# Transformer-Based Architectures versus Large Language Models in Semantic Event Extraction: Evaluating Strengths and Limitations

Tin Kuculo [a,*], Sara Abdollahi [a] and Simon Gottschalk [a]

[a] *L3S Research Center, Leibniz Universität Hannover, Hannover, Germany*
*E-mails: kuculo@L3S.de, abdollahi@L3S.de, gottschalk@L3S.de*

**Abstract.** Understanding complex societal events reported on the Web, such as military conflicts and political elections, is crucial in digital humanities, computational social science, and news analyses. While event extraction is a well-studied problem in Natural Language Processing, there remains a gap in semantic event extraction methods that leverage event ontologies for capturing multifaceted events in knowledge graphs since existing methods for event extraction often fall short in the semantic depth or lack the flexibility required for a comprehensive event extraction.

In this article, we aim to compare two paradigms to address this task of semantic event extraction: The fine-tuning of traditional transformer-based models versus the use of Large Language Models (LLMs). We exemplify these paradigms with two newly developed approaches: `T-SEE` for transformer-based and `L-SEE` for LLM-based semantic event extraction. We present and evaluate these two approaches and discuss their complementary strengths and shortcomings to understand the needs and solutions required for semantic event extraction.

For comparison, both approaches employ the same dual-stage architecture; the first stages focus on multilabel event classification, and the second on relation extraction. While our first approach utilises a span prediction transformer model, our second approach prompts an LLM for event classification and relation extraction, providing the potential event classes and properties. For evaluation, we first assess the performances of `T-SEE` and `L-SEE` on two novel datasets sourced from Wikipedia, Wikidata, and DBpedia, containing over $80,000$ sentences and semantic event representations. Then, we perform an extensive analysis of the different types of errors made by these two approaches to discuss a set of phenomena relevant to semantic event extraction.

Our work makes substantial contributions to (i) the integration of Semantic Web technologies and NLP, particularly in the underexplored domain of semantic event extraction, and (ii) the understanding of how LLMs can further enhance semantic event extraction and what challenges need to be considered in comparison to traditional approaches.

Keywords: Event Extraction, Transformer Models, Large Language Models, Event Knowledge Graph

## 1. Introduction

Event extraction aims to identify and classify events and their relations in text, including Web sources such as social media, news websites, and online encyclopedias like Wikipedia. Typically, this extraction process is conducted without relying on pre-existing knowledge structures or further structuring of extracted data. In contrast, the goal of

*Corresponding author. E-mail: kuculo@L3S.de.

*semantic event extraction* is to leverage an existing event ontology to lift unstructured text into a structured representation capturing the essence of the event, including its type (e.g., `presidential election`) and relations to entities (e.g., <`US presidential election 2020`, `successful candidate`, `Joe Biden`>). Specifically, semantic event extraction aims at enriching knowledge graphs to make event information more accessible, i.e., by adding events that are not yet contained in the knowledge graph because (i) the input texts are about recent events or (ii) the events of that type are considered out of domain (e.g. if the knowledge graph only contains more coarse-grained event types). Practical applications of event knowledge graphs include event-centric visualisations [1, 2], biography generations [3], event narrativisation [4] and question answering over event-related information [5].

Semantic extraction operates at a critical juncture of the Semantic Web and Natural Language Processing (NLP) technologies:

– The Semantic Web offers rich event ontologies such as LODE [6] and the Simple Event Model [7] to represent events. However, cross-domain knowledge graphs such as DBpedia [8] and Wikidata [9] typically focus on named events, such as political summits and natural disasters and lack adaptability to diverse expressions in text-based event descriptions. In addition, relation extraction and link prediction for knowledge graph population typically suffer from noisy data [10–12] and require the presence of the related entities in the knowledge graph [13] and are thus not applicable for extracting relations of newly identified events.
– NLP employs named entity recognition and event extraction techniques to identify finer-grained, transient events like individual meetings or transactions [14] from text. However, traditional NLP methods often deconstruct the task of semantic event extraction into smaller sub-tasks such as event detection [15, 16], and argument extraction [17–19] with each garnering their specific benchmark datasets [20, 21] typically not bound to semantic event ontologies.

This divergence results in a critical gap, creating a need for *semantic event extraction*, blending structured, ontology-based classification with the adaptability to handle a wide range of event types – from transient interaction to significant historical occurrences.

Although some efforts have been made towards semantic event extraction [22, 23], Guan et al. denote that the construction of event knowledge graphs still suffers from the unsatisfactory performance of existing event extraction methods, especially for argument extraction [24]. Most methods still fall short in delivering an integrative approach that works across various domains and effectively accommodates sufficiently rich and diverse ontologies [25–27] centring instead around aged NLP benchmark datasets such as *ACE05* [28] or conversely on highly specific domains [29, 30].

**Example:** As an example of semantic event extraction, consider the event represented in Fig. 1. The text on the left is extracted from the Wikipedia article regarding the "2017 UEFA European Under-21 Championship Final". We aim to extract relevant event information[1] from that text, such as the final match itself or, potentially, other events mentioned in the text, and enrich an event knowledge graph with newly extracted events and event relations. The right-hand side of the figure illustrates a knowledge graph representation of an extracted event. This representation includes an event class (`final`), an event description derived from the text, the precise location of the game, the date, and other relations.

In this article, we introduce two approaches for semantic event extraction, which follow the same structure but two different paradigms: Transformer-based architectures and Large Language Models (LLMs).[2]

**Transformer-based Semantic Event Extraction (`T-SEE`):** `T-SEE` benefits from the strengths of both Semantic Web and NLP techniques and is trained and evaluated on two new datasets, specifically created as a resource for semantic event extraction. `T-SEE` disentangles the complexities of the task into two manageable sub-tasks:

[1]Given an event and its event class, we consider any information that can be expressed with a property typically used on the respective event class as relevant (e.g., the type of sport of a final).

[2]While LLMs also employ transformers, in this article, we refer to the "traditional" use of transformers, which are fine-tuned to a specific target task, and compare them to pre-trained LLMs prompted for the target task.

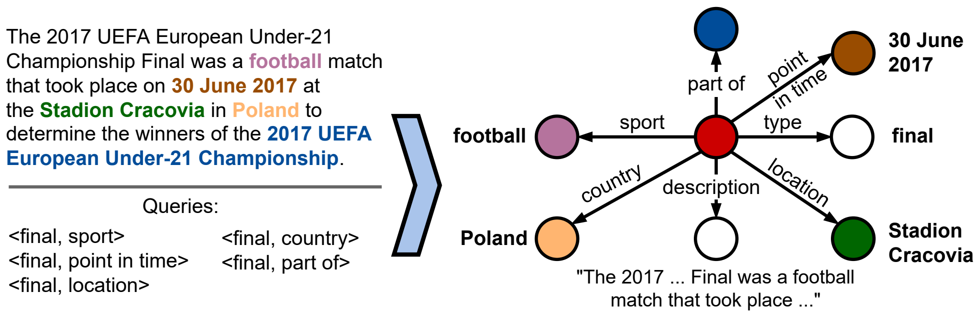


Fig. 1. Example of semantic event extraction for an event mentioned in the Wikipedia article "2017 UEFA European Under-21 Championship Final".

- Event Classification: Approached as a multilabel classification problem, `T-SEE` determines the most appropriate event labels given a text from a pre-defined set of event classes. In our example, `T-SEE` applies multilabel classification to categorise the event into `final`.
- Relation Extraction: Utilising a span prediction transformer model, we target class-specific relations to construct a nuanced representation of events. In our example, we extract relations such as <`country`, `Poland`> employing a set of queries.

**LLM-based Semantic Event Extraction (`L-SEE`):** With `L-SEE`, we examine the application of LLMs for semantic event extraction. Given the current prominence of LLMs in various NLP tasks [31, 32], it is pertinent to assess their utility and performance in extracting structured event information from text. In analogy to `T-SEE`, `L-SEE` also performs event classification followed by relation extraction, both through specific prompts.

**Evaluation:** To train `T-SEE` and to evaluate `T-SEE` and `L-SEE`, we provide two new semantic event extraction datasets created from Wikipedia, Wikidata, and DBpedia, containing over $80,000$ sentences and semantic event representations. Through a subsequent manual error analysis, we not only aim to gauge the capabilities of LLMs against transformer-based methods but also to identify specific challenges and areas where LLMs might offer novel insights or complement existing approaches. In this way, we aim to contribute to the ongoing discourse on the potential and limitations of leveraging LLMs for information extraction and knowledge engineering, particularly in cases where LLMs may uncover information beyond the predefined ground truth or existing knowledge graphs.

