Honours Project Literature Review: Event Driven Causal Information Extraction from Text

Nathan Scott - 18913101

1 Oct 2024

1 Introduction

This literature review aims to survey relevant and state-of-the-art techniques for the Information Extraction (IE) sub-task: ***"event-centric causal triplet extraction from natural language text"***. While there exists a research gap for this specific sub-task, it will be shown that it shares the same granular components as other main-stream IE sub-tasks such as entity, entity-relation and event extraction. Thus all forms of IE sub-task and associated models will be reviewed.

1.1 Information Extraction

Information Extraction (IE) from text is an important NLP task. The fundamental goal being to extract structured data from unstructured text.

> **"Information extraction isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework.... The goal of IE research is to build systems that find and link relevant information while ignoring extraneous and irrelevant information."**[1]

IE is a way of bridging the domain of human knowledge to the world of machine data processing power. IE is commonly structured as a collection of sequence-to-sequence or sequence-to-span tasks at the sub-task level, however on the macro-level it is best framed as a sequence-to-graph task, where the goal is to output graph fragments (triplets) consisting of a head node, an edge, and a tail node. This approach is beneficial, as many downstream tasks involve constructing and utilizing knowledge graphs, such as question answering, event prediction, causal chain analysis, discourse comprehension, decision making, scenario generation, text mining [2]. Three core IE sub-tasks are thus:

- **Entity Extraction (NER):** the extraction of spans of text that refer to entities.
- **Relationship Extraction (RE):** the classification of the relationship given a head and tail.
- **Event Extraction (EE):** the extraction of spans of text that best describe the occurrences of events.

Additionally, auxiliary sub-tasks such as entity/event co-reference resolution can be critically important as it helps models disambiguate not only linguistic co-references (eg. $\text{entity}_{\text{pronoun}}$ and $\text{entity}_{\text{noun}}$) but also those found between extracted elements (eg. entity_{x} and entity_{y} or event_{x} and event_{y}).

1.2 Evolution of IE

Historically, methods for IE have followed the evolution of NLP through several key phases:

- **Knowledge based methods:** relying pattern and rule based approaches, these can be inflexible and difficult to maintain [3, 4]
- **Classical Machine Learning based methods:** utilizing classical machine learning and statistical techniques. Requiring extensive feature engineering, limitations include the inability to natively handle the sequential nature of text. [5, 6]
- **Deep Learning based methods:** the advent of deep neural networks allowed for the development

of models able of capturing more nuanced relationships in data. Specifically for IE from text, CNNs [7, 8], originally developed for image processing, enabled the detection of different scales of structure in sequences. While RNN [9] based models were the first type of neural network to actually model the sequential relationships between words, albeit relatively short range relative to later advancements with Bi-LSTMs [10] and Bi-GRUs [11] which addressed the vanishing gradient problem allowing the capturing of longer range relationships in both directions.

- **Discriminative Transformer based methods:** these involve models using an pre-trained and fine-tuned transformer encoder [12] like BERT [13] to produce enriched token representations for discriminative and predictive downstream IE tasks. The ability of the attention mechanism in the encoder to detect bi-directional, long range dependencies, as well as the large scale pre-training of these models have enabled them to largely hold and maintain the state of the art in IE performance.
- **Generative Transformer based methods:** this rapidly developing class of transformers are highly flexible, they use a pre-trained and fine-tuned auto-regressive transformer decoder core [12] to generate structured output text. Decoder only transformers such as GPT [14] or LLAMA [15] operate by predicting the next token given the input prompt tokens as well as any previously generated tokens. When scaled appropriately and fine-tuned for IE tasks, these types of transformer can show emergent behaviour. This allows them to perform IE tasks like a discriminative and predictive model, while the fundamental underlying operation remains purely generative. Encoder-decoder transformers such as BART [16] and FLAN [17] are also generative due to the auto-regressive decoder. However, they are distinct from decoder only transformers in that the next predicted output token is a function of all previously generated tokens along with the hidden representations of the input tokens, which are generated by the encoder.

2 Problem Definition

Here we define the specifics of event-centric causal triplet extraction from text.

2.1 Events and States

Causal relationships occur between 2 types of linguistic concepts, namely Events and States. Events typically describing the concept of a notable state of change of a system, while states represent the non-change of state of a system.[1]

Linguistically, events and states, almost exclusively take either of two main forms:

1. **the simple form:** with an entity(noun) trigger, eg.
   - the earthquake*(event)*
   - the depressed economy*(state)*
2. **the complex form:** with a verbal trigger and arguments, eg.[2]
   - she dropped the glass *(event)*
   - it's economy was running smoothly during that period *(state)*

2.2 Event Relationships

Defining a pair of events as e1 and e2, the pair can have various types of relationships including and not limited to:

- **temporal**
  - precedence: One event occurs before another without any causal implications.

[1]Note that states could be thought of as a special class of event, i.e. the anti-event, the event of a system not changing, usually with the associated trigger verb "to be" or another simple tense verb.

[2]Note that it is possible in the complex form, that the verbal trigger is omitted in some implicit cases where the meaning is left for the reader to infer from context, eg. "...the country's economy, running smoothly..."

    - succession: One event follows another in sequence, also without direct causation.
    - synchronous: Events that occur at the same time or during the same period, which might be correlated or have a mutual background cause.
    - conjunction: Two events occur together or are reported together.
- **conditional**
    - pre-conditional: One event is a necessary condition for the occurrence of another but doesn't directly cause it.
    - post-conditional: An event must occur if another specific event has occurred.
    - Enabling: An event creates the conditions that make it possible for another event to occur.
- **causal**
    - direct: One event directly causes another event to happen.
    - contributory: Where an event significantly contributes to the outcome but is not sufficient on its own to cause the event without other contributing factors.
    - complex: where multiple events occurring create the conditions for the final event to occur with no one event being solely causal.
    - catalytic: An event acts as a catalyst for another, speeding up the occurrence of the second event without being a direct cause.
    - aggravating: an event worsens or intensifies another event but does not initiate it.

Thus while the general concept of event-relation-event triplets appears simple, extraction from natural language text remains a non-trivial task. Complicating factors include the complexity of the relationship between two events, the various types of events and states and how relationships and events are encoded in natural language text:

- **explicit event/relation:** where an event or relationship is defined in a short span of text with an explicit trigger phrase.
- **implicit event/relation:** where an event or relationship is only implied by context, there being no definite trigger phrase.
- **intra-sentence:** where all the text pertaining to e1, e2 and their relationship is contained within a sentence.
- **inter-sentence:** where the text pertaining to e1, e2 and their relationship is spread over more than one sentence.

3 Information Extraction Sub-Tasks

The following section describes the main IE sub-tasks in more detail.

3.1 Entity Extraction from Text

Entity Extraction (NER) and classification is concerned with finding entities, which are spans of tokens that represent nouns (which could be events or states). This is a fundamental IE task. The task is typically done one of two ways:

- token classification methods where each token is classified as belonging to a type of entity or not, an N-to-N task.
- span classification methods where a span of tokens is classified as belonging to an entity type or not, an N-to-M task.

Token Classification vs Span Classification
The disadvantage of the token classification method is that it has trouble with overlapping spans. The span based approach allows overlapping spans, but requires some form of feature fusion technique to generate the span representations from the token representations.

3.2 Relation Extraction from Text

Relation Extraction (RE) is the task of classifying the relationship between a head and a tail, forming a triplet. The head and tail could represent entities, events, or states. The relation can be explicitly described by a trigger phrase, such as "because" in "the ground got wet because it rained,". Or it could be implied without a clear trigger, as in "it rained and the ground got wet." These relationships may occur within a single sentence or across multiple sentences. Thus, the relevant contextual information could be found in the text surrounding or between the head/tail spans, whether the spans are close together or far apart. An relation extraction system needs to be able to capture intricate contextual information pertaining to the head and the tail in order to identify relationships in implicit and explicit scenarios.

3.3 Event Extraction from Text

Event Extraction (EE) can be more complex than entity extraction as events themselves are more complex concepts that entities. As previously described (2.1), events can be encoded in various ways in text. When using the noun(entity) form, identification and classification follows the same procedure as for entities. However, when an event is encoded in the complex form, the process is more involved, a comprehensive EE extraction system should be able to handle both forms of event. Using the ACE annotation guidelines [18, 19, 20] and [6] as inspiration we define several key properties of events:

- **Event Extent/Mention:** This is the span in which all the event text lies. Typically it is going to be within one sentence but not always. In ACE [18], they actually assume that it is the entire sentence where the event resides for simplicity.
- **Event Trigger:** This is a phrase that is the primary descriptor of an event or state within text. The trigger is often a verb, especially in the full explicit form of events. However, in implicit forms of events the trigger can use an entity (usually an argument of the event) as a proxy for the full form of the event. For instance, the entity "explosion" in "The explosion caused widespread damage" serves as an entity/noun form of the verb "to explode," implicitly suggesting the event. Another implicit scenario is using an argument as the proxy for the event, eg. "Oxytocin causes brain-cancer", in this case "Oxytocin" is the object in the event of administering or long term usage of Oxytocin. Such variations, where entities imply events, can complicate EE as it blurs the direct causality usually expressed between events or states. For this reason, there is a perspective suggesting that it may be more effective for causal relation extraction systems to focus primarily on extracting the span that describes an event, rather than attempting to deconstruct the event into triggers and arguments, particularly when the event is only implicitly indicated.
- **Event Argument and Role:** An event argument is a span of text that has a definable role in relation to the event. These arguments are typically identifiable entities but can also include other data types such as temporal statements and quantitative values. The role of each argument clarifies its function within the event, effectively answering the 5W1E questions (Who, What, When, Where, Why, and How) that detail the specifics of the event, the exact nature of which arguments and roles will be event schema dependent.
- **Schema:** An event schema is a structure template for an event. The schema defines the type of event and the associated roles and types of the arguments.

The standard approach is to further sub-divide EE into two sub-tasks:

- **Event Trigger Classification:** The trigger phrase is detected, extracted and classified. For events with an explicit verbal/noun trigger words, this can be a straightforward task, however for implicitly

or unconventionally structured events, this can be more difficult.

- **Event Argument Role Classification:** Depending on the event type various arguments are detected, extracted and the role in the event classified. As the event arguments are spans of text which have a defined role in the event. The second sub-task can be framed as an entity detection/extraction task followed by event-argument relation classification.

