documents. Depending on the pre-processing approach, ACE features two main variants, where ACE covers only events with single-token triggers, and ACE+ keeps all events with either single- or multi-token triggers. Four F1 scores are usually reported on ACE: the trigger identification, the trigger classification into an event type, the argument identification, and the argument classification into a role.

The evaluation of ACE and similar datasets is structured primarily for sequence labelling models. It typically involves comparing the predicted position offsets (specific locations of event mentions in the text) with the corresponding correct offsets. Consequently, if a name appears multiple times within a sentence, only one of those occurrences is considered correct.[1] Since generative models only extract and generate out-of-context surface forms without incorporating position offsets, evaluating them on datasets like ACE may give these approaches an unfair advantage. The current best practice for generative approaches is to search for the text generated by the model in the input text, transform the output to offsets to simulate a sequence labelling model, and subsequently evaluate it as such.

More recently, the ERE annotation effort (Entities, Relations, and Events, Song et al., 2015) has contributed both data and annotation guidelines for event extraction purposes. The ERE effort has evolved from the Light ERE to Rich ERE datasets, advancing from simple ACE-based annotations to more complex handling of entities and events, ultimately enabling document-level event co-reference. The ERE effort covers English, Spanish and Chinese documents from discussion forums, newswire, and proxy sources. The Rich ERE extends the annotation scheme of ACE, covering 9 main event types and 38 event subtypes. In Light ERE, only asserted events are annotated (events that have occurred), with each event trigger linked to a single event. In contrast, Rich ERE allows for event triggers to be annotated for multiple events and includes annotations for event modality, capturing events that did not actually occur.

Another sentence-level event dataset is MAVEN (MAssive eVENt detection dataset, Wang et al., 2020). It aims to alleviate problems of data scarcity and low coverage and contains 111 611 distinct events across 4480 human-annotated documents in total, corresponding to event-related articles from English Wikipedia. It comprises 168 hierarchically organised event types derived from FrameNet (Baker et al., 1998), intended to cover general-domain events.

Li et al. (2021) introduce a document-level annotated dataset based on English Wikipedia articles and their referenced news articles called WIKIEVENTS. While only containing 246 documents

[1] The ACE corpora include coreference information. However, it is not an established part of the standard formulation when evaluating the event extraction task.

with 8544 sentences, the dataset serves as an essential benchmark for event extraction systems beyond the sentence-level. Each document is annotated with event types, event mentions (triggers and arguments), and co-references across sentences, even in sentences lacking an explicit event trigger. Annotating co-references enables a fairer evaluation of generative models, as an extracted argument is considered correct if the model generates any co-reference of the gold argument. In WIKIEVENTS parlance, these are referred to as coref scores. The annotators also aimed to annotate the most informative event mention, giving precedence to name mentions over nominal mentions rather than focusing solely on the mention closest to the trigger word. This allows for another evaluation mode for WIKIEVENTS termed informative argument extraction, where models are evaluated on their ability to extract the most informative argument mention. The annotations of the dataset resemble ACE, but expand the number of sub-events from 33 to 67 following the KAIROS ontology.[2] Additionally, WIKIEVENTS has a more fine-grained event-type hierarchy. For instance, whereas ACE identifies the event type and subtype such as `Conflict.Attack`, WIKIEVENTS introduces event types at three levels, such as `Conflict.Attack.DetonateExplode`.

In recent years, the fourth Message Understanding Conference (MUC-4, Sundheim, 1992) dataset has resurfaced in research on document-level event extraction. The dataset is based on English newswire provided by the Federal Broadcast Information Services. It is annotated with the event types `Arson`, `Attack`, `Bombing`, `Kidnapping`, `Robbery`, `Forced work stoppage`, covering political conflicts in Latin America. MUC-4 contains 1700 documents, which may be associated with zero or more events of each type. Moreover, the event type is associated with a template, each with the same set of 24 argument roles to be filled with either a numeric value, a categorical value, a text string, or a canonical form extracted or derived from the text. However, beyond event type classification, most recent works on the dataset are based on a simplified set of template slots restricted to five argument roles, where all have text string values that can be directly extracted from the source document (Du et al., 2021a,b; Gantt et al., 2024). Given that a large proportion of the documents are linked to empty templates, indicating the absence of relevant events, the ED task is important for MUC-4.

Some of the models discussed in this survey employ additional datasets alongside those outlined above. RAMS (Ebner et al., 2020) covers 9124 events from news articles and is annotated in a 5-sentence window around each event trigger. PHEE (Sun et al., 2022) is a biomedical domain-specific dataset focused on drug safety, consisting of nearly 5000 sentences extracted from public medical case reports. Finally, CASIE (Satyapanich et al., 2020), consisting of 5000 news articles, with 1000 of these annotated on the sentence-level for cyber-attack events.