**Contributions:** In summary, our contributions are:

- We outline the underexplored area of semantic event extraction, situated at the Semantic Web and NLP intersection.
- We present `T-SEE` and `L-SEE`, our approaches for semantic event extraction following comparable pipelines, where `T-SEE` uses a transformer-based architecture and `L-SEE` uses an LLM.
- We provide two new semantic event extraction datasets created from Wikipedia, Wikidata, and DBpedia: *Wikidata-SEE* and *DBpedia-SEE*.
- We demonstrate the efficacy of `T-SEE` and `L-SEE` through empirical evaluations against existing methods.
- We perform an extensive manual annotation of the predictions of `T-SEE` and `L-SEE` to identify typical error types and compare the strengths and shortcomings of these two paradigms.
- We make the code[3] and the data[4] available online.

**Structure:** The remainder of this article is structured as follows: In Section 2, we define the task of semantic event extraction. Then, we introduce `T-SEE` (Section 3) and `L-SEE` (Section 4). After an automated evaluation of these approaches on a test set (Section 5), we perform our error analysis and discussion in Section 6. After presenting related work (Section 7), we conclude in Section 8.

[3]https://github.com/t-kuculo/T-SEE
[4]https://zenodo.org/records/10818676

## 2. Problem Statement

We formally define the problem of semantic event extraction to bridge the gap between granular, structured information and the adaptability required to capture a wide variety of events.

In the context of this work, an *event* is an occurrence of societal importance, typically happening at a specific time and location, involving a set of participants. Examples of events include military conflicts, such as the Second World War, political shakeups, such as Brexit, but also more fine-grained events, such as the battles and air raids in the Second World War or specific football games.

We model information regarding entities (representing real-world events and real-world objects such as persons or locations) and their relations in an event knowledge graph. The classes and properties within the knowledge graph are defined by an event ontology:

**Definition 1** (Event Ontology)**.** *An event ontology $O = (P, C)$ defines the properties and classes in an event knowledge graph.*

- *$P$ is a set of properties describing the types of relations that can hold between two entities.*
- *$C$ is a set of event classes. An event class can be a sub-class of another event class.*

Classes and properties in an event ontology are uniquely identified by an Internationalized Resource Identifier (IRI).[5] Specifically, the property $p_{\text{type}} \in P$ (typically identified via the property IRI `rdf:type`) assigns an event class to an event.

Other example properties describe the `location` and `number of participants` of events. Examples of event classes include `final` as a sub-class of `sporting event`.

Based on an event ontology, we formally define an event knowledge graph as follows:

**Definition 2** (Event Knowledge Graph)**.** *An event knowledge graph $G_O = (E, V, L, R)$ models entities, events, literals, and their relations following an event ontology $O = (P, C)$:*

- *$E$ is a set of nodes representing real-world entities.*
- *$V \subset E$ is a subset of nodes representing real-world events.*
- *$L$ is a set of literals such as numbers or texts.*
- *$R = E \times P \times E \cup L \cup C$ is a set of relations, i.e., edges.*

Using the property $p_{\text{type}}$, an event $e \in V$ can be assigned a class $c \in C$ through the following relation $r \in R$: $(e, p_{\text{type}}, c)$.

We define the task of *semantic event extraction* as follows:

**Definition 3** (Semantic Event Extraction)**.** *Given an event ontology $O = (P, C)$, an event knowledge graph $G_O = (E, V, L, R)$ and a text $t$, detect a set of events $V_t$ described in $t$ which are not yet in $G_O$. For each event $e_t \in V_t$, identify its event class $e_t.c \in C$ (event classification) and extract a set of relations from $t$ (relation extraction). These relations, and the classes they involve, adhere to the properties and classes of $O$.*

Fig. 1 illustrates an example text ($t$) taken from the Wikipedia article regarding the "2017 UEFA European Under-21 Championship Final". The semantic event extraction leads to the creation of a new event $e_t \in V_t$, which is typed as the event class `final` and assigned to relations with properties of the event ontology $O$ (e.g., `location` and `point in time`). These relations can be serialised as RDF triples to be used in downstream applications.

## 3. T-SEE: Transformer-based Semantic Event Extraction

In this section, we present `T-SEE` (<u>T</u>ransformer-based <u>S</u>emantic <u>E</u>vent <u>E</u>xtraction), an approach for semantic event extraction based on a transformer architecture. The design of `T-SEE` is guided by two goals:

[5] Relevant prefixes and namespaces of IRIs used in this article include: `wd`: http://www.wikidata.org/entity/, `wdt`: http://www.wikidata.org/prop/direct/, `dbr`: https://dbpedia.org/resource/, `dbo`: https://dbpedia.org/ontology/ and `rdf`: http://www.w3.org/1999/02/22-rdf-syntax-ns#.

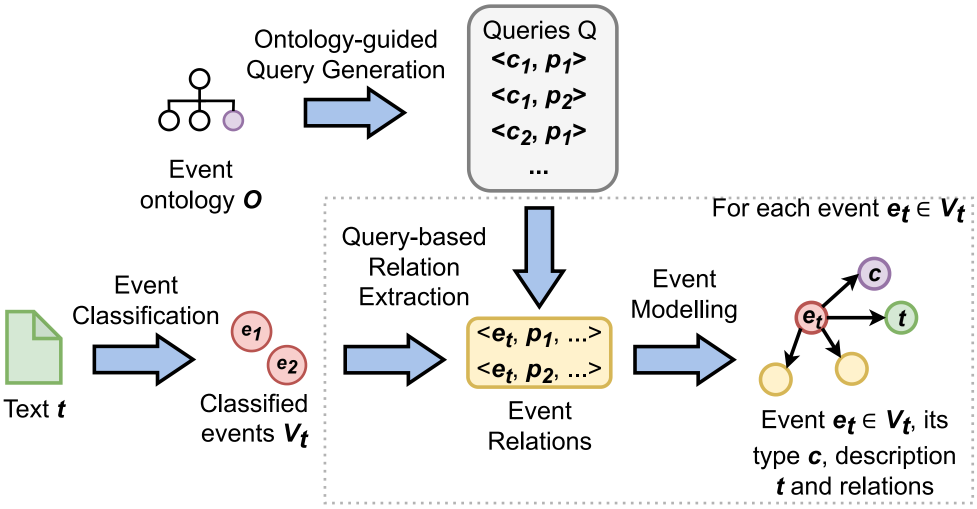


Fig. 2. Overview of T-SEE, showing how it extracts and models a single event.

1. Through a 3-step procedure of event classification, relation extraction and event modelling, we ensure comparability with L-SEE, our LLM-based approach presented in the next section (Section 4).
2. To allow seamless integration into the Semantic Web, the whole architecture of T-SEE needs to be guided through an event ontology, its RDF classes and properties.

Fig. 2 offers a visual summary of T-SEE following these goals. Given an event ontology $O$, we generate a set of queries $Q$ during the preprocessing phase (*query generation*) as a basis of the query-based relation extraction. T-SEE then carries out a three-step process to extract and semantically represent events from a given text $t$:

1. *Event classification*: We formulate event classification as a multilabel classification problem and apply it to a given text $t$ to identify event mentions and their classes. This enables us to classify all event mentions within the text concurrently.
2. *Query-based relation extraction*: For each identified event, we extract its relations using a transformer-based extraction model and a subset of $Q$, i.e., selected queries used to extract relevant relations of the detected events. After the appropriate queries have been selected, we train our relation extraction model on pairs of event classes and properties.
3. *Event modelling*: We transform the extracted event information into triples and add them to the event knowledge graph $G_O$.

With this process, T-SEE focuses on event classification and subsequent event relation extraction, aiming to generate a robust and comprehensive representation of event knowledge. We build on three key factors: (i) the inherent strengths of transformer models, including their capacity to encapsulate complex semantic relationships within the text; (ii) the use of task-specific fine-tuning of these models that allows us to tailor their powerful general language understanding capabilities to our specific extraction tasks, and (iii) the structural guidance provided by an event ontology, which not only aligns the model's understanding of events with existing schemas but also offers adaptability accommodating emerging event types, such as "pandemic".