Simplifying Event Extraction
Note that it is also potentially beneficial to stay above this complexity and just train a model to extract the whole span that best describes an event or state. This aligns with datasets such as Altlex [21] and is a path to consider as opposed to the extra complexity of breaking complex events into their components. Complex events can be further broken down at a later stage if needed, but for the process of causal triplet extraction this may not be necessary.

3.4 Causal Relation Extraction from Text

Causal Relation Extraction (CRE) is a more specific form of RE that focuses on identifying and classifying causal relationships between events or states. Unlike traditional RE, which primarily handles relationships between entities, CRE requires understanding the temporal and logical connections between events, which often involves deeper semantic reasoning. CRE overlaps with mainstream IE sub-tasks for NER-RE and EE, but has seen less research activity, potentially as this moves into other non-IE research areas such as causal reasoning that tend to processes knowledge graphs that have already been extracted previously.

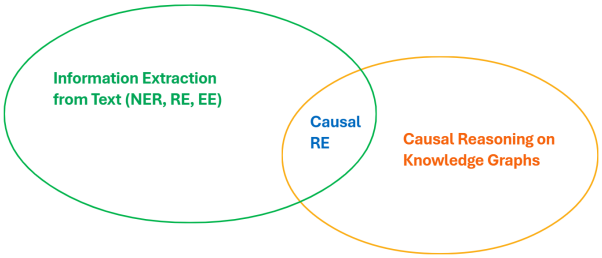


Figure 1: CRE Straddles the IE and Causal Reasoning Domains

CRE can be categorized into different types based on the explicitness and location of the causal relationship [22, 2, 23]:

- **Explicit Intra-sentential Causality:** The causal relationship is clearly expressed within a single sentence, often using connectives such as "because," "since," "therefore," or causative verbs like "cause," "lead to," or "result in."
- **Implicit Causality:** The causal relationship is implied but not explicitly stated, requiring inference based on contextual information and background knowledge. For example, "The rain stopped, and the sun came out."
- **Inter-sentential Causality:** The causal relationship spans multiple sentences, necessitating an understanding of connections between events across sentence boundaries. For example, "The stock market crashed. Investors lost a lot of money."

CRE presents several challenges compared to traditional RE:

- **Ambiguity:** Causal connectives and verbs can have multiple interpretations, making it difficult to determine if a relationship is truly causal.
- **Implicit Causality:** Many causal relationships are not explicitly stated, requiring models to infer connections from context and background knowledge.
- **Complex Event Form:** While states and many events take the noun/entity-proxy form, some events

use the complex form with a trigger + arguments. This adds complexity to the causal relation identification task, eg. "The police locked up the criminal", requires either the whole event span to be detected and/or the components to be extracted.

- **Inter-sentential Dependencies:** Identifying causal relationships across sentence boundaries requires more complex systems to extract contextual signal important to an event pair. This becomes increasingly challenging over longer distances due to co-reference issues, context window sizes, as well as contextual signal dilution.
- **Lack of Annotated Data:** There is a general lack of quality annotated data for model training.

4 Models

In this section, we first review some background on the evolution of discriminative IE models and give a detailed breakdown of the generative transformer class of models 4.1.

We then offer a model summary table 4.2, followed by reviews of key IE models, organised by IE sub-task:

- Entity and Relation Extraction (NER-RE) 4.3
- Event Extraction (EE)4.4
- Multi-Task models which perform all three sub-tasks (NER-RE-EE) 4.5
- Causal Relation Extraction (CRE) 4.6

Lastly a review of strategies for Inter-Sentential RE focussed on transformer based models is offered 4.7.

> Note that the foundational IE sub-tasks of NER, RE and EE are critical to the overall performance of Causal Relation Extraction (CRE), as CRE relies on accurately identifying entities, events and relationships. By examining the range of approaches applied to IE, we aim to highlight techniques that have proven effective in their respective areas, while also identifying those that could be adapted or re-purposed specifically for CRE systems.

4.1 Background

This background section reviews several technical aspects of contemporary IE models:

- the distinction between pipeline and joint models 4.1.1
- the evolution of discriminative IE models 4.1.2
- breakdown of key aspects of generative transformer models 4.1.3

4.1.1 Pipeline vs Joint Methods

Pipeline models for Information Extraction (IE) typically use independent classifiers trained on their respective loss functions for tasks such as Named Entity Recognition (NER) and Relation Extraction (RE). This approach simplifies the individual tasks, allowing each model to focus on its specific problem, making them more task-focused. However, pipeline models have a major downside: error propagation. If the NER component misclassifies an entity, those errors may carry over into the RE component, affecting the final results. In contrast, joint models aim to address the issue of error propagation by sharing internal components, such as a common encoder like BERT, and combining loss functions. This integration allows the model to learn from both tasks simultaneously, often leading to improvements across tasks due to shared knowledge.

Historically, pipeline models were more prevalent, though they are now less common with the rise of joint models. The empirical study [24] emphasizes that while the best joint approaches generally outperform pipeline models, a well-designed pipeline can achieve competitive results, especially when task independence benefits overall performance. The current state-of-the-art model for NER-RE is a pipeline model [25]. Its success is partly due to the independence between the NER and RE BERT based classifiers, which highlights that joint models are not always the superior choice, pipelines can still be effective depending on the context

and task formulation.

These principles can also apply to more complex tasks such as event extraction and causal relation extraction, where both pipeline and joint approaches can be valuable depending on the specific requirements of the task.

4.1.2 Discriminative IE Methods

As previously indicated discriminative IE models have followed the evolution of NLP techniques through knowledge based approaches, machine learning, neural networks and finally discriminative transformers.

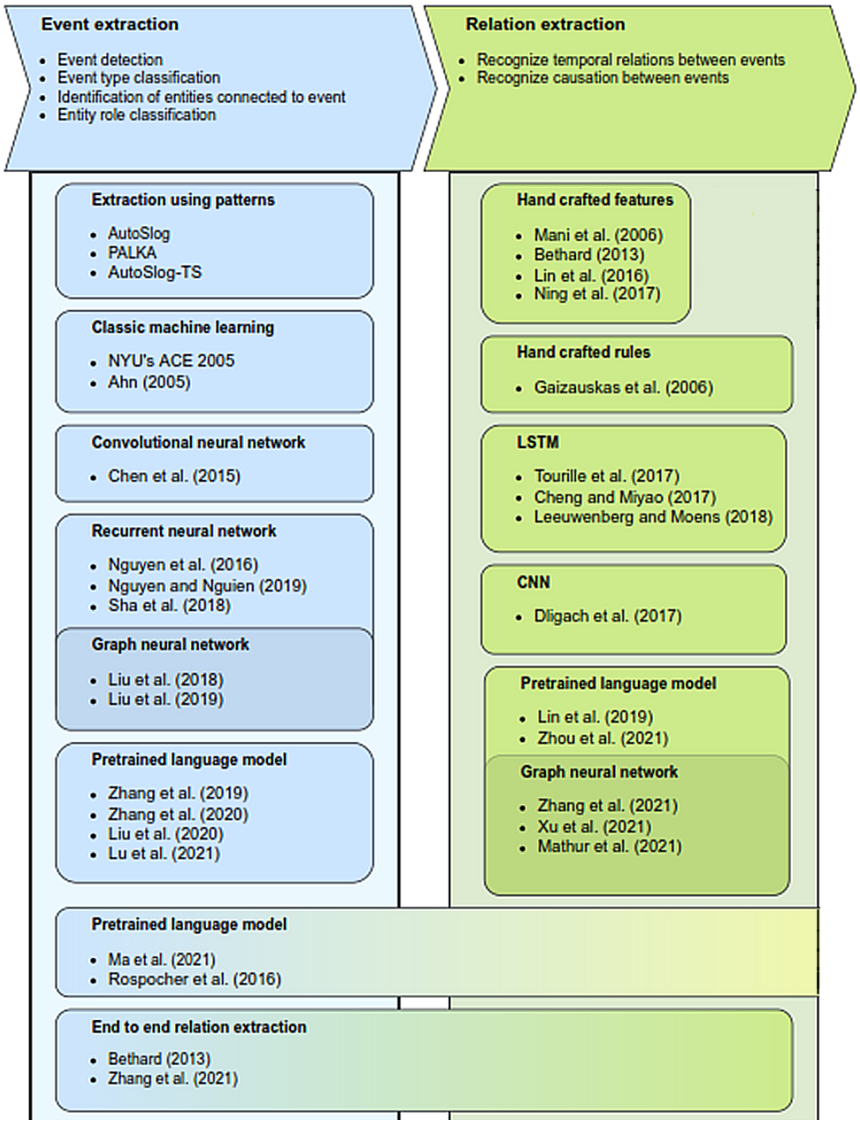


Figure 2: EE-RE Evolution [26]

4.1.3 Generative Transformer based IE Methods

Generative Pre-trained Transformers (commonly referred to as Large Language Models - LLMs) when fine-tuned for IE tasks can behave like discrimitaive models in that they output structured text, however the underlying processes are generative in nature. These models typically need to be large in size to operate effectively, featuring 10^{9+} to over 10^{11+} parameters [14, 15, 17, 27, 28]. On an end-2-end level, they are

text-in-text-out models, meaning that they are prompted with text and produce a textual response. This opens up possibilities for not only information extraction but also the generation of synthetic training data and automated annotation [29].