3 Models

The majority of the models we survey follow a similar pattern as shown in Figure 2: the input text is fed to an encoder–decoder transformer that is fine-tuned to generate a representation of the events conveyed in said text. Most of them can be divided into one of two groups according to how the events are represented in the generated text. To structure this survey, we first consider the representation of events in the output of the model; two main approaches exist: i) either an event is represented using a formal structure template in line with Text2Event (Lu et al., 2021), or (ii) the event is represented using a natural language template in line with BART-Gen (Li et al., 2021). We present these two distinct approaches in two separate sections. A classification of all models is also given in Figure 3. Note that the organisation of this survey results from the fact that most of the models within the scope of this work build upon Text2Event and BART-Gen. However, this structure does not necessarily reflect a deep fundamental difference between the two sections.

The first model using a generative transformer to address EE falls partly outside this dichotomy. **TANL** (Paolini et al., 2021) introduces an ED model and an EAE model using T5 (Raffel et al., 2020). However, TANL does not focus solely on event extraction and can be trained on multiple information extraction (IE) tasks (named entity recognition, coreference resolution, etc). This is a recurring pattern in the papers we survey; different IE tasks are often similar enough that a single architecture can be reused. TANL goes one step further by simultaneously training on multiple tasks before

[2] https://www.darpa.mil/program/knowledge-directed-artificial-intelligence-reasoning-over-schemas

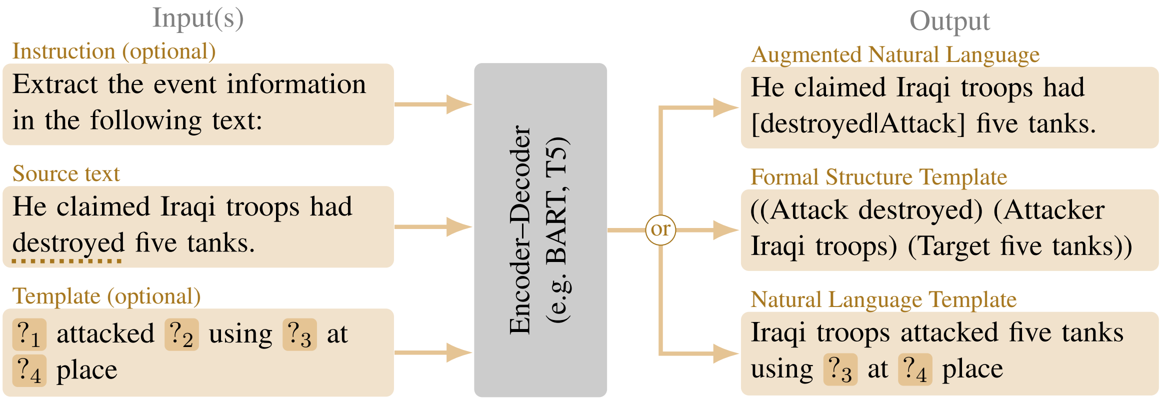
Input(s)
Instruction (optional)
Extract the event information in the following text:
Source text
He claimed Iraqi troops had destroyed five tanks.
Template (optional)
?1 attacked ?2 using ?3 at ?4 place
Encoder–Decoder
(e.g. BART, T5)
or
Output
Augmented Natural Language
He claimed Iraqi troops had [destroyed|Attack] five tanks.
Formal Structure Template
((Attack destroyed) (Attacker Iraqi troops) (Target five tanks))
Natural Language Template
Iraqi troops attacked five tanks using ?3 at ?4 place

Figure 2: Schema of a standard generative event extraction model. On the left-hand side are common features given to the model as input. Some models rely only on the source text being present. A trigger word can be marked in the source text if the task being worked on is argument extraction. The inputs are generally given to an encoder–decoder model, which then generates a representation of the event. Three examples of possible outputs are shown on the right-hand side. TANL uses augmented natural language, while models based on generating a formal structure or natural language templates are described respectively in Sections 3.1 and 3.2 respectively. Note that some models do not follow this general pattern (see for example, QGA-EE).

evaluating event extraction. However, this setup is not common and makes direct comparisons of results difficult. TANL is also unique among generative models in that it is the only one that relies on offset-based annotation for training. This is because it uses an augmented text representation where the input text is generated in the model's output together with the extracted information. This can be seen in Figure 2, where the first box on the right showcases an example of TANL's output for event detection. Subsequent models only generate the structured information without generating the whole sentence. In this regard, TANL is more directly comparable to a sequence tagging scheme. For example, if the word "destroyed" appeared twice in the given example, the model would be able to distinguish between the two and tag only the relevant one.

3.1 Formal Structure Template

The first popular approach to represent events following TANL is to discard the source text from the output and keep only what is evaluated: the event structure. The exact structure used differs across models and needs only to be able to encode an associative dictionary between role and arguments (e.g. S-expression, JSON).