In the following, we describe T-SEE's steps in more detail, along with its algorithm and a running example for a more intuitive understanding.

**Algorithm:** Algorithm 1 provides an overview of T-SEE. The algorithm embodies the three main inference steps explained earlier, namely event classification, query-based relation extraction, and event modelling.

**Example:** We exemplify each of the steps based on the example illustrated in Fig. 3, where the text $t$ pertains to protests in Tehran. T-SEE extracts two events ($e_{t_1}$ and $e_{t_2}$), their classes[6] (conflict and revolution)

[6]As our event ontology $O$ in this example, we select an event ontology extracted from Wikidata.

**Algorithm 1** Transformer-based Semantic Event Extraction (`T-SEE`)

**Input**
    $t$ Text
    $O$ Event ontology
    $R$ The relations in an event knowledge graph $G_O$
    $Q$ Set of queries
    $ECM$ Event Classification Model (trained)
    $REM$ Relation Extraction Model (trained)

$V_t \leftarrow ECM.classifyEvents(t, O))$ ▷ Event classification (Section 3.2)

**for each** $e_t \in V_t$ **do** ▷ Query-based relation extraction (Section 3.3)
    $R_{e_t} \leftarrow []$
    **for each** $q \in getQueries(Q, e_t.c)$ **do**
        $result \leftarrow REM.getQueryResult(t, q)$
        $R_{e_t} \leftarrow R_{e_t} \cup$
          $REM.createRelations(e_t, q.p, result)$

    $R = R \cup \ (e_t, p_{type}, c)$ ▷ Event modelling (Section 3.4)
        $\cup \ (e_t, p_{description}, t) \ \cup \ R_{e_t}$

and relations. This example demonstrates how `T-SEE`'s relation extraction model is capable of extracting different relations for each detected event, for instance, ($e_{t_1}$, $p_{participant}$, Government of Iran)[7] and ($e_{t_2}$, $p_{location}$, Tehran).

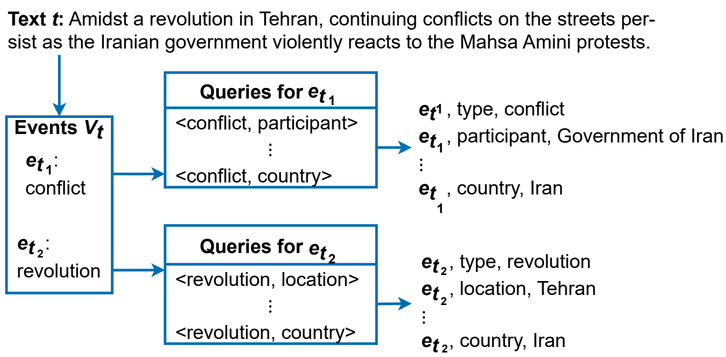


Fig. 3. Example of event classification and query-based relation extraction on a sentence in the Wikipedia article "Mahsa Amini protests".

### 3.1. Ontology-guided Query Generation

The query generation step is a preprocessing step that creates a set of queries $Q$ used later as input to the query-based relation extraction model. The generation of $Q$ is guided by an event ontology $O$, such that each query $q = \langle c, p \rangle \in Q$ comprises the event class $c$ and a corresponding property $p$ as defined in the event ontology. For each considered event class in $O$[8], a set of queries is added to $Q$. These queries are then used in `T-SEE`'s query-based relation extraction step.

[7] For readability of the example relations, we represent selected entities and classes through their labels.
[8] We consider all event classes and properties in the event ontology $O$ that appear in the training data.

Given an event ontology $O = (P, C)$ and an event knowledge graph $G_O = (E, V, L, R)$, we create these queries as follows: For each event class $c \in C$, we select a set of properties that are used together with events of this class in $G_O$: $\{p|(e, p, x) \in R \wedge (e, p_{type}, c) \in R\}$. To avoid the inclusion of inappropriate queries (e.g., infrequent event classes and metadata properties), additional constraints can be applied to remove queries from $Q$. We describe our constraints in Section 5.1 and make our sets of event classes, properties, and queries available[9].

### *3.1.1. Example*

Fig. 4 shows an example Wikidata SPARQL query to extract Wikidata properties commonly (more than 50 times) used on entities classified as `wd:Q180684` (conflict). It returns 22 properties, including `wdt:P17` (country) and `wdt:P710` (participant).