Generative Transformer Learning

While training generative transformers from scratch is requires industrial scale resources, various learning paradigms exist to utilize pre-trained LLMs, fig. 3 ([30]) highlights 4 categories of pre-trained generative transformer learning:

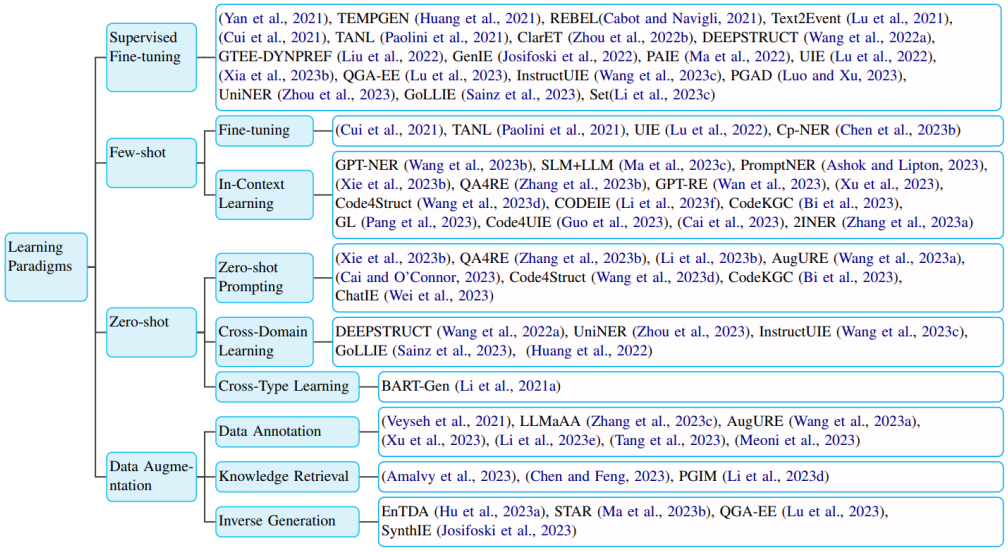


Figure 3: Generative Transformer Learning Methods [30]

- **Supervised fine tuning (SFT):** this is transfer learning where the pre-trained model weights are adjusted in a supervised learning approach based on a annotated dataset. Models using this paradigm currently show the best performance on IE tasks, see 4, 5, 6.
- **Zero-shot Learning:** is when a pre-trained model is prompted to perform a specific task with no training examples given. For IE tasks this could take the form of a prompt with the text and instructions and schema to return, but no examples.
- **Few-shot/In-Context Learning:** is for the scenario that a pre-trained model is prompted with instructions, output schema, text to process and some training examples showing acceptable outputs for various inputs. The difference with zero-shot learning is that the model has more task specific information with which to construct its output [31].
- **Data Augmentation:** is when the generative abilities of the model are leveraged to perform various dataset augmentation tasks to boost the performance of SFT or few-shot learning. Data annotation is when the generative model is utilized to annotate a dataset [29], it can greatly benefit IE models by increasing the amount of annotated data, however management of noise and error is paramount.

Comparative performance results from [30] for three primary IE tasks (NER, RE, EE) is shown in figs. 4, 5, 6. Key take-aways are:

- **NER:** models using the SFT learning paradigm tend to outperform non-fine-tuned generative models using zero/few-shot prompting.
- **RE:** UIE models tend to perform better on complex RE tasks due to the inter-dependencies with other IE tasks. SFT-UIE models being the best performers.
- **EE:** SFT models out-perform ICL models, with UIE based models being top performers.

Uni?: *UIE;* ***SFT:*** *supervised Fine Tuning;* ***DA:*** *Data Augmentation;* ***CDL:*** *Cross Domain Learning;* ***ZS Pr:*** *ZEro-shot prompting;* ***ICL:*** *In-Context/few-shot Learning*

Representative Model	Paradigm	Uni. ?	Backbone	ACE04	ACE05	CoNLL03	Onto. 5[1]	GENIA
DEEPSTRUCT (Wang et al., 2022a)	CDL		GLM-10B		28.1	44.4	42.5	47.2
(Xie et al., 2023b)	ZS Pr		Gpt-3.5-turbo		32.27	74.51		52.06
CODEIE (Li et al., 2023f)	ICL	√	Code-davinci-002	55.29	54.82	82.32		
Code4UIE (Guo et al., 2023)	ICL	√	Text-davinci-003	60.1	60.9	83.6		
PromptNER (Ashok and Lipton, 2023)	ICL		GPT-4			83.48		58.44
(Xie et al., 2023b)	ICL		Gpt-3.5-turbo		55.54	84.51		58.72
GPT-NER (Wang et al., 2023b)	ICL		Text-davinci-003	74.2	73.59	90.91	82.2	64.42
TANL (Paolini et al., 2021)	SFT	√	T5-base		84.9	91.7	89.8	76.4
(Cui et al., 2021)	SFT		Bart			92.55		
(Yan et al., 2021)	SFT		Bart-large	86.84	84.74	93.24	90.38	79.23
UIE (Lu et al., 2022)	SFT	√	T5-large	86.89	85.78	92.99		
DEEPSTRUCT (Wang et al., 2022a)	SFT	√	GLM-10B		86.9	93.0	87.8	80.8
(Xia et al., 2023b)	SFT		Bart-large	87.63	86.22	93.48	90.63	79.49
InstructUIE (Gui et al., 2023)	SFT	√	Flan-T5-11B		86.66	92.94	90.19	74.71
UniNER (Zhou et al., 2023)	SFT		LLaMA-7B	87.5	87.6		89.1	80.6
GoLLIE (Sainz et al., 2023)	SFT	√	Code-LLaMA-34B		89.6	93.1	84.6	
EnTDA (Hu et al., 2023a)	DA		T5-base	88.21	87.56	93.88	91.34	82.25

Figure 4: Generative Transformer NER Comparison, micro-F1 [30]

Representative Model	Paradigm	Uni. ?	Backbone	NYT	ACE05	ADE	CoNLL04	SciERC
CodeKGC (Bi et al., 2023)	ZS Pr	√	Text-davinci-003			42.8	35.9	15.3
CODEIE (Li et al., 2023f)	ICL	√	Code-davinci-002	32.17	14.02		53.1	7.74
CodeKGC (Bi et al., 2023)	ICL	√	Text-davinci-003			64.6	49.8	24.0
Code4UIE (Guo et al., 2023)	ICL	√	Text-davinci-002	54.4	17.5	58.6	54.4	
REBEL (Cabot and Navigli, 2021)	SFT		Bart-large	91.96		82.21	75.35	
UIE (Lu et al., 2022)	SFT	√	T5-large		66.06		75.0	36.53
InstructUIE (Wang et al., 2023c)	SFT	√	Flan-T5-11B	90.47		82.31	78.48	45.15
GoLLIE (Sainz et al., 2023)	SFT	√	Code-LLaMA-34B		70.1			
USM† (Lou et al., 2023)	SFT	√	Roberta-large		67.88		78.84	37.36
RexUIE† (Liu et al., 2023)	SFT	√	DeBERTa-v3-large		64.87		78.39	38.37

Figure 5: Generative Transformer RE Comparison, micro-F1 strict-relation [30]

Representative Model	Paradigm	Uni. ?	Backbone	Trg-I	Trg-C	Arg-I	Arg-C
Code4Struct (Wang et al., 2023d)	ZS Pr		Code-davinci-002			50.6	36.0
Code4UIE (Guo et al., 2023)	ICL	√	Gpt-3.5-turbo-16k		37.4		21.3
Code4Struct (Wang et al., 2023d)	ICL		Code-davinci-002			62.1	58.5
TANL (Paolini et al., 2021)	SFT	√	T5-base	72.9	68.4	50.1	47.6
Text2Event (Lu et al., 2021)	SFT		T5-large		71.9		53.8
BART-Gen (Li et al., 2021a)	SFT		Bart-large			69.9	66.7
UIE (Lu et al., 2022)	SFT	√	T5-large		73.36		54.79
GTEE-DYNPREF (Liu et al., 2022)	SFT		Bart-large		72.6		55.8
DEEPSTRUCT (Wang et al., 2022a)	SFT	√	GLM-10B	73.5	69.8	59.4	56.2
PAIE (Ma et al., 2022)	SFT		Bart-large			75.7	72.7
PGAD (Luo and Xu, 2023)	SFT		Bart-base			74.1	70.5
QGA-EE (Lu et al., 2023)	SFT		T5-large			75.0	72.8
InstructUIE (Wang et al., 2023c)	SFT	√	Flan-T5-11B		77.13		72.94
GoLLIE (Sainz et al., 2023)	SFT	√	Code-LLaMA-34B		71.9		68.6

Figure 6: Generative Transformer EE Comparison on ACE05, micro-F1 [30]

Generative Transformers for IE

The advantages of generative transformers for IE are that the text-in-text-out paradigm leads to very fast model prototyping as well as a level of model flexibility that has never previously been possible. The large

model sizes also mean that the LLM can capture far more nuanced relationships within textual inputs than smaller models. However on the downside, LLMs are compute resource hungry, can be very expensive to pre-train and can hallucinate.

Some specific breakthroughs that are currently occurring with generative transformers which could benefit IE are:

- Larger, more capable open source models.
- Better model quantization technologies to enable training and inference on smaller resource footprints
- Larger context windows.

The following tree of Generative Transformer model tasks [30] breaks down IE tasks and the associated models into NER (Named Entity Recognition), RE (Relation Extraction), EE (Event Extraction) and UIE (Universal Information Extraction). UIE models are interesting as they leverage the flexibility of Generative Transformers to adapt to a prompting schema and perform multi-task IE. Models tend to follow either a code prompting paradigm (prompting in code form) or a natural language prompting paradigm.

Figure 7: Generative Transformer IE Tasks [30]

4.2 Model Summary

The following table summarises some high performing recent models of interest on the various IE tasks along with their performance on associated datasets 5.1. Of note are are the "Model Type" and "Backbone" columns which refer to the general architecture and transformer backbone respectively. If available, links to the associated paper and source are given. Acronyms used in these columns are highlighted below:

- TR = Transformer (any type)
- DTR = Discriminative Transformer (pre-trained)
- GTR = Generative Transformer (pre-trained)
- PT = extra Pre-Training Required
- FT = Fine-Tuning Required
- Att = Attention Mechanism