This approach was pioneered by **Text2Event** (Lu et al., 2021), which jointly models the ED and EAE subtasks. They use a T5 encoder–decoder model, where the encoder is given the source sentence alone and the decoder is supervised by an S-expression, as illustrated by Figure 2. The output of the model is therefore a mix of labels (event type, argument roles), structure tokens (separating the events and arguments), and input tokens (the extracted trigger and arguments) following a strict ordering. To enforce this ordering (e.g. an argument role must be followed by input tokens, then by a structure token), Text2Event introduces constrained decoding: the output vocabulary is restricted to valid tokens at each step (e.g. the softmax is only applied over tokens appearing in the input if an argument role was just generated). They show that this is particularly helpful with small training sets. Their ablation study also includes curriculum learning and shows that using natural language tokens for argument roles is preferable to arbitrary tokens. While TANL is often used as a baseline, it was not used as a basis for future work. Conversely Text2Event prompted a series of follow-up models bringing incremental improvements. For example, **Set Learning** (Li et al., 2023) improves Text2Event by attempting to enforce permutation-invariance of its output. In Text2Event a sample is supervised with a sequence of event arguments in an arbitrary order, whereas Li et al. (2023) supervises every sample with multiple orderings of the arguments and events.

The **KC-GEE** model (Wu et al., 2023) also follows most of the Text2Event architecture but uses prefix-tuning to enhance the performance on the task. Specifically, schema information – what are the possible event types and roles – is used to condi-

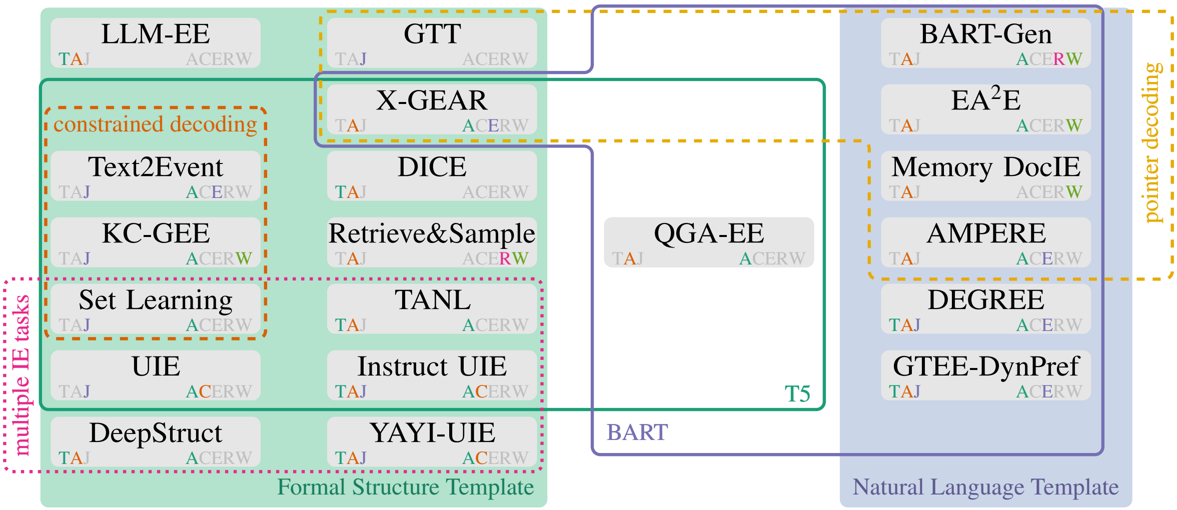

Figure 3: Overview of the models covered by the survey. The two shaded blocks correspond to the type of event representation generated by the models. The T5 and BART boxes indicate the backbone LLM of the different models. The constrained and pointer decoding boxes envelop models that do not freely generate from their entire vocabulary. Instead, their output is restricted either through a masking mechanism (constrained decoding) or by similarity with the input (pointer decoding). The multiple IE tasks box groups together models that are also used for other information extraction tasks such as named entity recognition and relation extraction. For each model, the lower left letters indicate which subtasks are tackled, while the lower right letters indicate the datasets they are evaluated on. The subtasks are: Trigger extraction & classification (ED), Argument extraction, and Joint trigger and argument extraction. A model can be used in a pipeline setup if it is marked for both trigger and argument extraction subtasks. The listed datasets are: ACE, CASIE, ERE, RAMS, and WikiEvents. A slightly expanded version of this figure is presented in the appendix as Table 1.

tion both the encoder and the decoder through vector prefixes. This enables the model to generalise to unseen event types in a zero-shot setting. Additionally, KC-GEE targets document-level event extraction and incorporates a cross-attention mechanism to effectively process entire documents. KC-GEE achieves notable performance gains compared to Text2Event on WikiEvents, and in the zero-shot setting.

Retrieve&Sample (Ren et al., 2023) focuses solely on document-level event argument extraction with retrieval-augmented generation (RAG). Specifically, they first retrieve top-k potentially helpful documents from the training corpus. The helpfulness of a document is computed using a T5-encoder-based siamese network from the input text and event schema. The retrieved documents are fed as an additional input to the model (Figure 2) together with the input document and schema information. They also explore two other retrieval strategies: context-consistent retrieval and schema-consistency retrieval. As it is designed for document-level extraction, the model is evaluated on RAMS and WikiEvents.