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?property WHERE {
  ?s ?property ?o .
  ?s wdt:P31 wd:Q180684 . # instance of conflict
  FILTER(STRSTARTS(STR(?property ), STR(wdt:))) .
} GROUP BY ?property HAVING(COUNT(?s) > 50)
```

Fig. 4. SPARQL query on Wikidata to extract Wikidata properties commonly (more than 50 times) used on entities classified as “conflict”.

Using such SPARQL queries, we can generate queries used by `T-SEE`. Table 1 provides examples of four such queries for two Wikidata event classes, each together with their properties.

Table 1
Example queries extracted from the Wikidata ontology.

| **Event class** ($c$) | **Property** ($p$) | **Query** ($q = \langle c, p \rangle$) |
|---|---|---|
| conflict (`wd:Q180684`) | participant (`wdt:P710`) | <conflict, participant> |
| | country (`wdt:P17`) | <conflict, country> |
| revolution (`wd:Q10931`) | location (`wdt:P276`) | <revolution, location> |
| | country (`wdt:P17`) | <revolution, country> |

## *3.2. Event Classification*

Given a text $t$, the goal of `T-SEE`'s event classification step is to identify a set of events $V_t$ that occur in $t$ and to detect their event classes (line 9 in Algorithm 1). To do so, we propose a multilabel event classification model based on a transformer architecture [33], which allows for the efficient and effective processing of input texts.

Specifically, the input to our event classification model is a sequence of tokens derived from $t$ representing one or more event mentions in the text. The model processes the input sequence using a series of self-attention mechanisms, allowing it to capture complex relationships between contextual and semantic information of the input $t$. The output of the transformer-based architecture is a sequence of hidden states, which encodes the relevant information from the input sequence.

The hidden states are then passed through a dropout layer to reduce the number of connections between the pre-trained layers and the downstream layers, effectively forcing the downstream layers to learn more robust and generalisable representations of the input data. Finally, a fully-connected layer and a Sigmoid activation function are used in the output layer, generating a probability distribution over the possible event classes in the input text.

[9] https://github.com/t-kuculo/T-SEE/blob/main/processing/filtered_wikidata_event2.schema

Additionally, we conduct threshold optimisation on a validation set. Traditional multilabel classification approaches often employ a fixed decision threshold (usually 0.5) to convert predicted probabilities into class labels. However, this may not be optimal for all classes, especially in cases with imbalanced data or differing class complexities. To address this issue, we utilise an optimisation strategy that fine-tunes individual decision thresholds for each label, aiming to maximise the $F_1$ score.

#### 3.2.1. Example

In our example, the event classification model receives the whole text shown in Fig. 3 ("Amidst a revolution in Tehran, continuing conflicts on the streets persist as the Iranian government violently reacts to the Mahsa Amini protests.") as an input and returns two event classes (`conflict` and `revolution`) corresponding to the two events in the text.

#### 3.2.2. Training

To train `T-SEE`'s event classification model, a corpus that contains texts and event class labels corresponding to the events represented in each individual text is required. Specifically, we utilise two datasets that contain sentences from Wikipedia, annotated with events and their relations from Wikidata and DBpedia, respectively. These datasets are described in detail in Section 5.1. The multilabel classification model is fed the tokenised input texts and uses a focal loss function [34].

### 3.3. Query-based Relation Extraction

Given the text $t$ and the set of detected events $V_t$ together with their predicted event classes, the goal of relation extraction is to detect, extract, and assign relations found in $t$ to the matching events.

`T-SEE` utilises a subset of the generated queries $Q$ that can be matched to the predicted event classes of the extracted events $V_t$. Specifically, we only consider those queries $\{q = \langle c, p\rangle \in Q | \exists e_t \in V_t \text{ such that } e_t.c = c\}$ covering the respective event classes (line 13 in Algorithm 1). Together with $t$, these queries serve as input to our query-based relation extraction model.

We leverage BERT [35] as the base of `T-SEE` as it provides a nuanced understanding of semantics, capturing the meaning and context of words and sentences in text. BERT is known for its proficiency in capturing long-range dependencies, a crucial aspect of comprehending the complexities of textual narratives. In addition, BERT incorporates a Next Sentence Prediction loss, which is specifically designed to model the coherence between sentences. This element of coherence is particularly valuable for relation extraction tasks. By understanding the continuity of text, the model is empowered to decipher the intricate relationships between entities that might be scattered across the text.

Specifically, we encode the text $t$ and a query $q$ as fixed-length vectors. The decoded results then correspond to a probability distribution over token spans that represent possible relation values.

As shown in line 14 of Algorithm 1, each selected query $q = \langle c, p\rangle$, and context represented by the text $t$ are passed through our query-based relation extraction model, generating results and their associated confidence scores. Together with the respective event and the property $p$, each result resembles a relation.

#### 3.3.1. Example

For our predicted event classes `conflict` and `revolution`, the queries in $Q$ cover a variety of Wikidata properties such as `participant` and `location`. As shown in Fig. 3, given the query `<revolution, location>`, we infer its result "Tehran", i.e., the relation $(e_{t_2}, p_{location}, \text{Tehran})$. This process is repeated for each query-context pair, creating, for each accepted result, a relation.

#### 3.3.2. Training

To train our query-based relation extraction model, we use a corpus of texts with event mentions and their relations with properties in the event ontology $O$. As in [36], the model is jointly trained using a span extraction loss and a logistic regression loss for an additional classifier that predicts answerability [37, 38]. During training, the model is rewarded for selecting token spans that correspond to correct relation values between an event of a given event class label and entities or literals that occur in the text.

### 3.4. Event Modelling

In the event modelling step, we materialise the extracted event information as triples and enrich the event knowledge graph with them (line 18 - 19). Precisely, for each text $t$, and each of its events $e_t \in V_t$, its class, and relations, we create the following relations:

- Type relation for $e_t$: $(e_t, p_{type}, c)$
- Description of $e_t$: $(e_t, p_{description}, t)$
- Relations extracted with our query-based relation extraction

This process is repeated for all texts in an input corpus and the events extracted within them, after which the ontology-mapped relations can be transformed into RDF triples. As described in Definition 3, the event modelling step creates new triples of events not yet represented in the target knowledge graph $G_O$. Event classes, properties and their values were identified in the extraction process guided by the event ontology $O$.

For representing the provenance and explicitly providing the source of the semantic event representation, further information could be added, e.g., a URL pointing to the source text and a description of the extraction method. To do so, sources can be directly linked to a source statement in Wikidata[10]. Another option would be to use the PROV-O ontology [39].

#### 3.4.1. Example

Fig. 3 illustrates relations extracted for the example events `conflict` and `revolution`. Given the `conflict` event, the following relations are created:

- $(e_{t_1}, p_{type}, \texttt{conflict})$
- $(e_{t_1}, p_{description},$"Amidst a revolution in Tehran, continuing conflicts on the streets persist as the Iranian government violently reacts to the Mahsa Amini protests.")
- $(e_{t_1}, p_{participant},$ Government of Iran)
- $(e_{t_1}, p_{country},$ Iran)

We provide examples of generated RDF triples in Section 5.6.

## 4. `L-SEE`: LLM-based Semantic Event Extraction

In this section, we present `L-SEE` (LLM-based Semantic Event Extraction), an approach for semantic event extraction based on a Large Language Model. As LLMs continue redefining the boundaries of NLP, their application in semantic event extraction presents a compelling approach for assessing their standalone capabilities and potential synergies with traditional methodologies.

Fig. 5 offers a visual summary of `L-SEE` whose bottom part is analogous to `T-SEE` in Fig. 2. Given an event ontology $O$, the set of all event classes $C$ is extracted beforehand. As in `T-SEE`, `L-SEE` then carries out a three-step process to extract and semantically represent events from a given text $t$:

1. *Event classification*: We perform event classification as a multilabel classification problem by prompting an LLM to detect events and their classes in a text $t$ given $C$.
2. *Relation extraction*: We prompt the LLM to extract relations of all identified events.
3. *Event modelling*: We transform the extracted event information into triples and add them to the event knowledge graph $G_O$.

**Algorithm:** Algorithm 2 provides an overview of `L-SEE` and its three steps: event classification, relation extraction and event modelling.

[10] https://www.wikidata.org/wiki/Help:Sources

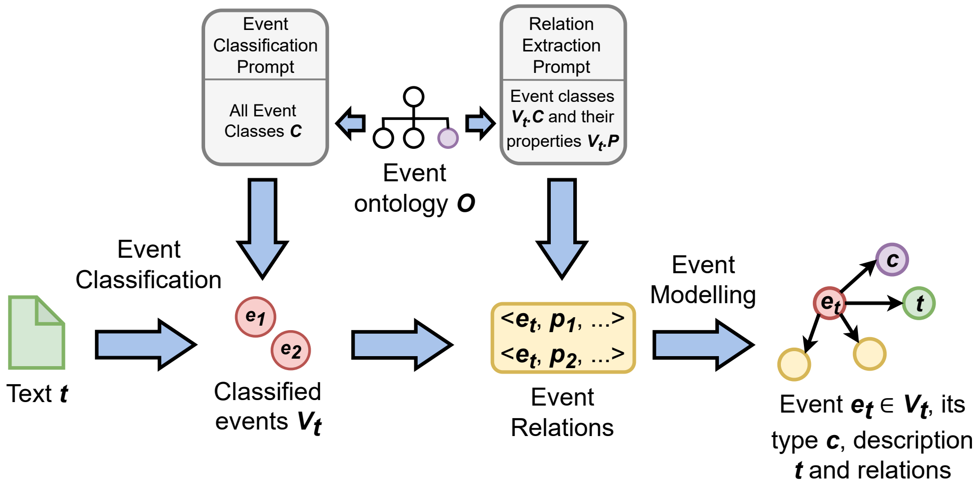


Fig. 5. Overview of `L-SEE`, showing how it extracts and models a single event.

**Algorithm 2** LLM-based Semantic Event Extraction (`L-SEE`)

1: **Input**
2: $t$ Text
3: $O$ Event ontology
4: $R$ The relations in an event knowledge graph $G_O$
5: $LLM$ Large Language Model (pre-trained)
6:
7: $C \leftarrow getAllEventClasses(O)$
8: $V_t \leftarrow LLM.classifyEvents(t, C)$ ▷ Event classification (Section 4.1)
9:
10: $V_t.P \leftarrow getPropertiesOfClasses(O, V_t.C)$
11: $R_{e_t} \leftarrow LLM.getRelations(t, V_t.C, V_t.P)$ ▷ Relation extraction (Section 4.2)
12:
13: **for each** $e_t \in V_t$ **do**
14: $R = R \cup \ (e_t, p_{type}, c)$ ▷ Event modelling (Section 4.3)
15: $\cup \ (e_t, p_{description}, t) \ \cup \ R_{e_t}$

### 4.1. Event Classification

For event classification (line 8 in Algorithm 2), `L-SEE` guides the LLM with a precise prompting mechanism to identify and categorise events in the text $t$, given the set $C$ of event classes in the target event ontology. This step builds upon the LLM's ability to discern events of significance akin to those warranting dedicated Wikipedia entries, ensuring the extraction of events with substantial relevance.

The event classification LLM prompt template is shown in Fig. 16 in the Appendix, where $C$ is formatted like `['conflict', 'revolution']`.

### 4.2. Relation Extraction

For relation extraction (line 11 in Algorithm 2), `L-SEE` prompts the LLM a second time, now to extract the relations of each identified event, given the event classes $V_t.C$ identified in the previous step together with the set of properties $V_t.P$ used on these classes (extracted as described in Section 3.1).

The relation extraction LLM prompt template is shown in Fig. 17 in the Appendix. For our condensed example in Table 1, $V_t.C$ and $V_t.P$ would be added to the prompt formatted as `{ 'conflict': ['participant', 'conflict'], 'revolution': ['location', 'country'] }`.

### 4.3. Event Modelling

The event modelling step (line 14 - 15 in Algorithm 2) follows the procedure outlined in `T-SEE`, as detailed in Section 3.4. This process results in the creation of RDF triples that represent newly identified events and their relations.

## 5. Evaluation

In this section, we introduce two new datasets for semantic event extraction and compare `T-SEE` and `L-SEE` to event extraction baselines. Finally, we show an example of the generated RDF triples.

### 5.1. Datasets

We introduce two new large-scale datasets that currently stand as the largest and most diverse datasets for the task of semantic event extraction: *DBpedia-SEE* and *Wikidata-SEE*. They are available online.[11] *DBpedia-SEE* and *Wikidata-SEE* serve as training and test corpora for semantic event extraction based on event ontologies of DBpedia and Wikidata. To comply with the definition of semantic event extraction in Definition 3, each dataset belongs to an event ontology $G_O$ and contains a set of texts, where each text $t$ is annotated with a set of events $V_t$, their classes and relations.

#### 5.1.1. Event Ontology Extraction

In the first step, we extract relevant event classes and their properties from DBpedia and Wikidata to create two event ontologies. To ensure their quality and the relevance of their event classes and properties, we apply stringent filtering protocols. Specifically, we restrict event classes and properties to those used in the context of events and apply a minimum threshold for event classes (100 appearances) and properties (50 appearances). While we try to keep manual interventions minimal and to be as consistent as possible in our annotations, for the remaining events and properties, we need to manually filter out overly specific event classes and metadata properties. Specifically, for Wikidata, we filtered out the following three types of event classes and properties:

- Event classes specifically about a country (we still consider their parent classes. For example, instead of "UK Parliamentary by-election", there still is "by-election"). Examples are:
  * Turkish general election (`wd:Q22333900`)
  * Spanish Grand Prix (`wd:Q9208`)
  * Sydney International (`wd:Q248952`)
- Classes that are wrongly categorised as event classes in Wikidata. Examples are:
  * communications satellite (`wd:Q149918`)
  * space telescope (`wd:Q148578`)
  * crewed spacecraft (`wd:Q7217761`)
- Properties that do not represent real-world relations (e.g., identifiers). An example is:
  * X username (`wdt:P2002`)

Statistics of the resulting DBpedia and Wikidata event ontologies are shown in Table 2. For example, the Wikidata event ontology has 60 event classes and $5,901$ events typed as `sport season`. We consider the two event ontologies independently from each other and do not align them.

[11] https://zenodo.org/records/10818676

Table 2

Statistic of the extracted DBpedia and Wikidata event ontologies.

| | *DBpedia-SEE* | *Wikidata-SEE* |
|---|---|---|
| **Event Classes** | 19 | 60 |
| **most occurrences** | `dbo:MilitaryConflict` (23, 264) | `wd:Q27020041` (sports season) (5, 901) |
| **least occurrences** | `dbo:MixedMartialArtsEvent` (104) | `wd:Q1079023` (championship) (55) |
| **Properties** | 29 | 17 |
| **most occurrences** | `dbo:place` (17, 618) | `wdt:P585` (point in time) (31, 378) |
| **least occurrences** | `dbo:previousMission` (72) | `wdt:P571` (inception) (20) |

### 5.1.2. Extraction of Event Triples

To extract texts and the RDF triples representing mentioned events, we follow a distance-label generation process. The individual texts are sentences extracted from articles in the English Wikipedia describing events[12]. Event classes and relations are extracted by exploiting existing links to events and their DBpedia or Wikidata representations.

Fig. 6 illustrates the distance-label generation process at an example: The Wikipedia article "Turkish involvement in the Syrian civil war" has a link to the event "Operation Euphrates Shield" which has a relation to Syria and is also mentioned in the same text. Consequently, we select the text, the event class `military operation`[13], and the `country` relation to `Syria`.

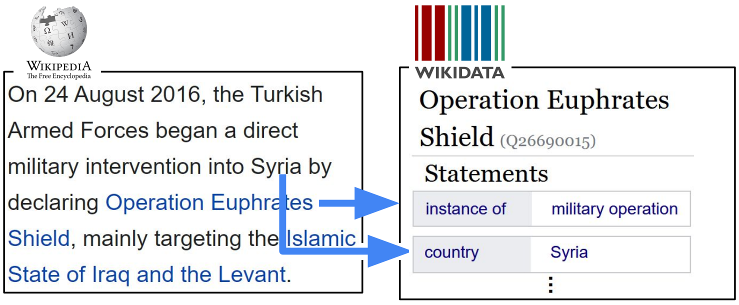


Fig. 6. Example illustrating how we label texts with events and relations. The Wikipedia text on the left links to the Wikidata event on the right side, which also has a relation to an entity mentioned in the text: <`country`, `Syria`>.

### 5.1.3. Statistics

As delineated in Table 3, *DBpedia-SEE* includes 42, 648 texts, and *Wikidata-SEE* contains 37, 988 texts, where each text contains at least one annotated event and its corresponding relations. Together, these datasets feature over 80, 636 uniquely annotated events and in excess of 111, 663 relation instances, making them the most extensive repositories for training and evaluating event extraction models to date.

Table 3

Statistic of our datasets for semantic event extraction.

| | *DBpedia-SEE* | *Wikidata-SEE* |
|---|---|---|
| **Texts** | 42, 648 | 37, 988 |
| **Events** | 42, 726 | 38, 014 |
| **Relations** | 47, 666 | 63, 997 |

[12]Event articles typically contain descriptions of related events.

[13]If an event has multiple event classes, we select the most infrequently used event class among them in order to add fine-grained event classes to the dataset.

#### 5.1.4. Comparison to Existing Datasets

*DBpedia-SEE* and *Wikidata-SEE* distinctly surpass existing benchmarks for the task of semantic event extraction due to their use of RDF annotations, their focus on general-domain events with societal impact and the coverage of both event detection and relation extraction annotations. Datasets such as SuicideED [40], SCIERC[41] and GENIA [42] only cover very domain-specific events. MAVEN [20] and MINION [43] only provide annotations for event detection, not relation or argument extraction. The existing larger event datasets like GDELT [44, 45] are less structured and not in RDF[14]. In a comparison to the *ACE05* [28] dataset typically used for event extraction, our datasets *DBpedia-SEE* and *Wikidata-SEE*:

- are freely available
  - *ACE05* is only available for $4,000.00 to non-members of the Linguistic Data Consortium.
- have wider coverage of event domains
  - e.g., *ACE05* does not have sport-related events
- use RDF classes and properties
- have a large number of event classes and properties
  - *DBpedia-SEE*: 19 event classes and 29 properties
  - *Wikidata-SEE*: 60 event classes and 17 properties
  - *ACE05*: 33 event classes and 22 arguments
- provide a large number of texts
  - *DBpedia-SEE*: 42,648 texts
  - *Wikidata-SEE*: 37,988 texts
  - *ACE05*: 599 texts

These attributes amplify the datasets' potential for semantic event extraction, which can not be performed with other existing datasets.