Table 1: Models and Performance Across Datasets

Model	Model Type	Backbone	NER	RE	EE	CRE	Code	Paper
CMAN	**LSTM-Att**	**non-TR**	✓	✓	✗	✗	no code	paper
	F1-micro on "ADE"		89.4	81.1	-	-		
	F1-micro on "CoNLL04"		90.6	73	-	-		
DeepStruct	**GTR-PT-FT**	**GLM-10B**	✓	✓	✓	✗	code	paper
	F1-micro on "ACE05"		90	66.8	64.7	-		
	F1-micro on "ADE"		91.1	83.8	-	-		
	F1-micro on "CoNLL04"		90.7	78.3	-	-		
	F1-micro on "Genia2011"		80.8	-	-	-		
	F1-micro on "NYT"		95.9	93.3	-	-		
	F1-micro on "TACRED"		-	76.8	-	-		
DREEAM	**DTR-FT**	**RoBERTa**	✗	✓	✗	✗	code	paper
	F1-micro on "DocRED"		-	67.5	-	-		
DYGIE++	**DTR-FT-GNN**	**BERT**	✓	✓	✓	✗	code	paper
	F1-micro on "ACE05"		88.8	63.4	64.5	-		
	F1-micro on "Genia2011"		77.9	-	-	-		
	F1-micro on "SciERC"		67.5	48.4	-	-		
EXOBRAIN	**DTR-FT**	**RoBERTa**	✗	✓	✗	✗	no code	paper
	F1-micro on "Re-TACRED"		-	91.4	-	-		
	F1-micro on "TACRED"		-	75	-	-		
FourIE	**DTR-FT-GCN**	**BERT**	✓	✓	✓	✗	no code	paper
	F1-micro on "ACE05"		88.9	68.9	68.1	-		
GoLLIE	**GTR-FT**	**Code-LLaMA**	✓	✓	✓	✗	code	paper
	F1-micro on "ACE05"		89.6	70.1	70.3	-		
	F1-micro on "Onto. 5"		84.6	-	-	-		
GraphER	**DTR-FT-Att-GNN**	**ALBERT**	✓	✓	✗	✗	code	paper
	F1-micro on "ACE05"		89.8	68.4	-	-		
	F1-micro on "CoNLL04"		89.6	76.5	-	-		
	F1-micro on "SciERC"		69.2	50.6	-	-		
InstructUIE	**GTR-FT**	**FLAN-T5-11B**	✓	✓	✓	✗	code	paper
	F1-micro on "ACE05"		86.7	-	75	-		
	F1-micro on "ADE"		-	82.3	-	-		
	F1-micro on "CoNLL04"		-	78.5	-	-		
	F1-micro on "Genia2011"		74.7	-	-	-		
	F1-micro on "NYT"		-	90.5	-	-		
	F1-micro on "Onto. 5"		90.2	-	-	-		
	F1-micro on "SciERC"		-	45.1	-	-		
JMEE	**LSTM-GNN**	**non-TR**	✗	✗	✓	✗	no code	paper
	F1-micro on "ACE05"		-	-	69.6	-		
KEECI	**DTR-FT**	**BERT**	✗	✗	✗	✓	code	paper
	F1-micro on "SemEval2010 T8"		-	-	-	66		
MCDN	**Att-GRU-CNN**	**non-TR**	✗	✗	✗	✓	code	paper

Continued on next page

Table 1 continued from previous page

Model	Model Type	Backbone	NER	RE	EE	CRE	Code	Paper
	F1-micro on "AltLex"		-	-	-	82.5		
MCNN	**CNN**	**non-TR**	✗	✗	✗	✓	no code	paper
PFN	**PFN**	**non-TR**	✓	✓	✗	✗	code	paper
	F1-micro on "ACE05"		89	66.8	-	-		
	F1-micro on "ADE"		91.5	83.9	-	-		
	F1-micro on "NYT"		95.8	92.4	-	-		
	F1-micro on "SciERC"		66.8	38.4	-	-		
	F1-micro on "WebNLG"		98	93.6	-	-		
PLmarker	**DTR-FT**	**ALBERT**	✓	✓	✗	✗	code	paper
	F1-micro on "ACE05"		91.1	73	-	-		
	F1-micro on "Onto. 5"		91.9	-	-	-		
	F1-micro on "SciERC"		69.9	53.2	-	-		
PLMEE	**DTR-FT**	**BERT**	✗	✗	✓	✗	no code	paper
	F1-micro on "ACE05"		-	-	72.1	-		
RAG4RE	**GTR-RAG**	**FLAN-T5-11B**	✗	✓	✗	✗	code	paper
	F1-micro on "Re-TACRED"		-	73.3	-	-		
	F1-micro on "SemEval2010 T8"		-	14.1	-	-		
	F1-micro on "TACRED"		-	86.6	-	-		
RCEE	**DTR-FT**	**BERT**	✗	✗	✓	✗	code	paper
	F1-micro on "ACE05"		-	-	69.3	-		
REBEL	**GTR-FT**	**BART**	✗	✓	✗	✗	code	paper
	F1-micro on "ADE"		-	82.2	-	-		
	F1-micro on "CoNLL04"		-	75.4	-	-		
	F1-micro on "DocRED"		-	47.1	-	-		
	F1-micro on "NYT"		-	92	-	-		
	F1-micro on "Re-TACRED"		-	90.4	-	-		
SP	**DTR-FT**	**BERT**	✗	✓	✗	✗	no code	paper
	F1-micro on "SemEval2010 T8"		-	91.9	-	-		
	F1-micro on "TACRED"		-	74.8	-	-		
SPERT	**DTR-FT**	**BERT**	✓	✓	✗	✗	code	paper
	F1-micro on "ADE"		89.2	79.2	-	-		
	F1-micro on "CoNLL04"		88.9	71.5	-	-		
	F1-micro on "SciERC"		70.3	50.8	-	-		
SPLSTM	**LSTM**	**non-TR**	✗	✗	✗	✓	code	paper
	F1-micro on "AltLex"		-	-	-	82		
UIE	**GTR-PT-FT**	**T5-Large**	✓	✓	✓	✗	code	paper
	F1-micro on "ACE05"		85.8	66.1	64.1	-		
	F1-micro on "CoNLL04"		-	75	-	-		
	F1-micro on "SciERC"		-	36.5	-	-		
UniRel	**DTR-FT**	**BERT**	✗	✓	✗	✗	code	paper
	F1-micro on "NYT"		-	93.7	-	-		
	F1-micro on "WebNLG"		-	94.7	-	-		
USM	**DTR-FT**	**RoBERTa**	✓	✓	✓	✗	no code	paper
	F1-micro on "ACE05"		87.1	67.9	64.1	-		
	F1-micro on "CASIE"		-	-	53.6	-		
	F1-micro on "CoNLL04"		-	78.8	-	-		
	F1-micro on "NYT"		-	94.1	-	-		
	F1-micro on "SciERC"		-	37.4	-	-		

4.3 NER-RE Models

These models are focused two primary IE sub-tasks, NER (a span classification task) and RE (an span-pair relation classification task). Where the RE is for the relation between entities. This area of IE has more research and code resources than other areas.

4.3.1 CMAN

(2018-[32, 33]) CMAN is a joint NER-RE classification model that uses non-transformer embeddings with Bi-LSTMs, self-attention, and cross-attention blocks, followed by a Conditional Random Field (CRF) head for NER and a multi-relation classification head for relation extraction (RE). Notable aspects of this model include the use of self-attention to enrich input word representations, cross-attention to incorporate NER label information into token representations, and a multi-relation classifier head structure.

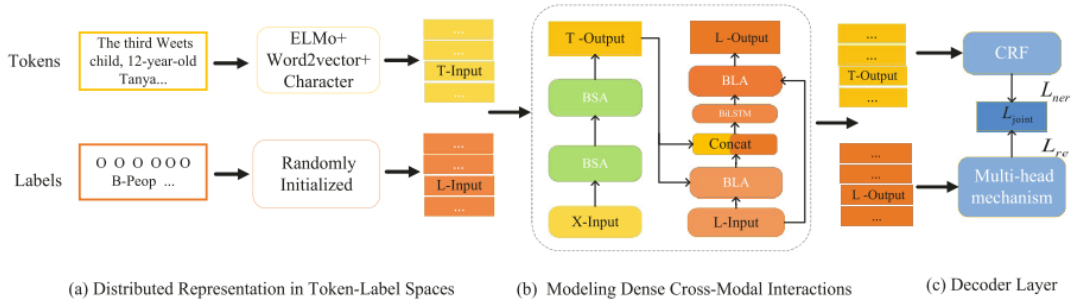


Figure 8: CMAN [32]

4.3.2 GraphER

(2024-[34, 35, 36]) GraphER is a joint NER-RE model. This model leverages transformer encodings and uses heuristics to form an initial noisy graph of entities and their relationships. It applies a graph structure-aware full attention mechanism (TokenGT [35]) to refine the initial graph representations before applying graph structure learning (GSL [36]) to prune and optimize the graph. This approach is designed to overcome two significant challenges seen in traditional GCN/GAT-based models: the bottleneck problem and the lack of structure learning. While this model is designed for NER-RE, it could be adapted for EE-CRE.

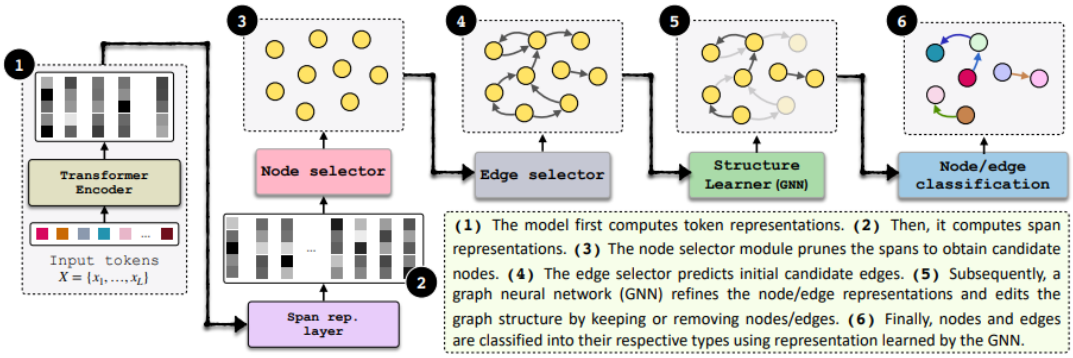


Figure 9: Knowledge Enhanced Event Causality Identification [34]

4.3.3 PL-marker

(2022-[25]) PLmarker is an LM based pipeline NER-RE span model based on efficient pre-marking entity and relation spans in input text prior to two independent transformer encoders, inspired by [37]. Notable aspects of this approach are that it is the current SOTA for NER-RE, it's simplicity and the counter-intuitive fact that pre-encoder marking in a pipelined manner exceeds the performance of other more complex joint models.