Lu et al. (2022) introduce **UIE** as a unified information extraction framework via text-to-structure generation. Like TANL, the authors aim to tackle multiple IE tasks, however this is done with a formal structure similar to Text2Event. UIE formalise a unified structure for encoding different information elements (i.e. entities, relations, events), dubbed structural extraction language. The authors argue that any information extraction task can be decomposed into two atomic operations: *spotting* and *associating*; where the former operation locates relevant text spans and the latter connects the spans with a task-specific schema. The task to perform is indicated with a prefix referred to as *structural schema instructor*. For event extraction, this prefix contains the full dictionary of possible event types or roles, depending on the subtask. UIE is developed by the same team as Text2Event, and can be considered as its TANL-inspired generalisation. In particular they use an IE-specific pre-training that removes the need for constrained decoding. ACE, CASIE and PHEE are used to evaluate the models performance on the EE task. The UIE experimental setup composed of multiple IE tasks is subsequently re-used by other works we present in the next paragraphs. In parallel, Wang et al. (2022) introduce **DeepStruct**, a similar text-to-structure model which also addresses multiple IE tasks. However, they use GLM (Du et al., 2022b), a decoder-only transformer as a backbone, and only use ACE for evaluation.

Inspired by instruction tuning, Wang et al. (2023a) propose **InstructUIE** as a unified information extraction framework for multiple IE tasks in line with UIE. More specifically, all IE tasks are reformulated into the task of natural language generation with expert-designed instructions, which include a description of the output format (e.g. Output format is "type: trigger"). InstructUIE features joint training of multiple IE tasks on a collection of 32 datasets by creating a unified and consistent label set based on semantics, thus benefiting from cross-task knowledge sharing and more training data. Although InstructUIE is trained to extract the trigger and arguments jointly, it is only evaluated in a pipeline fashion on the same datasets as UIE.

Further extending instruction tuning, Xiao et al. (2024) propose **YAYI-UIE** as an end-to-end universal information extraction framework. Xiao et al. (2024) employ a two-step instruction tuning procedure: first, real-life dialogue data are used to enhance the model's capacity to understand human language instructions; second, the model is instruction fine-tuned for IE tasks on the InstructUIE datasets extended with Chinese-language datasets – in particular DuEE (Li et al., 2020b; Han et al., 2022) for event extraction. The instruction and output setup is somewhat similar to InstructUIE, except that a JSON-based format is used for the event structure. Similarly to DeepStruct, YAYI-UIE is based on a decoder-only model. They use Baichuan2 (Yang et al., 2023) as a backbone model, which is pre-trained using RLHF (Christiano et al., 2017) on English and Chinese data. YAYI-UIE achieves competitive results on the EAE subtask on the UIE experimental setup; the authors showcase in their ablation study the effectiveness of using real-life dialogue data to aid the model in understanding human instructions.

A few works evaluate decoder-only large language models (LLMs) for EE in a zero or few-shot fashion (Wang et al., 2024, 2022; Xiao et al., 2024; Wei et al., 2024), however outside of DeepStruct and YAYI-UIE, these efforts tend not to involve any fine-tuning. Worth noting is the work of Chen et al. (2024) that we refer to as **LLM-EE**. It sets out to assess the value of using pre-trained LLMs for EE, experimenting with a wide variety of different strategies. In a first suite of experiments, they prompt pre-trained LLMs to extract event information directly. Using ACE and MAVEN for evaluation, the LLMs tested include PaLM (Chowdhery et al., 2022), GPT-3.5-Turbo, and GPT-4 (OpenAI, 2024). Chen et al. (2024) report experiments for several different configurations; zero-shot and one-shot approaches, including both joint and pipeline strategies for the subtasks of ED and EAE, in addition to extraction of multiple events, for all event types simultaneously and individually. However, the results show that LLMs fall short of fine-tuned supervised approaches as was already shown by Gao et al. (2023). In a second suite of experiments, Chen et al. (2024) prompt the LLMs to generate annotated examples, aiming to improve the performance of fine-tuned models by augmenting the training data. This is motivated by the problems of data scarcity and class imbalance seen in many common datasets where certain low-frequent event types have very few annotated examples. The selection of models used for fine-tuning to evaluate the data augmentation comprises generative approaches like Text2Event discussed above. The results show that training on the augmented data yields a modest but consistent improvement in F-score (due to an increase in precision at the slight recall cost). An obvious avenue for future work left unexplored by Chen et al. (2024), is to further instruction fine-tune the LLM itself on EE specifically. Moreover, the context size of current LLMs would likely make them better positioned for document-level EE, rather than the sentence-level analysis required by datasets like ACE and MAVEN. Some works explore some specific characteristics of LLMs for EE, for example Code4Struct (Wang et al., 2023b) look at the possibility of transfer learning between python code and event structure using code-imitation prompts for few-shot event extraction. TISE (Fu et al., 2024) extends this by designing a method to select appropriate samples for the in-context learning prompts.