#### 5.1.5. Data Preparation and Experiment Design

With our distantly labelled datasets *DBpedia-SEE* and *Wikidata-SEE*, we are able to i) train `T-SEE` and the baselines on large-scale datasets and (ii) evaluate their performance in the semantic event extraction of events which already exist in DBpedia or Wikidata. We exclude links to existing events when running the experiments to simulate the situation in which the events do not yet exist in the target knowledge graph.

In our experiments, we split the datasets into training, test, and validation sets using 70:15:15 splits.

### 5.2. Evaluation Setup

Next, we describe our evaluation setup, i.e., baselines and metrics.

#### 5.2.1. Baselines

We compare `T-SEE` against two baselines:

- `Text2Event` [46]: A state-of-the-art method for event extraction using a sequence-to-structure generation paradigm.
- `EventGraph` [47]: A method for event extraction using semantic graph parsing that has shown state-of-the-art results for the task of argument role classification.

The selection of baselines for our study is carefully considered but constrained by the availability and adaptability of existing event extraction methodologies due to the following reasons: (i) Despite their valuable contributions, several works do not provide any accessible implementations [48–50], which is a critical barrier to replication and

[14]Instead of concise event mentions, GDELT considers whole articles as texts and does not provide relation types between events and entities.

Table 4

F1 scores for event classification on *DBpedia-SEE* and *Wikidata-SEE*.

| Approach | DBpedia | | | Wikidata | | |
|---|---|---|---|---|---|---|
| | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| `Text2Event` | **0.94** | **0.94** | **0.94** | 0.84 | 0.84 | 0.84 |
| `EventGraph` | 0.75 | 0.69 | 0.72 | 0.77 | 0.52 | 0.62 |
| `T-SEE` | 0.92 | 0.92 | 0.92 | **0.85** | **0.85** | **0.85** |
| `L-SEE*` | 0.88 | 0.89 | 0.89 | 0.53 | 0.58 | 0.55 |

further research. (ii) The usability of many event extraction frameworks is hampered by a lack of comprehensive documentation and a dependency on specific or proprietary datasets, notably the *ACE05* dataset [51–54]. Other methodologies like *DEGREE* [53] and the question-answering paradigms by [54] and [52] necessitate additional, task-specific inputs such as argument and description queries, complicating their integration into diverse research settings. Similarly, [51] and *ChatIE* [55] are hindered by very limited documentation and strict data formatting requirements. (iii) *CLEVE* [27] cannot be adapted to our definition of semantic event extraction due to its presupposition of argument type knowledge. (iv) Frameworks like *AllenNLP* (on which *DyGIE++* [56] is built) have been discontinued, and (v) the substantial computational resources required for models like the 10-billion parameter *Deepstruct* [57] model further limit their viability.

Given these considerations, we have chosen `Text2Event` and `EventGraph` as our baselines. These methodologies have demonstrated strong performance in the event extraction task (e.g., they both outperform *Deepstruct* event classification [46, 47, 57]), provide publicly available code, and are adaptable to our task definition.

#### 5.2.2. Metrics & Setting

To evaluate `T-SEE`'s and `L-SEE`'s performance on semantic event extraction, we assess their performances both on event classification and relation extraction.

We judge the accuracy of event classification using precision, recall, and $F_1$ scores to assess if events with correct classes were extracted. Analogously, we use the same metrics for evaluating relation extraction, where relations are only considered to be correct if connected to a correctly classified event via the correct property and to the correct entity or value.

In this section, we report the results of `L-SEE` as `L-SEE*` since we only consider those texts for which the output of the LLM was formatted syntactically correctly and in a consistent way allowing us to use automatic evaluation. Consequently, `L-SEE` is not evaluated on an identical dataset as `T-SEE` and the baselines, but on a smaller dataset ($5,602$ of the full $6,407$ texts for *DBpedia-SEE* and $3,958$ texts of the original $5,711$ for *Wikidata-SEE*).

### 5.3. Event Classification

Table 4 shows the evaluation results of `T-SEE`, `L-SEE` and the baselines on the tasks of event classification. In general, we observe that `T-SEE` performs well on event classification, reaching $F_1$ scores of 0.92 and 0.85. `T-SEE` and `Text2Event` outperform by a notable margin `EventGraph`. While `Text2Event` performs better than `T-SEE` on *DBpedia-SEE*, `T-SEE` performs better on the more diverse Wikidata dataset. This performance of `T-SEE` may be attributed to its capability of dealing with rich event ontologies, given that *Wikidata-SEE* has three times more event classes than *DBpedia-SEE*.

The performance of `L-SEE` closely follows that of the baselines on *DBpedia-SEE*. However, we do see a notable drop-off in the case of Wikidata, likely correlated with the larger number of fine-grained event classes notable to Wikidata.

### 5.4. Relation Extraction

Table 5 presents the relation extraction performance of `T-SEE` and our baselines. While it is evident that `EventGraph` lags behind `Text2Event` and `T-SEE`, it demonstrates a notable precision in its extractions (0.85

Table 5

Precision (P), recall (R) and $F_1$ scores for relation extraction on DBpedia and Wikidata.

| Approach | DBpedia | | | Wikidata | | |
|---|---|---|---|---|---|---|
| | **P** | **R** | **$F_1$** | **P** | **R** | **$F_1$** |
| `Text2Event` | 0.74 | 0.75 | 0.74 | **0.75** | **0.77** | **0.76** |
| `EventGraph` | 0.72 | 0.57 | 0.64 | 0.85 | 0.16 | 0.27 |
| `T-SEE` | **0.75** | **0.76** | **0.75** | **0.75** | **0.77** | **0.76** |
| `L-SEE*` | 0.28 | 0.52 | 0.37 | 0.37 | 0.37 | 0.37 |

in *Wikidata-SEE*), albeit with a significantly lower recall (0.16). This suggests that `EventGraph` is highly accurate in the instances it chooses to label, but it misses many relevant relations. In contrast, `T-SEE` consistently matches or outperforms all baseline performances across both datasets, demonstrating its robustness in the relation extraction task.

`L-SEE` shows a notably lower performance compared to both `T-SEE` and the baselines. This is likely associated with the relatively higher complexity of the relation extraction task compared to the event classification task. It should also be noted that as `L-SEE` is untrained, it relies more on the natural language understanding abilities it acquired through pre-training than other evaluated baselines. As such, it is also more likely to fail when the property labels are inadequately descriptive as to what purpose they are meant to fulfil. We go into further detail about the nature of these errors and the limitations of `L-SEE` in Section 6.

### 5.5. Cascading Errors

Given the sequential structure of `T-SEE`'s and `L-SEE`'s approach, where event classification precedes relation extraction, inaccuracies in the initial phase of event classification might negatively influence the subsequent relation extraction performance. Therefore, we analyse the impact of cascading errors by comparing the F1 scores for relation extraction in isolated and end-to-end settings at the example of `T-SEE`.

For `T-SEE` on *DBpedia-SEE*, the isolated setting shows a precision score of 0.81, recall of 0.82, and an 0.82 $F_1$ score, indicating the model's performance in an ideal scenario with perfect event classification. However, in the end-to-end setting, the scores decrease to the precision of 0.75, recall of 0.76, and an $F_1$ score of 0.75. This drop in performance suggests that errors in the event classification phase cascade down, as expected, affecting the model's ability to extract relations accurately.

Similarly, on *Wikidata-SEE*, `T-SEE` demonstrates high scores in the isolated setting with the precision of 0.87, recall of 0.88, and an $F_1$ score of 0.88. In contrast, the end-to-end setting yields lower scores: Precision 0.75, Recall 0.77, and $F_1$ 0.76. This reduction further underscores the presence of cascading errors. However, the effect is limited and does not prevent `T-SEE` from outperforming or matching the baselines in both datasets.

### 5.6. Example Result

Finally, we provide example RDF triples of an event extracted with `T-SEE`. Fig. 7 shows the RDF triples created from an event which we extracted from the Wikipedia article "1991 Monte Carlo Open"[15] using `T-SEE`. As we can see, `T-SEE` successfully extracted an event of the fine-grained event class `recurring tennis tournament` and several relations, including properties such as `season starts`, `located in the administrative territorial entity` and `part of`.

### 5.7. Implementation

In order to implement our multilabel classification model, we leverage a pre-trained uncased BERT base model[16]. The model is fine-tuned for 30 epochs using the focal loss function with a gamma of 2, and the Adam optimiser,

[15] https://en.wikipedia.org/w/index.php?title=1991_Monte_Carlo_Open&oldid=1101607800

[16] https://huggingface.co/bert-base-uncased