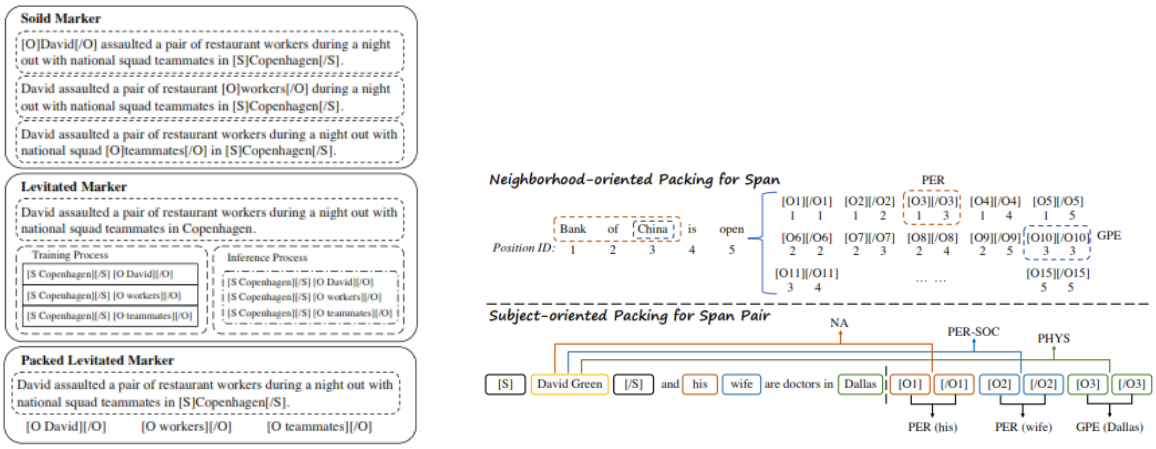


Figure 10: PLmarker [25]

4.4 EE Models

These models are focused the EE subtask. EE is typically broken down into Event Trigger identification/extraction/classification (a span classification task) and Event Argument identification/extraction and role classification (a span-pair relation classification task).

4.4.1 DEEB-RNN

(2018-[38]) DEEB-RNN, is a non-transformer word-token classification model that only performs Event Trigger detection and classification. It also requires NER results to work. While older, a notable aspect is the use of hierarchical attention to generate cross sentence (entire document) level representations, which are then combined with word and entity type embeddings prior to a Bi-GRU mixing and classification.

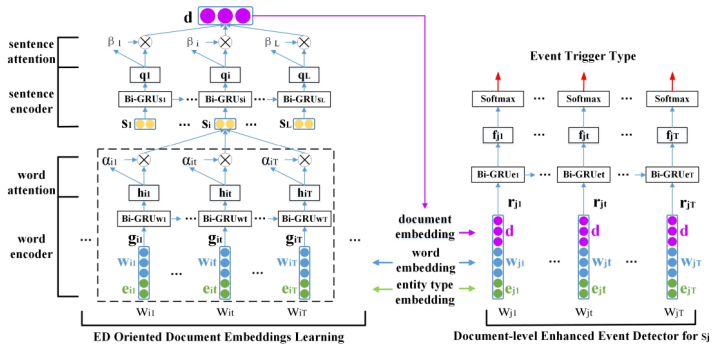


Figure 11: DEEB-RNN [38]

4.4.2 JMEE

(2018-[39]) JMEE is an Event Extraction model. This model utilized non-LM embeddings along with Bi-LSTM, GCN and self attention networks for joint event trigger and argument extraction. Note that it requires that the entities have already been extracted.

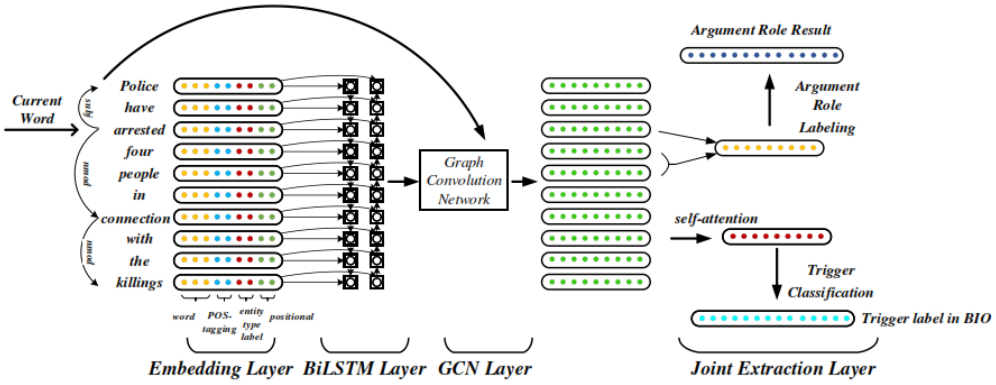


Figure 12: JMEE [39]

4.4.3 PLMEE

(2019-[40]) PLMEE is a model for Event Extraction, it operates as an pipeline model based on token classification with two independent BERT instances, one for the event trigger and the other for the arguments.

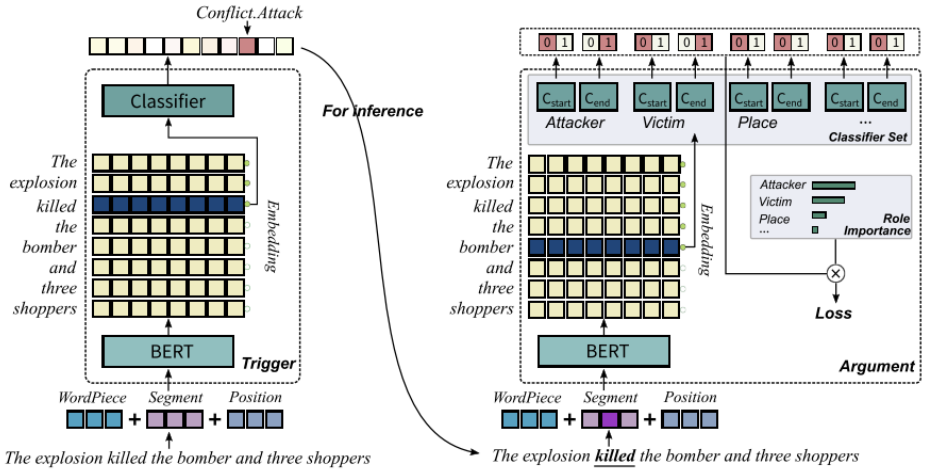


Figure 13: PLMEE [40]

4.5 Multi-Task Models

These models attempt to perform all three IE subtasks (NER, RE and EE).

4.5.1 DYGIE++

(2019-[41, 42]) DYGIE++ is a model for NER-RE and EE. It is a multi-task span-graph based joint model. It operates by processing transformer representations into span representations, then enriching them via graph techniques, allowing for the simultaneous extraction of entities, relations, and events. Another notable aspect is the use of overlapping input sequences to capture inter-sentence context.

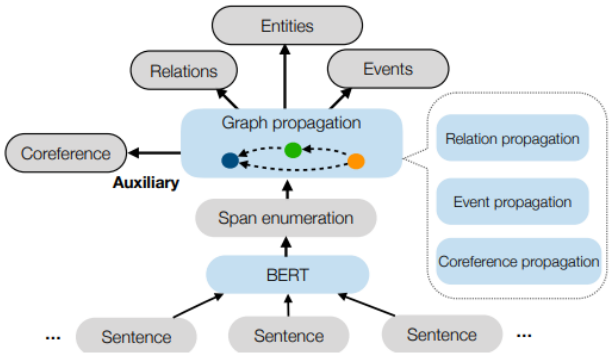


Figure 14: DYGIE++ [41]

4.5.2 InstructUIE

(2023-[43]) IntructUIE (Multi-task Instruction Tuning for Unified Information Extraction) is a high performing natural language prompting UIE Generative Transformer that can use fine-tuning to train just one model for multiple tasks. Additionally, it can be used in Few-shot mode with a few examples and no fine-tuning. The fundamental premise is to engineer descriptive natural language prompts to clearly outline each of the various IE tasks, each prompt has the following schema:

- **Instruction Clause:** detailing what task to perform, this details what IE task to perform, what elements to output and in what format along with any other specifics.
- **Option Clause:** detailing the result schema, i.e. the range of values that are allowed in the output.
- **Input Clause:** the text to process.
- **Output Clause:** has the labels (training) or blank to be filled by the model (inference).

Model fine-tuning (called instruction-tuning in the paper) is performed via labeled data for the main IE tasks (NER, RE, EE). Additionally these main tasks are broken down in to smaller auxiliary sub-tasks and added to the training to further help the model learn how to perform the more complex main tasks.

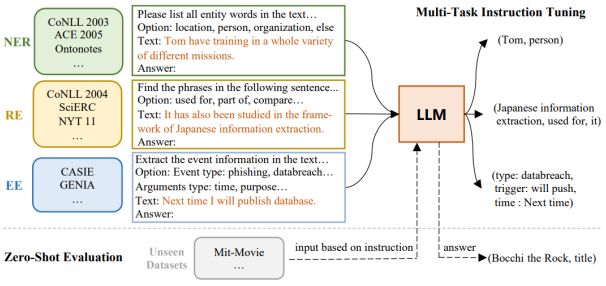


Figure 15: InstructUIE [43]

4.5.3 GoLLIE

(2024-[44]) GoLLIE (Guideline following Large Language Model for IE) is an code prompting UIE Generative Transformer. GoLLIE thus is built on a code optimized version of Llama which is better suited to parsing and generating structured outputs. Notable aspects of this model are:

- The model allows for fine-tuning over a range of datasets and tasks.
- They employ several regularization techniques to minimise hallucinations and make the model more robust and generalizable.
- The idea of using a code-in/code-out approach with code-specialized models is novel and makes a lot of sense for structured information extraction as well as specifying complex schema.

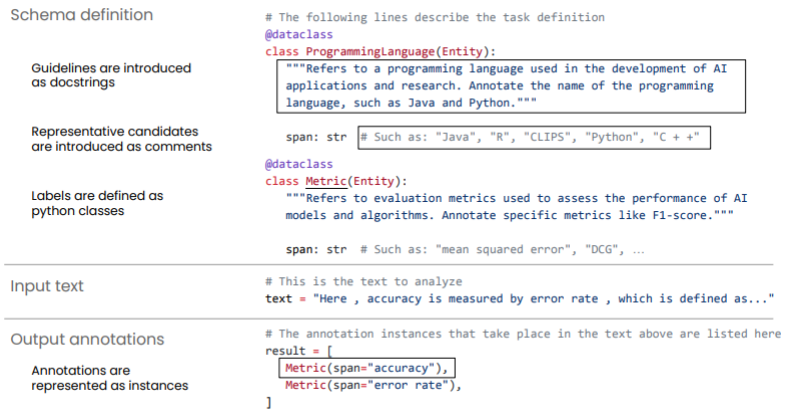


Figure 16: GoLLIE [44]

4.5.4 UIE

(2022-[45]) UIE (Unified Information Extraction) is a natural language prompting UIE Generative Transformer model. They frame the model as text-to-structure where all tasks are basically transformations of the input prompt text to a structured output text. They define an input guideline format SSI (Structural Schema Instructor) and output format SEL (Structured Extraction Language), which are both pseudo JSON-like formats. The prompt contains the SSI along with the input text and the model generates the output in SSI format. A fundamental premise is that all IE tasks can be broken down to two granular components:

- **Spotting:** these are span semantic types to detect and extract.
- **Associating:** these are span relation semantic role types to detect and extract.