As described in Section 2, the template-filling dataset MUC-4 has reemerged in recent EE research. The **GTT** framework introduced by Du et al. (2021b) is one of the pioneering efforts in building an end-to-end generative model for the task of template filling, transforming it into a sequence generation problem. Although it is an encoder-only model, we include it in our survey for its seminal role. Extending the role filler entity extraction system GRIT (Du et al., 2021a), the framework relies on BERT with a partially-causal attention mask. Word prediction is done with a dot-product pointer selection mechanism to restrict output word predictions to the input vocabulary. The input includes a list of possible event types and structure tokens so that they can be generated, while the output is based on a formal structure template with a fixed set of (unlabelled) roles. In summary, GTT shows strong similarities with Text2Event, yet with some differences due to the use of BERT with a partial causal attention mask instead of an encoder–decoder. Compared to similar non-generative models, Du et al. (2021b) find that GTT performs better on MUC-4 documents with multiple events.

Some generative EE models focus on more specific problems. For example, **DICE** (Ma et al., 2023) is a T5-based model focused on the clinical domain, introducing a dataset alongside a Text2Event-like EAE model and a DEGREE-like ED model (described in the next section). Similarly, while most efforts focus on monolingual event extraction, Huang et al. (2022) explore zero-shot cross-lingual argument extraction on ACE and ERE using language-agnostic templates. They propose **X-GEAR** (Cross-lingual Generative Event Argument extractoR), which, given an input sentence, the trigger, and a type-dependent template, replaces the placeholder in the template either by generating a token or directly copying a token from the source text. The copy mechanism, adapted from See et al. (2017), conditions the generation of a token on a weighted sum of two distributions: the vocabulary distribution from the pre-trained mT5 model, serving as the backbone, and the copy probability derived from the cross-attention weights, which allows for directly copying tokens from the input sequence. Although X-GEAR is primarily developed for cross-lingual applications, it demonstrates strong performance in argument classification when both the source and target languages are English. While multiple studies (Paolini et al., 2021; Lu et al., 2021; Ren et al., 2023) highlight the benefit of using natural language for role labels in the generated template, X-GEAR conducts an ablation study showing that this approach does not generalise to cross-lingual settings.

3.2 Natural Language Template

Using natural language labels for event types and roles is expected to improve performance in the standard setup, as it allows models to leverage the LM pretraining of the backbone transformer (commonly BART). However, these architectures still use a non-natural formal structure to delimit different arguments. An alternative to this approach is to use a natural language template to structure the event as is shown on the right of Figure 2. We describe these approaches in what follows.

The first model of this type is **BART-Gen** (Li et al., 2021), a document-level EAE model. Argument extraction is framed as a conditional generation task, using a BART encoder–decoder model (Lewis et al., 2020). The output generated follows a predetermined natural language template given by the event ontology. The templates are specific to each event type and are also given in the input with special tokens in lieu of arguments. This allows BART-Gen to use a pointer-like mechanism for generation: the vectors at the output of BART-Gen are compared with the input embeddings, and the model then generates the token with the highest similarity, ensuring that all generated tokens appear in the input. Additionally, clarification statements in the form of type statements (e.g. `<arg>` is a Person), are included to avoid mismatches in entity types for arguments, and are used to re-rank the output sequences. A distinct trigger identification and classification model is introduced, as BART-Gen serves solely as an argument identification and classification system. However, this event detection model is not generative.

Zeng et al. (2022) introduce **EA^{2}E** (Event-Aware Argument Extraction) to solve document-level argument extraction by incorporating explicit event–event relations into an iterative inference process. Building upon BART-Gen (Li et al., 2021), the task is formulated as conditional generation, filling the argument placeholders of a pre-defined template. Moreover, event–event relations are also exploited by labelling the arguments of previously extracted events in the input. This allows the model to learn regularities, such as an entity previously extracted as a `Defendant` being more likely to be the `Perpetrator` in attack events. EA^{2}E performs this in an iterative fashion: first, the model generates the result for each target trigger, and then the predicted results will be used to augment the context for a second extraction. Evaluated on ACE and WikiEvents, EA^{2}E achieves advantageous results compared to previous works, such as BART-Gen. Du et al. (2022a) present a similar model evaluated on WikiEvents alone. Dubbed **Memory DocIE**, their approach takes as input a natural language template and a document, augmented with the most similar event already extracted from the document, where the latter is intended to act as a "document memory store". Event similarity is computed as the cosine between S-BERT embeddings of the filled event templates. Furthermore, all possible pairs of event roles are checked to mark incompatibilities, e.g. the `jailer` slot of an `arrest` event, is unlikely to be filled by the `attacker` of an `attack-detonate` event. The resulting constraints are enforced by masking incompatible tokens when generating arguments.