```
@base <http://example.org/>.
@prefix wd: <https://www.wikidata.org/entity/>.
@prefix wdt: <https://www.wikidata.org/prop/direct/>.
@prefix so: <https://www.w3.org/2000/01/rdf-schema#>.

:E1 a wd:Q47443726 (recurring tennis tournament);
   so:description "It was ... part of the ATP Super 9 of the 1991 ATP Tour. It took place at
       the Monte Carlo Country Club in Roquebrune-Cap-Martin, France, near Monte Carlo, Monaco,
        from 22 April through 28 April 1991."@en ;
   wdt:P4794 (season starts) wd:Q118 (April) ;
   wdt:P17 (country) wd:Q142 (France) ;
   wdt:P131 (located in the administrative territorial entity)
      wd:Q45240 (Monte Carlo) ;
   wdt:P276 (location) wd:Q3861317 (Monte Carlo Country Club) ;
   wdt:P361 (part of) wd:Q300008 (ATP Tour) .
```

Fig. 7. Example of RDF triples generated from the Wikipedia article "1991 Monte Carlo Open" using the Turtle syntax.

with a learning rate of $1e-5$ and a Dropout layer with a probability of $0.3$. For the relation extraction model, we utilise the same BERT model and fine-tune it on the relation extraction task. Similarly to the classification model, we train the model for 30 epochs with the Adam optimiser and a learning rate of $3e-5$.

`Text2Event` and `EventGraph` are trained for $40$ epochs using a batch size of 30 and their original training settings.

To generate the training data, we extract Wikipedia articles using the MWDumper[17]. For entity linking, we use the Spacy Entity Linker[18], a named entity linking tool specifically designed for Wikidata.

For `L-SEE`, we use `gpt-3.5-turbo-1106`, a version of GPT-3.5 Turbo that supports a 16K context and supports improved instruction following, JSON mode, and parallel function calling. We pick this version as it has shown a $38\%$ improvement in format following tasks such as generating JSON, XML and YAML.[19]

## 6. Comparison of `T-SEE` and `L-SEE`

A significant finding of our evaluation is the worse performance of `L-SEE` compared to `T-SEE` on the task of relation extraction (Section 5.4). This leads to the question of whether LLMs are not suited for the task of semantic event extraction at all, in contrast to fine-tuning a transformer-based architecture. To answer this question, this section delves into a manual evaluation and a multifaceted error analysis, followed by a discussion.

### 6.1. Manual Evaluation

In this section, we aim to understand the differences between the two paradigms of transformer-based architecture versus using LLMs for semantic event extraction. Therefore, on top of the automatic evaluation performed in Section 5, we perform a comparison of `T-SEE` and `L-SEE` based on a manually annotated subset of the test dataset used in the automatic evaluation.

We create *DBpedia-SEE*$_{100}$ – a subset of *DBpedia-SEE* with 100 randomly selected texts, their events and relations. We ensure that `L-SEE` successfully performs semantic event extraction on these texts without syntactical errors. For each text in *DBpedia-SEE*$_{100}$, we manually annotate the semantic event representations generated by `T-SEE` and by `L-SEE` with respect to each other and the ground truth. For example, given a text *t*, if `T-SEE` generates a relation *r* that is not in *DBpedia-SEE*$_{100}$, we manually assess whether *r* is correct and expressed in *t*. If this

[17] https://www.mediawiki.org/wiki/Manual:MWDumper
[18] https://github.com/egerber/spaCy-entity-linker
[19] https://openai.com/blog/new-models-and-developer-products-announced-at-devday

assessment is positive and *r* is also missing in `L-SEE`, we denote a true positive for `T-SEE` and a false negative for `L-SEE`.

Table 6 shows the results of evaluating `T-SEE` and `L-SEE` on *DBpedia-SEE*$_{100}$ before and after our manual assessment. The results before manual assessment confirm our results given in Table 5, where `T-SEE` and `L-SEE` both perform well on event classification ($F_1$ scores of 0.92 and 0.89), but `L-SEE` is clearly outperformed by `T-SEE` for relation extraction ($F_1$ scores of 0.72 and 0.39), mainly due to 200 false positive extracted relations. This indicates a considerably better ability of `T-SEE` to accurately identify and categorise relationships within the data under controlled conditions.

Table 6

Evaluation of `T-SEE` vs `L-SEE` on *DBpedia-SEE*$_{100}$ before and after manual assessment (TP: true positives, FP: false positives, FN: false negatives).

| | **TP** | **FP** | **FN** | **$F_1$** |
|---|---|---|---|---|
| | | **Event Classification** | | |
| `T-SEE` | 92 → 90 | 7 → 12 | 10 → 9 | **0.92** → 0.90 |
| `L-SEE` | 91 → 100 | 11 → 2 | 11 → 2 | 0.89 → **0.98** |
| | | **Relation Extraction** | | |
| `T-SEE` | 83 → 103 | 33 → 23 | 32 → 114 | **0.72** → 0.58 |
| `L-SEE` | 77 → 178 | 200 → 99 | 38 → 29 | 0.39 → **0.74** |

After manual assessment, `L-SEE` shows a remarkable improvement in event classification, achieving an almost perfect $F_1$ score of 0.98, suggesting that with manual verification of the ground truth, the LLM's capabilities are more effectively utilised. Regarding relation extraction, while `L-SEE` improves performance ($F_1$ score of 0.74), `T-SEE` experiences a significant drop in effectiveness ($F_1$ score of 0.58), indicating challenges in adapting to the intricacies of manually annotated samples and the complexity of real-world data.

These results underscore the strengths and limitations of both methodologies. While `T-SEE` demonstrates superior performance in a controlled environment, particularly in relation extraction tasks, `L-SEE` shows remarkable adaptability and potential in handling complex, real-world scenarios when supplemented with manual verification and annotation processes: not being constrained by any limitations in training data, `L-SEE` is able to extract more than double the amount of relations. This highlights the importance of context and the level of detail in ground truth annotations when evaluating and comparing data extraction methodologies.

### 6.2. Error Taxonomy

To understand the differences in behaviours between `T-SEE` and `L-SEE`, we manually annotate the specific errors that occur when performing semantic event extraction on *DBpedia-SEE*$_{100}$. While doing so, we create an error taxonomy presented in this section. Later, to contextualise said error taxonomy, we present examples of generated RDF triples and the errors in them.

Our manual annotation process has unveiled a structured classification of errors, which we have divided into three principal categories:

*Extraction Inaccuracies* Errors arising from the model's inability to accurately interpret information within texts:

- *Omissions or Missing Events/Relations*: The event or its relations are not extracted.
- *Type misalignment*: An inappropriate type of entity or value is selected for a given property.
- *Granularity mismatch*: The model's predictions lack the specificity of the ground truth, e.g., categorising an event broadly as `dbo:SportsEvent` rather than the more specific `dbo:TennisTournament`.
- *Erroneous extraction*: The extraction of incorrect properties or values, leading to a misrepresentation of the factual content.

*Annotation Discrepancies* Errors stemming from inconsistencies, errors or omissions in the ground truth:

- *Imprecise event class*: The model's predictions provide a more detailed event classification.
- *Imprecise property*: The model predicts property values with greater accuracy than the ground truth, such as specifying the exact match score when the ground truth only acknowledges the victory.
- *Annotation error*: The presence of omissions or inaccuracies within the ground truth itself, such as neglecting to annotate the specific date of a match or other pertinent details.

*Other Anomalies* Errors arising from other sources:

- *Event ambiguity*: The model struggles to distinguish between multiple distinct events described within a single sample, which may lead to conflated or mixed property assignments.
- *Processing error*: `T-SEE` and `L-SEE` match spans of text to specific entities, relying on an external entity linking component and a date parsing module which are prone to errors.

#### 6.2.1. Examples of Errors

We provide four semantic event representations generated by `L-SEE` as examples of the identified error types in the error taxonomy. For each of the examples, we provide the input text *t*, selected RDF triples describing an event *e* in the ground truth as well as selected triples generated by `L-SEE`.[20]. Errors are marked in red, relations only in the ground truth are marked in blue, and relations only in the prediction are marked in green.

***Example 1*** *(Fig. 8) – ground truth extracted from `dbr:Black_Monday_(1360)`:*

- Omission: `L-SEE` failed to extract the `dbo:commander` relation.
- Annotation error: On the other hand, `L-SEE` accurately extracts a relevant date and territory for the event, however, these are not contained within the ground truth.

**Text**: This was in part caused by Black Monday (1360), the freak storm that devastated the English army and forced Edward III into peace talks.

**Ground Truth**