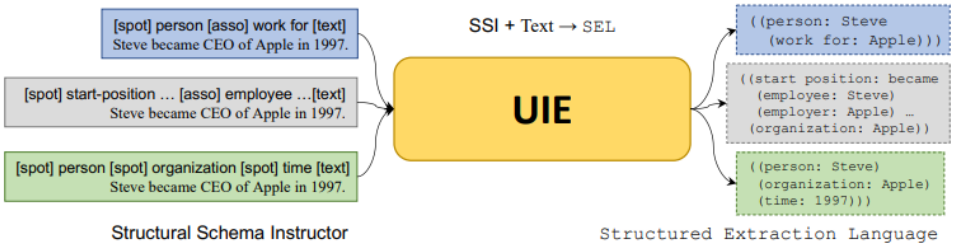


Figure 17: UIE [45]

4.6 CRE Models

These models are only concerned with identifying causal relations between spans of text. In all the papers reviewed the model require that the spans have already been identified.

4.6.1 MCNN

(2017-[8]) MCNN is an CNN-based non-transformer model for event causality classification. Notable features are how it combines external knowledge and contextual information with the event pair representations. Note that it requires that the event spans have already been extracted and decomposed to function.

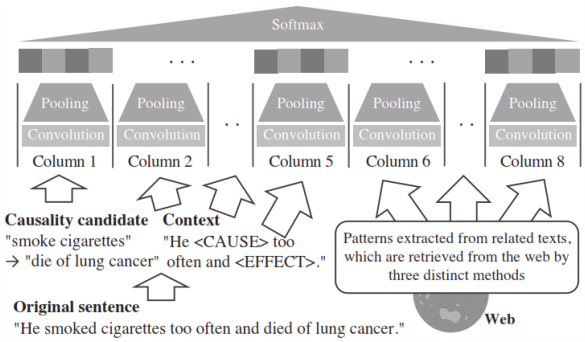


Figure 18: MCNN [8]

4.6.2 Stated Pair LSTM

(2017-[46]) The Stated Pair LSTM model leverages GloVe embeddings to process textual data along with event spans for causality classification. The model splits a given sentence into two segments, which are padded to equal lengths: (1) the first event span and (2) the concatenation of any relation trigger, a separator token and the second event span. These segments are each processed by separate LSTM networks. Noteworthy aspects of this model include:

- Its utilization of complete event spans without decomposing the events into smaller components, which simplifies the input structure.
- The model's adaptability to both explicit and implicit relational triggers, enabling it to handle varied linguistic contexts effectively.

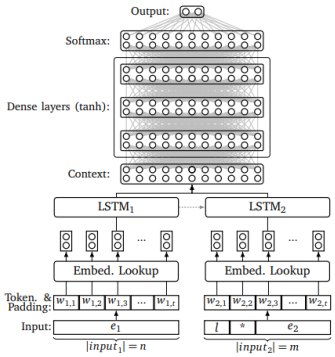


Figure 19: Stated Pair [46]

4.6.3 MCDN

(2021-[47]) uses non-transformer embeddings, an MHA layer, Multi-column CNNs to classify the causal relationship between two events. The MCDN model operates on the premise that two events and a relation trigger (denoted as 'AltLex') within the text are pre-identified. This identification allows the model to segment the text into three parts: before the relation trigger (BL), the relation trigger itself (L), and after the relation trigger (AL). Notable aspects of this model:

- It operates on event spans as opposed to decomposed event details
- While it needs an relation trigger, it could easily be adapted for implicit scenarios
- The way it processes the segment representations and combines them with the word level representations

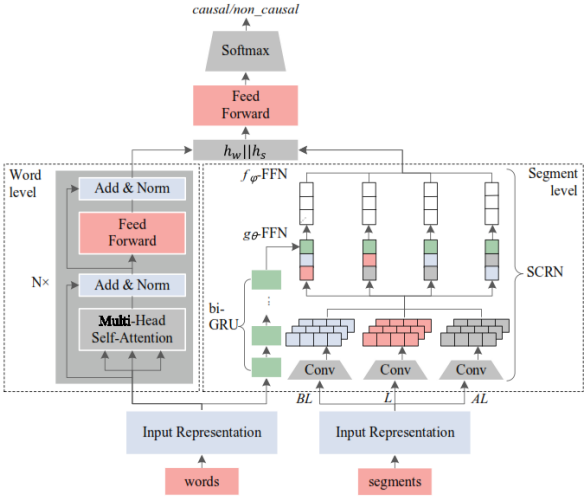


Figure 20: MCDN [47]

4.6.4 Knowledge Enhanced Event Causality Identification

(2020-[48]) This is an approach using an external knowledge base along with transformer encoders to build rich representations for an event-pair and to correspondingly classify it's causality. Note that this model expects the events spans have already been identified, not necessarily the event components though. Notable aspects of this model are the merging of external knowledge from a formal source, the novel method for building event-pair representations and the idea for the weighted combination of context only representations vs knowledge based enhanced representations.

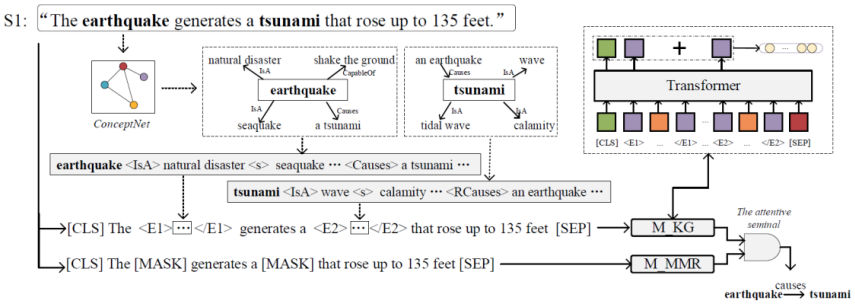


Figure 21: Knowledge Enhanced Event Causality Identification [48]

4.7 Inter-sentential RE

4.7.1 Discriminative Models for Inter-sentential RE

The specific issue of inter-sentential RE requires that the model can somehow encode contextual information over longer sequences of input text that a single sentence. Typically models work sentence by sentence, with transformer encoders like BERT having relatively small sequence length limits, which in-turn, limits the ability to include more inter-sentence context. Recent advances have developed encoder transformers able to ingest longer sequences than BERT, such as Longformer [49] 4096 tokens and BigBird [50] 4096 tokens. With these limits the simple approach would be to just use a context window to include text before and after the current sentence as was done by [41, 25] and this may well be sufficient as the vast majority of direct inter-sentential events are going to be within a few sentences of each other.

However, for longer range relationships, there would need to be some kind of strategy to gather relevant contextual signal from long distances:

- [38] use a hierarchical attention mechanism to generate document embeddings for cross sentence context.
- [51] proposed a GCN approach, building a whole document graph with links between sequential sentence graphs. The idea of adding in the sentence to sentence relationships to create document level graph could potentially be investigated.
- A more recent model (SENDIR) [52] uses LSTMs in addtion to BERT. SENDIR determines event representations with inter-sentential context via a combination intra-sentence BERT token representations and inter-sentence Bi-LSTM token representations. NOTE: the Bi-LSTM is used as it doesn't suffer the BERT sequence length limits. The Bi-LSTM takes in the BERT output token representations for all sentences in the document, as such, it adds inter-sentential contextual information to the token representations.

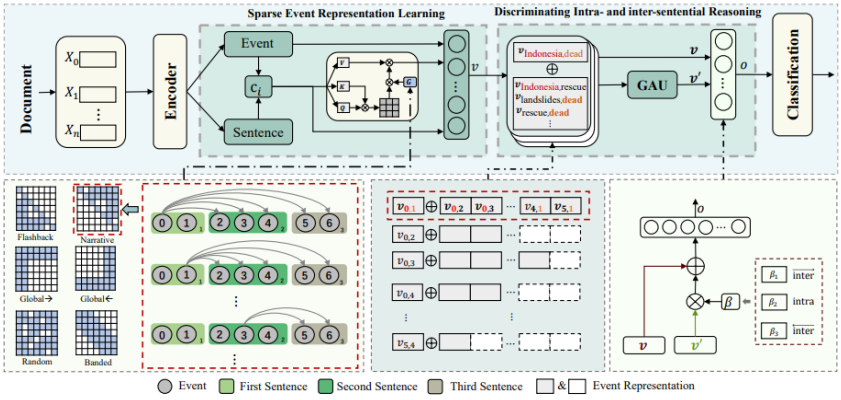


Figure 22: SENDIR [52]

4.7.2 Generative Transformers for Inter-sentential RE

Generative Transformers suffer from context window constraints, with the added issue for decoder only models that both the input and output tokens contribute to this context limit. This limit is increasing with some of the newer models [28], which potentially make it feasible to single prompt a model with an entire multi page report of text along with a comprehensive prompt including abundant examples and schema and still have ample context space for a long output.

However, the usefulness of the increased context window is currently limited due to the "lost in the middle problem" where the model seems to focus on the initial and last few 1000 tokens of a long document and gloss over the interior parts. This effect is well documented [53, 54] where they hypothesise that causes may range from training data bias, potentially issues with positional encoding, possible internal summarization

or attention window techniques to manage speed and compute resources. Fig 24 shows the performance of GPT-4o for extraction of 20 distributed unique keys for various clusterings of the keys (100 meaning all in the same area, 0 meaning highly distributed) vs corpus length. This shows the performance degrades as the corpus length increases and the distribution of the keys becomes more random.