Hsu et al. (2022) propose the **DEGREE** model, targeting low-resource event extraction. While DEGREE still follows BART-Gen in that it uses BART to fill in a natural language template, it differs in how the event extraction task is approached. BART-Gen requires the event type to be known in order to select the appropriate template to be filled since the event type is traditionally extracted together with the trigger. In contrast, DEGREE still uses event-type-specific templates, but initiates them with "Event trigger is `<trigger>`" thus, it is able to perform trigger identification together with argument extraction given the event type. However, DEGREE is also trained to classify the event type. This is done by supervising the models with every possible template such that negative templates leave the `<trigger>` placeholder as-is in the output, while the correct templates would replace it with the trigger word. This means that all samples must be run through BART with all possible event templates during inference. This allows DEGREE to be used both in joint and pipeline settings. Furthermore, compared to BART-Gen, the input is extended with event type descriptions, such as "The event is related to conflict and some violent physical act.", and event keywords that are semantically similar to the event type. Compared to other generation-based models such as BART-Gen (Li et al., 2021), Text2Event (Lu et al., 2021), and TANL (Paolini et al., 2021), DEGREE shows comparable or inferior performance on sentence-level datasets. However, DEGREE's strength lies in low-resource settings, where it achieves significantly better performance even when trained on just 1% of the data.

Following DEGREE, Liu et al. (2022) introduce **GTEE-DynPref**, an approach using BART for conditional generation while attempting to ease event typing in the model's input. Usually, DEGREE's input is event-typed in two ways: through a type instruction "Event type `Meet`" and the natural language template. GTEE-DynPref replaces the type instruction with a vector representation similar to that of KC-GEE. Compared to DEGREE, an additional embedding matrix is used to associate type instruction prefix vectors to each event type. Each sample is associated with a distribution over event types using BERT. This distribution defines a convex combination of prefix vectors that are used in substitution to static type instruction. Since the type information is still enforced through the template, the model relies on training with negative event types. A 3-step curriculum learning approach used to bootstrap the type instruction embeddings further increases the complexity of the training procedure. Still, Liu et al. (2022) report competitive results on their evaluation datasets, ACE and ERE.

Hsu et al. (2023) introduce **AMPERE**, which also extends DEGREE by adding a dynamically generated prefix. This prefix incorporates structured information from abstract meaning representation (AMR) of the input passage. The AMR graph is encoded into prefix vectors using a BART-based AMR parser called SPRING (Bevilacqua et al., 2021). They show that explicit semantic structure from AMR aids event argument extraction. Compared to DEGREE, AMPERE injects AMR prefixes both into the encoder's self-attention blocks and into the decoder's cross-attention blocks. Additionally, they re-introduce a copy mechanism previously discarded by DEGREE but condition it with regularisation to encourage more frequent copying.

3.3 Iterative Question-Answering Approaches

In recent years, several efforts have approached EE as a Question-Answering (QA) task (Du and Cardie, 2020; Li et al., 2020a; Lyu et al., 2021). As a recent and generation-oriented study within this framework, Lu et al. (2023) propose the **QGA-EE** model for argument extraction, consisting of a question generation model (QG), and a question answering model (QA). Unlike models such as BART-Gen, which uses fixed templates for each event type, the sequence-to-sequence QG model generates context-aware questions tailored to the input sentence and the argument roles. A series of questions is generated for each sample, one for each role, each depending on the already extracted arguments. In order to generate the questions, the model is trained on manually created templates for each role in the ACE ontology, such as "Who was the attacking agent?" and "Who attacked `<target>`?". The QA model is trained with all possible questions as inputs and generates the answer strings corresponding to the role questions. The extracted arguments are then cross-checked with the input sentence, retaining only those that match perfectly. The authors explore the use of both BART and T5 architectures as the backbone for the QA model, finding that T5 yields better performance.

4 Summary and Outlook

This paper has surveyed the uptake of generative approaches to event extraction in NLP, presenting a range of different methods from encoder–decoders to decoder-only models. While some approaches take entire documents into account and others focus on the sentence-level, all evaluate performance based on matching predicted strings towards the strings found in the original input text.

We argue that the field has yet to embrace generative approaches to EE fully. Sticking to the traditional formulation of an "extraction" task makes it difficult to take full advantage of the capabilities of generative models like LLMs. The wide context windows of current LLMs also make them more suited for capturing more general or "complex events" – to use the words of Qi et al. (2022) – rather than the more granular and predicate-centered events typically targeted in the field so far. Going forward, we hope to see new formulations of the task itself, focusing on more high-level event analysis or understanding. By moving away from span-based and sentence-level annotations to more abstract and document-level annotations, with an evaluation methodology that correspondingly focuses on semantics rather than string matching towards a source text, we believe that the field can have a version of event analysis that will be more useful for many downstream applications (Olsen et al., 2024) and more attuned to the strengths and possibilities of generative approaches and LLMs. In fact, the arguments for such a shift from an "extractive" to an "abstractive" view could also be made for many other IE tasks in NLP where both modelling and evaluation are traditionally tied to span-based text annotations.

5 Limitations

In this survey, we adopt a narrow definition of generative methods – encoder–decoder and decoder-only transformers generating some natural language – to provide a detailed description of the systems rather than offering a broad overview. This focus allows for a more in-depth analysis but may limit the breadth of the discussion. Consequently, we are not discussing closely related work within information extraction, such as Named Entity Recognition and Relation Extraction. For readers seeking a broader perspective, we recommend the work of Huang et al. (2023) and Xu et al. (2023).

While this survey paper strives to cover all generative approaches to the task of event extraction within our scope, it is still possible that some relevant work has been unintentionally excluded, not due to a deliberate omission, but rather because it was not identified during our search. Our search was conducted across main NLP and AI venues such as ACL, EMNLP, and AAAI.

Some generative models were excluded as they did not generate natural language in their output, such as PAIE (Ma et al., 2022) and EEQA (Du and Cardie, 2020). We also excluded models such as RAP (Yao et al., 2023) as it is a generic method that could be plugged into any IE model generative or not. Some data-augmentation articles blur the line between dataset and model papers, most notably Gao et al. (2022) and are not included in our survey.

The page limit imposed on some articles made it hard to assess their characteristics, for example UIE (Lu et al., 2022) does not mention using constrained decoding in their article even though it is present in the code they provide. However, it is unclear whether this code path was actively used.

It is also worth noting that this survey does not extensively cover all datasets relevant to event extraction. The selection of datasets is guided by those used in evaluating the models the paper covers, which has led to a focus on English-language sources. Consequently, most datasets discussed in this survey are based on English, further reinforcing the overrepresentation of the English language.

Finally, while we address evaluation and performance in our discussions, we do not present evaluation scores for any of the models. Peng et al. (2023) describe several challenges in evaluating event extraction systems, highlighting issues such as discrepancies in output space and data processing, as well as the absence of pipeline evaluation, which impact the fair comparison of model performance. During the course of this research, we observed the same discrepancies in system evaluations.

6 Ethics

This work is intended to encourage further research within the framework of generative methods for event extraction. However, we acknowledge that several ethical concerns are inherent in this approach and may even be enhanced within this framework, warranting careful consideration.

Reliance on mainly English datasets for event extraction, coupled with the issue of hallucinations from large language models, might pose risks of harm and generate non-factual events, especially if not properly addressed. These risks should be given particular attention when moving towards more "abstractive" generative approaches.

Acknowledgements

This work was supported by the Research Council of Norway with funding to the *Peace Science Infrastructure* (project number 322425) and *MediaFutures: Research Centre for Responsible Media Technology and Innovation* (through the centers for Research-based Innovation scheme, project number 309339).

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a large language model a good annotator for event extraction? *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17772–17780.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ACE) program-tasks, data, and evaluation. In *International Conference on Language Resources and Evaluation*, volume 2, pages 837–840. Lisbon.

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.

Xinya Du, Sha Li, and Heng Ji. 2022a. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021a. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021b. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022b. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. 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.

Yanhe Fu, Yanan Cao, Qingyue Wang, and Yi Liu. 2024. TISE: A tripartite in-context selection method for event argument 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 1801–1818, Mexico City, Mexico. Association for Computational Linguistics.

William Gantt, Shabnam Behzad, Hannah An, Yunmo Chen, Aaron White, Benjamin Van Durme, and Mahsa Yarmohammadi. 2024. MultiMUC: Multilingual template filling on MUC-4. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 349–368, St. Julian's, Malta. Association for Computational Linguistics.

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2022. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4537–4544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *Preprint*, arXiv:2303.03836.

Cuiyun Han, Jinchuan Zhang, Xinyu Li, Guojin Xu, Weihua Peng, and Zengfeng Zeng. 2022. DuEE-Fin: A large-scale dataset for document-level event extraction. In *Natural Language Processing and Chinese Computing*, pages 172–183, Cham. Springer International Publishing.

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.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics.

Kuan-Hao Huang, I Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei Chang, Nanyun Peng, Heng Ji, et al. 2023. A reevaluation of event extraction: Past, present, and future challenges. *arXiv preprint arXiv:2311.09562*.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020a. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

Jiangnan Li, Yice Zhang, Bin Liang, Kam-Fai Wong, and Ruifeng Xu. 2023. Set learning for generative information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13043–13052, Singapore. Association for Computational Linguistics.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Xinyu Li, Fayuan Li, Lu Pan, Yuguang Chen, Weihua Peng, Quan Wang, Yajuan Lyu, and Yong Zhu. 2020b. DuEE: A large-scale dataset for chinese event extraction in real-world scenarios. In *Natural Language Processing and Chinese Computing*, pages 534–545, Cham. Springer International Publishing.