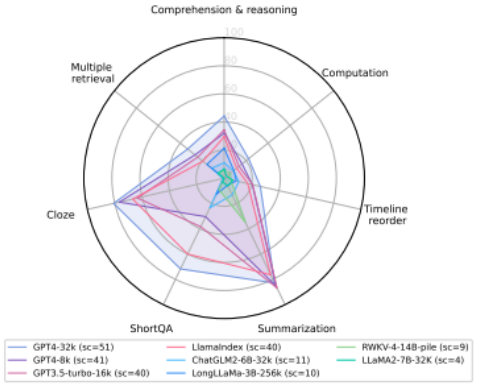


Figure 23: Generative Transformer Long Dependency Performance [53]

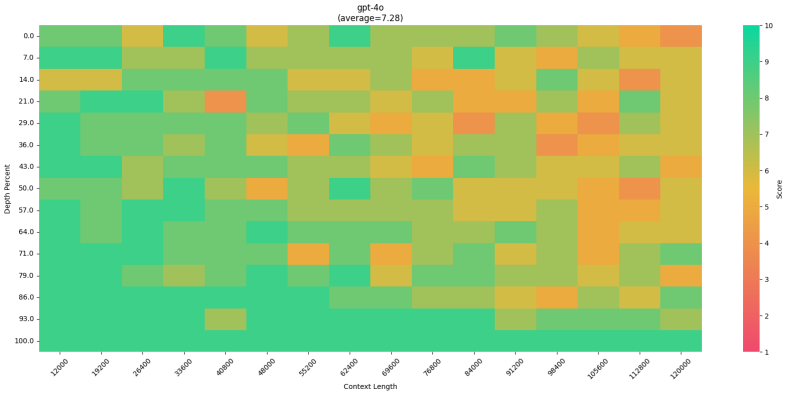


Figure 24: 20 needles in the haystack performance for GPT4o [54]

5 Datasets

This section summarises common IE datasets and following this there is a discussion on data synthesis and annotation with newer generative models.

5.1 Dataset Summary

The following table details some of the most prevalent datasets, which models use it, what tasks it is used for and links to the relevant paper and source.

Table 2: Summary of Datasets

Dataset	Models	UC*	NER	RE	EE	CRE	Source	Paper	Cost
ACE05	DeepStruct, DYGIE++, FourIE, GoLLIE, GraphER, InstructUIE, JMEE, PFN, PLmarker, PLMEE, RCEE, UIE, USM	✗	✓	✓	✓	✗	source	paper	$3100
ADE	CMAN, DeepStruct, InstructUIE, PFN, REBEL, SPERT	✗	✓	✓	✗	✓	source	paper	Free
AltLex	MCDN, SPLSTM	✓	✗	✗	✗	✓	source	paper	Free
Because 2	-	✓	✗	✗	✗	✓	source	paper	?Free?
CASIE	USM	✗	✓	✓	✓	✓	source	paper	Free
CausalTimeBank	-	✓	✗	✗	✓	✓	source	paper	Free
CoNLL04	CMAN, DeepStruct, GraphER, InstructUIE, REBEL, SPERT, UIE, USM	✗	✓	✓	✗	✗	source	paper	Free
DocRED	DREEAM, REBEL	✗	✓	✓	✗	✗	source	paper	Free
EventStoryLine	-	✓	✗	✗	✓	✓	source	paper	Free
Genia2011	DeepStruct, DYGIE++, InstructUIE	✗	✓	✗	✗	✗	source	paper	?Free?
NYT	DeepStruct, InstructUIE, PFN, REBEL, UniRel, USM	✗	✓	✓	✗	✗	source	paper	NA
Onto. 5	GoLLIE, InstructUIE, PLmarker	✗	✓	✗	✗	✗	source	paper	?Free?
PDTB	-	✓	✗	✗	✓	✓	source	paper	$760
Re-TACRED	EXOBRAIN, RAG4RE, REBEL	✗	✓	✓	✗	✗	source	paper	Free
SciERC	DYGIE++, GraphER, InstructUIE, PFN, PLmarker, SPERT, UIE, USM	✗	✓	✓	✗	✗	source	paper	Free
SemEval2010 T8	KEECI, RAG4RE, SP	✓	✓	✓	✗	✓	source	paper	Free
TACRED	DeepStruct, EXOBRAIN, RAG4RE, SP	✗	✓	✓	✗	✗	source	paper	35
WebNLG	PFN, UniRel	✗	✓	✓	✗	✗	source	paper	Free

*A recent (2023) resource in the realm of CRE specific datasets is UniCausal [55]. The authors have processed 6 causal datasets and released the 5 free ones on their github page with the goal of standardising the format of each of the component datasets. The 5 free datasets are: AltLex [21]; BECAUSE 2.0 [56]; CausalTimeBank [57]; EventStoryLine [58]; SemEval2010 T8 [59]. The single paid datset is PDTB [60].

5.2 Data Synthesis

Data Synthesis is when fake annotated data is generated for training an IE model. Generative Transformers, being text-in-text-out models, lend themselves to this task and could be utilised to supplement annotated datasets by generating synthesised data. However from some studies on data synthesis, some issues that can arise are domain shifts and quality problems. This can result in the need for sophisticated methods for quality control and also larger data quantity requirements for downstream model training [61, 62].

5.3 Data Annotation

Automated data annotation is another good fit for Generative Transformers, where the goal is to generate annotations for new data in larger quantity than a human could produce. An example is LLMaAA as described below.

5.3.1 LLMaAA

(2023-[29]) This is a data augmentation framework called LLMaAA (LLMs as Active Annotators) with the primary goal of efficiently increasing the supply of quality annotated datasets for IE models. They use an active sample selection approach using the performance of an associated downstream IE model as feedback to adjust the annotation sample selection process. Additionally a sample weighting mechanism is developed to minimise annotation noise on the downstream IE model. Lastly prompt engineering is proposed to select the most relevant gold samples as examples for the annotator for each sample to annotate.

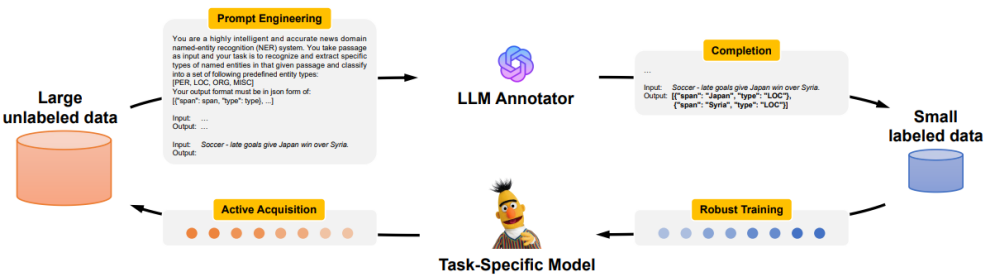
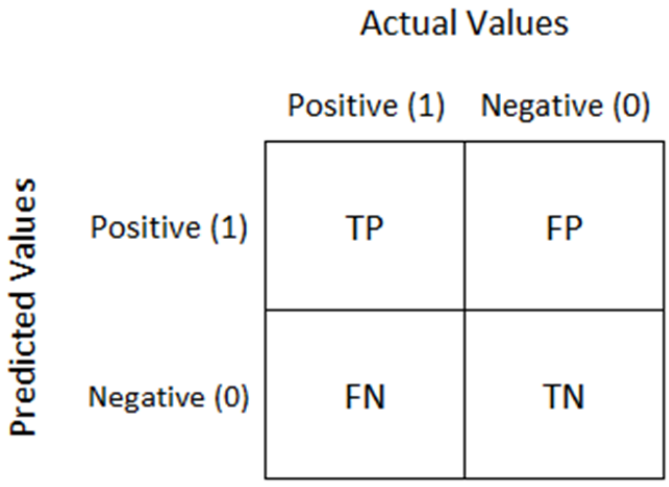


Figure 25: LLM Annotators [29]

6 Evaluation Metrics

As IE is primarily a discriminative task by nature, model performance viewed via classification metrics. The most common classification metric for IE models is F1 as the positive outcome is the focus and the positive classes are typically very sparse. Additionally, for multi-class scenarios, F1-macro will give a more balanced view of the F1 score (as opposed to F1-micro) as it averages across all classes irrespective of sample counts. If it is imperative to account for the True Negatives, then MCC can be a more balanced metric.



Accuracy (Acc): $= \frac{TP+TN}{TP+TN+FP+FN}$

Recall (TPR): $= \frac{TP}{TP+FN}$

Precision: $= \frac{TP}{TP+FP}$

F1 Score: $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Matthews Correlation Coefficient (MCC): $= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Figure 26: Classification Metrics

7 Discussion

Extracting causal relationships from textual data presents several significant challenges, particularly when dealing with varied linguistic structures, implicit relationships, and the lack of high-quality annotated data. Key challenges include:

- **Complexity of Causal Expressions:** Causal relationships can be varied in nature and in how they are described linguistically. In the simplest form, an explicit causal marker (eg. because) may relate a head and a tail pair, however there are many other more complex scenarios where this relationship is more complex, such as multiple causal heads or effect tails or a combination of causal heads and conditional heads etc. Additionally, confounding factors such as event/entity co-references, negation of causal markers (eg. not caused by) can further complicate the classification process.
- **Variability of Head and Tail Form:** While semantically the head and the tail of a causal relationship should be a state or event. Syntactically, these can take various forms, the most common being simple nouns/entities for both states and events (eg. **"the earthquake"** => *event as entity*; **"depressed economy"** => *state as entity*). However, another frequently encountered scenario is the complex structure for events with verbal triggers and a range of related arguments (eg. **"She dropped the glass"** => *complex event, trigger[dropped] + args[she, glass]*). Ideally the extraction system should handle all commonly encountered forms of head and tail.
- **Implicit Causal Relationships:** Many causal connections in text are implicit. At a minimum this requires use of surrounding contextual cues to identify causality. In some cases, it may require external word knowledge or prior context from the document being processed. This obviously adds to the complexity and reliability of a causal extraction model.
- **Inter-Sentential Relations:** Identifying causal relationships across sentences increases model complexity. For encoder-based discriminative transformer models such as BERT [13], the limited input sequence length can pose limitations for capturing these dependencies. This can be alleviated somewhat with newer variations such as Longformer [49] and Bigbird [50]. For decoder-based generative transformers, newer long context models are being developed, however there still appears to be issues (the cause being not currently well understood) with how well these models actually use this increased context window size. For both modelling scenarios, another issue is minimising relevant contextual information dilution when there is significant distance between head and tail spans.
- **Data Scarcity and Quality:** High-quality, large-scale annotated datasets for causal relation extraction are limited. This lack of annotated data hinders the development and fine-tuning of models. Moreover, synthetically generated datasets, while useful, can introduce domain shifts and noise, further complicating training. Bright spots are the potential to utilise powerful Generative Transformers for automated annotation.
- **Generative Transformer Hallucination:** While Generative Transformer models hold promise due to their capacity to capture nuanced relationships and ease of use, they are prone to hallucinating relationships, especially in the absence of clear causal indicators in the text. Managing and minimizing hallucinations remains a critical challenge for Generative Transformer based methods.
- **Dealing with False Data:** Causal relationships from text are just interpretations of what the writer was describing, it has no bearing on reality and whether there actually is causality. This 'fake' information effect is becoming more pronounced with the rise of unfiltered social media and other vectors for propagation of unverified information and opinion. A perfect causal relation extraction method is not in anyway a guarantee of actual causality.