Jiangwei Liu, Liangyu Min, and Xiaohong Huang. 2021. An overview of event extraction and its applications. *arXiv preprint arXiv:2111.03212*.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

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.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, 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.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Helene Olsen, Étienne Simon, Erik Velldal, and Lilja Øvrelid. 2024. Socio-political events of conflict and unrest: A survey of available datasets. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 40–53, St. Julians, Malta. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

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.

Zheng Qi, Elior Sulem, Haoyu Wang, Xiaodong Yu, and Dan Roth. 2022. Capturing the content of a document through complex event identification. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 331–340, Seattle, Washington. Association for Computational Linguistics.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306, Toronto, Canada. Association for Computational Linguistics.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. CASIE: Extracting cybersecurity event information from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8749–8757.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

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.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023a. InstructUIE: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, Jie Zhou, and Juanzi Li. 2024. MAVEN-ARG: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4072–4091, Bangkok, Thailand. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023b. 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.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

Tongtong Wu, Fatemeh Shiri, Jingqi Kang, Guilin Qi, Gholamreza Haffari, and Yuan-Fang Li. 2023. KC-GEE: knowledge-based conditioning for generative event extraction. *World Wide Web*, pages 1–17.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024. YAYI-UIE: A chat-enhanced instruction tuning framework for universal information extraction. *Preprint*, arXiv:2312.15548.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *Preprint*, arXiv:2309.10305.

Yunzhi Yao, Shengyu Mao, Ningyu Zhang, Xiang Chen, Shumin Deng, Xi Chen, and Huajun Chen. 2023. Schema-aware reference as prompt improves data-efficient knowledge graph construction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 911–921, New York, NY, USA. Association for Computing Machinery.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. EA^{2}E: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655, Seattle, United States. Association for Computational Linguistics.

Model	Subtasks				Datasets								
	ED	**EAE**	**Joint**	**Multiple IE**	**ACE**	**CASIE**	**ERE**	**RAMS**	**WikiEvent**	**Backbone**	**Output Structure**	**Template in input**	**Special decoding**
TANL	✔	✔	✘	✔	✔	✘	✘	✘	✘	T5	Formal	✘	✘
Text2Event	✘	✘	✔	✘	✔	✘	✔	✘	✘	T5	Formal	✘	constrained
Set Learning	✘	✘	✔	✔	✔	✘	✘	✘	✘	T5	Formal	✘	constrained
KC-GEE	✘	✘	✔	✘	✔	✘	✘	✘	✔	T5	Formal	✘	constrained
Retrieve&Sample	✘	✔	✘	✘	✘	✘	✘	✔	✔	T5	Formal	✘	✘
UIE	✘	✘	✔	✔	✔	✔	✘	✘	✘	T5	Formal	✘	✘
DeepStruct	✔	✔	✘	✔	✔	✘	✘	✘	✘	GLM	Formal	✘	✘
InstructUIE	✔	✔	✔	✔	✔	✔	✘	✘	✘	T5	Formal	✱	✘
YAYI-UIE	✔	✔	✔	✔	✔	✔	✘	✘	✘	Baichuan2	Formal	✱	✘
LLM-EE	✔	✔	✘	✘	✘	✘	✘	✘	✘	Several	Formal	✱	✘
GTT	✘	✘	✔	✘	✘	✘	✘	✘	✘	BERT	Formal	✘	pointer
DICE	✔	✔	✘	✘	✘	✘	✘	✘	✘	T5	Formal	✘	✘
X-GEAR	✘	✔	✘	✘	✔	✘	✔	✘	✘	BART∨T5	Formal	✔	pointer
BART-Gen	✘†	✔	✘	✘	✔	✘	✘	✔	✔	BART	Natural	✔	pointer
EA^{2}E	✘	✔	✘	✘	✔	✘	✘	✘	✔	BART	Natural	✔	pointer
Memory DocIE	✘	✔	✘	✘	✘	✘	✘	✘	✔	BART	Natural	✔	pointer
DEGREE	✔	✔	✔	✘	✔	✘	✔	✘	✘	BART	Natural	✔	✘
GTEE-DynPref	✔	✔	✔	✘	✔	✘	✔	✘	✘	BART	Natural	✔	✘
AMPERE	✘	✔	✘	✘	✔	✘	✔	✘	✘	BART	Natural	✔	pointer
QGA-EE	✘	✔	✘	✘	✔	✘	✘	✘	✘	BART∨T5	Iterative	✘	✘

Table 1: List of models we introduce alongside some of their properties. This is a slightly expanded table version of Figure 3. For the "Backbone" column, a BART ∨ T5 means that the model was trained with multiple configurations, some with BART and some with T5. For the "Template in input" column, a "✱" means that there is an instruction on the nature of the output, but not the exact output template. †: The BART-Gen paper describe an event detection model, but it is not generative.