8 Conclusion

Causal relation extraction remains a challenging but vital area of information extraction, with applications spanning decision-making systems, scenario generation, and knowledge graph construction. The evolution of methods from rule-based and machine learning approaches to deep learning and large language models has

significantly improved the ability to capture complex relationships. However, challenges persist, particularly in handling implicit causality, long-range dependencies, and the scarcity of high-quality training data.

Recent advancements in models like GraphER, PLmarker, and InstructUIE offer promising directions for addressing these challenges, particularly through innovations in attention mechanisms, extended context handling, and multi-task learning strategies. The emergence of Generative Transformers presents new opportunities for causal information extraction but also introduces challenges around efficiency and reliability, particularly in mitigating hallucinations.

Going forward, further research into scalable models that can effectively integrate external knowledge and reason over implicit causal structures will be crucial. Additionally, creating more comprehensive, annotated datasets and refining data augmentation techniques will help in training more robust models. These advancements could drive further improvements in extracting causality from text, ultimately enhancing downstream tasks such as event prediction, decision support systems, and automated reasoning.

References

[1] J. Cowie and W. Lehnert, "Information extraction," *Communications of the ACM*, vol. 39, no. 1, pp. 80–91, 1996.

[2] W. Ali, W. Zuo, W. Ying, R. Ali, G. Rahman, and I. Ullah, "Causality extraction: A comprehensive survey and new perspective," *College of Computer Science and Technology, Jilin University, China*, 2022.

[3] E. Riloff, "Autoslog: A system for domain-independent information extraction," in *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI)*, pp. 849–854, 1993.

[4] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *Proceedings of the fifth ACM conference on Digital libraries*, pp. 85–94, ACM, 2000.

[5] M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii, "Event extraction with complex event classification using rich features," *Journal of Bioinformatics and Computational Biology*, vol. 8, no. 01, pp. 131–146, 2010.

[6] D. Ahn, "The stages of event extraction," in *Proceedings of the Workshop on Annotating and Reasoning about Time and Events* (B. Boguraev, R. Muñoz, and J. Pustejovsky, eds.), (Sydney, Australia), pp. 1–8, Association for Computational Linguistics, July 2006.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] C. Kruengkrai, K. Torisawa, C. Hashimoto, J. Kloetzer, J.-H. Oh, and M. Tanaka, "Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017.

[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[10] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2524–2527, IEEE, 1997.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[12] A. Vaswani *et al.*, "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, (Long Beach, CA, USA), 2017.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[14] OpenAI, "Gpt-4 technical report," 2024.

[15] Meta, "The llama 3 herd of models," 2024.

[16] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[17] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," 2022.

[18] Linguistic Data Consortium, Philadelphia, PA, USA, *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 2005.

[19] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173111–173137, 2019.

[20] J. Liu, L. Min, and X. Huang, "An overview of event extraction and its applications," 2021.

[21] C. Hidey and K. McKeown, "Identifying causal relations using parallel wikipedia articles," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1424–1433, Association for Computational Linguistics, 2016.

[22] J. Yang, S. C. Han, and J. Poon, "A survey on extraction of causal relations from natural language text," 2021.

[23] K. Liu, Y. Chen, J. Liu, X. Zuo, and J. Zhao, "Extracting events and their relations from texts: A survey on recent research progress and challenges," *AI Open*, vol. 2, pp. 43–50, 2021.

[24] Z. Yan, Z. Jia, and K. Tu, "An empirical study of pipeline vs. joint approaches to entity and relation extraction," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, (Online), pp. 437–443, Association for Computational Linguistics, 2022.

[25] D. Ye, Y. Lin, P. Li, and M. Sun, "Packed levitated marker for entity and relation extraction," 2022.

[26] T. Knez and S. Žitnik, "Event-centric temporal knowledge graph construction: A survey," *Mathematics*, vol. 11, no. 23, p. 4852, 2023.

[27] Google, "Gemini: A family of highly capable multimodal models," 2024.

[28] Google, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," 2024.

[29] R. Zhang, Y. Li, Y. Ma, M. Zhou, and L. Zou, "Llmaaa: Making large language models as active annotators," 2023.

[30] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, "Large language models for generative information extraction: A survey," 2024.

[31] P. Li, T. Sun, Q. Tang, H. Yan, Y. Wu, X. Huang, and X. Qiu, "Codeie: Large code generation models are better few-shot information extractors," 2023.

[32] S. Zhao, M. Hu, Z. Cai, and F. Liu, "Modeling dense cross-modal interactions for joint entity-relation extraction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 4032–4038, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[33] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *Expert Systems with Applications*, vol. 114, p. 34–45, Dec. 2018.

[34] U. Zaratiana, N. Tomeh, N. E. Khbir, P. Holat, and T. Charnois, "Grapher: A structure-aware text-to-graph model for entity and relation extraction," 2024.

[35] J. Kim, D. T. Nguyen, S. Min, S. Cho, M. Lee, H. Lee, and S. Hong, "Pure transformers are powerful graph learners," in *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.

[36] B. Paassen, D. Grattarola, D. Zambon, C. Alippi, and B. E. Hammer, "Graph edit networks," in *International Conference on Learning Representations*, 2021.

[37] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, eds.), (Online), pp. 50–61, Association for Computational Linguistics, June 2021.

[38] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 414–419, Association for Computational Linguistics, July 2018.

[39] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018.

[40] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5284–5294, Association for Computational Linguistics, July 2019.

[41] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," 2019.

[42] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi, "A general framework for information extraction using dynamic span graphs," 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)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 3036–3046, Association for Computational Linguistics, June 2019.

[43] X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, Q. Zhang, T. Gui, J. Kang, J. Yang, S. Li, and C. Du, "Instructuie: Multi-task instruction tuning for unified information extraction," 2023.

[44] O. Sainz, I. García-Ferrero, R. Agerri, O. L. de Lacalle, G. Rigau, and E. Agirre, "Gollie: Annotation guidelines improve zero-shot information-extraction," 2024.

[45] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," 2022.

[46] E. Martínez-Cámara, V. Shwartz, I. Gurevych, and I. Dagan, "Neural disambiguation of causal lexical markers based on context," in *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers* (C. Gardent and C. Retoré, eds.), 2017.

[47] S. Liang, W. Zuo, Z. Shi, S. Wang, J. Wang, and X. Zuo, "A multi-level neural network for implicit causality detection in web texts," 2021.

[48] J. Liu, Y. Chen, and J. Zhao, "Knowledge enhanced event causality identification with mention masking generalizations," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 3608–3614, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[49] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020.

[50] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: Transformers for longer sequences," 2021.

[51] S. K. Sahu, F. Christopoulou, M. Miwa, and S. Ananiadou, "Inter-sentence relation extraction with document-level graph convolutional neural network," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 4309–4316, Association for Computational Linguistics, July 2019.

[52] C. Yuan, H. Huang, Y. Cao, and Y. Wen, "Discriminative reasoning with sparse event representation for document-level event-event relation extraction," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 16222–16234, Association for Computational Linguistics, July 2023.

[53] J. Li, M. Wang, Z. Zheng, and M. Zhang, "LooGLE: Can long-context language models understand long contexts?," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 16304–16333, Association for Computational Linguistics, Aug. 2024.

[54] O. C. User, "Reasoning degradation in llms with long context windows." `https://community.openai.com/t/reasoning-degradation-in-llms-with-long-context-windows-new-benchmarks/906891/5`, 2024. Accessed: 2024-09-13.

[55] F. A. Tan, X. Zuo, and S. Ng, "Unicausal: Unified benchmark and repository for causal text mining," in *Big Data Analytics and Knowledge Discovery - 25th International Conference, DaWaK 2023, Penang, Malaysia, August 28-30, 2023, Proceedings* (R. Wrembel, J. Gamper, G. Kotsis, A. M. Tjoa, and I. Khalil, eds.), vol. 14148 of *Lecture Notes in Computer Science*, pp. 248–262, Springer, 2023.

[56] J. Dunietz, L. Levin, and J. Carbonell, "The because corpus 2.0: Annotating causality and overlapping relations," in *Proceedings of the 11th Linguistic Annotation Workshop*, (Valencia, Spain), pp. 95–104, Association for Computational Linguistics, 2017.

[57] P. Mirza, R. Sprugnoli, S. Tonelli, and M. Speranza, "Annotating causality in the TempEval-3 corpus," in *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)* (O. Kolomiyets, M.-F. Moens, M. Palmer, J. Pustejovsky, and S. Bethard, eds.), (Gothenburg, Sweden), pp. 10–19, Association for Computational Linguistics, Apr. 2014.

[58] T. Caselli and P. Vossen, "Eventstoryline: Creating a cross-lingual event-centric timeline," in *Proceedings of the 1st Workshop on Computing News Storylines (CNS 2017)*, (Vancouver, Canada), pp. 1–11, Association for Computational Linguistics, 2017.

[59] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. O'Seaghdha, S. Pado, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, (Uppsala, Sweden), pp. 33–38, Association for Computational Linguistics, 2010.

[60] R. Prasad, N. Dinesh, A. Lee, A. Joshi, and B. Webber, "Attribution and its annotation in the penn discourse treebank," in *Proceedings of the Linguistic Annotation Workshop*, (Prague, Czech Republic), pp. 31–38, Association for Computational Linguistics, 2007.

[61] J. Gao, R. Pi, L. Yong, H. Xu, J. Ye, Z. Wu, W. ZHANG, X. Liang, Z. Li, and L. Kong, "Self-guided noise-free data generation for efficient zero-shot learning," in *The Eleventh International Conference on Learning Representations*, 2023.

[62] Y. Meng, J. Huang, Y. Zhang, and J. Han, "Generating training data with language models: Towards zero-shot language understanding," in *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.