```
:MilitaryConflict1 dbo:commander dbr:Edward_III_of_England .
```

**Prediction**

```
:MilitaryConflict1 dbo:date "1360-01-01"^^xsd:date ;
                   dbo:territory dbr:England .
```

Fig. 8. Example of an omission error and an annotation error.

***Example 2*** *(Fig. 9) – ground truth extracted from `dbr:Al-Qusayr_offensive`:*

- Type misalignment: The commanders are incorrectly identified and assigned to group entities instead of individuals. Specifically, `L-SEE` detects two commanders extracting "Syrian Army" and the "Lebanese militia Hezbollah".
- Processing error: in the entity linking process, "Lebanese militia Hezbollah" is wrongly linked to three entities.

[20] For brevity, we skip $p_{type}$ and $p_{description}$ relations. The event class is indicated by its URL (e.g., `:MilitaryConflict1` is an event classified as `dbo:MilitaryConflict`).

**Text**: The second of two battles in al-Qusayr started on 19 May 2013, as part of the larger al-Qusayr offensive, launched in early April 2013 by the Syrian Army and the Lebanese militia Hezbollah, during the Syrian civil war, with the aim of capturing the villages around the rebel-held town of al-Qusayr and ultimately launching an attack on the town itself.

**Ground Truth**

```
:MilitaryConflict2 dbo:place dbr:Al-Qusayr,_Syria ;
                   dbo:isPartOfMilitaryConflict dbr:Syrian_Civil_War .
```

**Prediction**

```
:MilitaryConflict2 dbo:place dbr:Al-Qusayr,_Syria ;
                   dbo:commander dbr:Syrian_Army ;
                   dbo:commander dbr:Lebanon ;
                   dbo:commander dbr:Militia ;
                   dbo:commander dbr:Hezbollah ;
                   dbo:isPartOfMilitaryConflict dbr:Syrian_Civil_War .
```

Fig. 9. Example of type misalignment and processing errors.

***Example 3*** *(Fig. 10) – ground truth extracted from* `dbr:2016_Wuhan_Open`*:*

- Imprecise event class: `L-SEE` identifies a more precise event class (`dbo:TennisTournament` versus `dbo:Tournament`).
- Erroneous extraction and event ambiguity: The same tennis tournament did not happen in Wuhan and in Beijing; `L-SEE` fails to distinguish between the tournaments Wuhan Open and China Open.

**Text**: However, she rebounded in the Asian swing by reaching the quarterfinals of Wuhan and the semifinals of Beijing.

**Ground Truth**

```
:Tournament1 dbo:location dbr:Wuhan .
```

**Prediction**

```
:TennisTournament1 dbo:location dbr:Wuhan ;
                   dbo:location dbr:Beijing .
```

Fig. 10. Example of an imprecise event class, an erroneous extraction and event ambiguity.

***Example 4*** *(Fig. 11) – ground truth extracted from* `dbr:1959_Ontario_general_election`*:*

- Event ambiguity: The date "1961-01-01" indicates confusion between multiple events. Specifically, this is because the event annotated in the ground truth is derived from the link tied to the string "previous election", referring to `dbr:1959_Ontario_general_election`.
- Erroneous extraction: The use of `dbo:secondLeader` to indicate a chronological successor is highlighted in red, illustrating a misunderstanding of the property, as `dbo:secondLeader` is meant to instead describe second ranking in a competition.
- Annotation error: The relation using the `dbo:affiliation` property is missing in the ground truth.

**Text**: The Ontario Progressive Conservative Party, led by John Robarts, who had replaced Leslie Frost as PC leader and premier in 1961, won a seventh consecutive term in office, and maintained its majority in the legislature, increasing its caucus from the 71 members elected in the previous election to 77 members in an enlarged legislature.

**Ground Truth**

```
:Election1 dbo:country dbr:Ontario ;
           dbo:firstLeader dbr:Leslie_Frost .
```

**Prediction**

```
:Election1 dbo:startDate "1961-01-01"^^xsd:date ;
           dbo:country dbr:Ontario ;
           dbo:secondLeader dbr:John_Robarts ;
           dbo:affiliation dbr:Progressive_Conserv._Party_of_Canada ;
           dbo:firstLeader dbr:Leslie_Frost .
```

Fig. 11. Example of event ambiguity, erroneous extraction and an annotation error

### 6.3. Error Analysis

On the basis of our error taxonomy, we annotated each semantic event representation generated by `T-SEE` and `L-SEE` with the set of errors occurring in them.

First, we categorise errors into extraction inaccuracies, annotation discrepancies, and other anomalies to clarify our approaches' error landscapes. Fig. 12 visualises these error profiles for `T-SEE` and `L-SEE`, highlighting the challenges in semantic event extraction. In general, we register fewer errors for `T-SEE` than `L-SEE` across all three error categories, which results from `T-SEE`'s capability to mimic the dataset characteristics. On the other hand, we annotate 180 annotation discrepancies for `L-SEE`, more than its 107 extraction inaccuracies. Since annotation discrepancies represent cases where the model extracts valid triples which are not covered in the ground truth, this analysis demonstrates how `L-SEE` is capable of semantic event extraction without being closely attached to the characteristics of training data and, implicitly, the data coverage in the target knowledge graph.

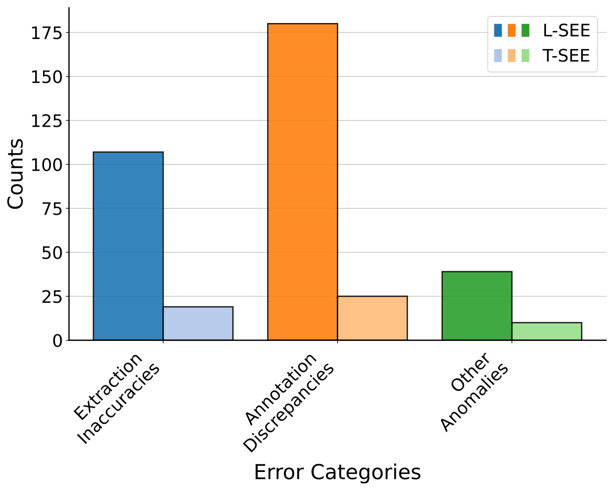


Fig. 12. Distribution of error categories for `T-SEE` and `L-SEE`.

Fig. 13 provides a detailed analysis of the error types. As can be seen in the figure, different error types manifest with varying frequencies across `L-SEE` and `T-SEE`.

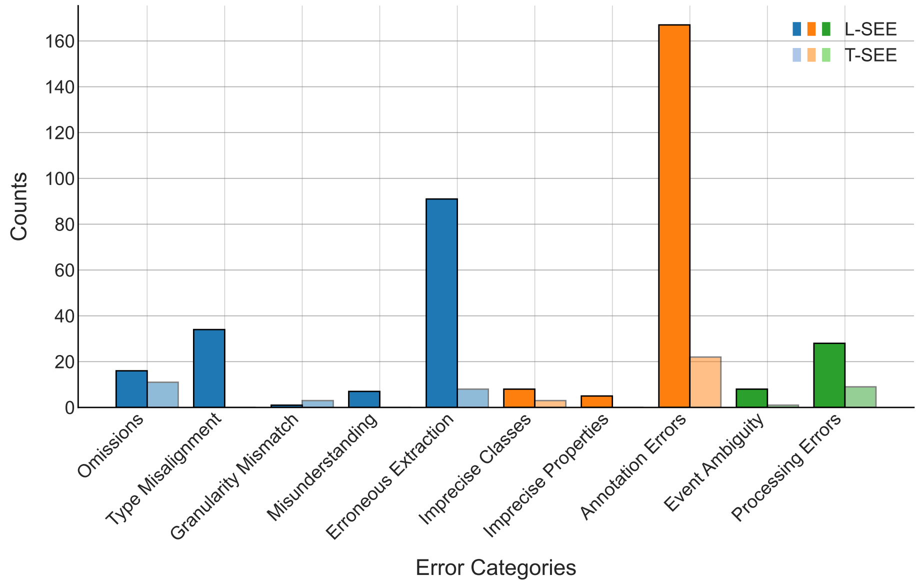


Fig. 13. Distribution of error types for `T-SEE` and `L-SEE` grouped by category.

For `L-SEE`, the most prevalent error type are *Annotation Errors*, with a count of 167, reflecting instances where `L-SEE` identifies relevant semantic relations not annotated in the ground truth. Following closely is the *Erroneous Extraction* category, with 91 instances, which encompasses errors related to the incorrect identification of properties or values. A notable portion of these errors can be attributed to *Processing Errors*, amounting to 28 instances, where the entity linking and date extraction methods utilised by `L-SEE` falter in accurately extracting dates or correctly linking entities, leading to inaccuracies in the relation extraction.

*Misunderstanding* and *Type Misalignment* errors, with 7 and 34 instances respectively, further contribute to the *Erroneous Extraction* count. These errors emerge when `L-SEE` misinterprets the intended meaning of properties or incorrectly aligns relations with inappropriate entities or values. For instance, the common misunderstanding of the `dbo:secondLeader` property (Example 4) exemplifies how more careful prompting approaches may lead to better performance. For an example of a type misalignment error, we may observe instances where in the absence of precise time expressions in the the text, `L-SEE` assigns imprecise values such as "yesterday" to date relations.

Conversely, `T-SEE` demonstrates a lower overall error frequency, with *Annotation Errors* again emerging as the dominant error type, albeit with a substantially lower count of 22. This suggests a more precise alignment with the ontology. Notably, `T-SEE` exhibits no *Type Misalignment* errors and only 1 *Event Ambiguity* error. However, both methodologies encounter *Omissions* and *Processing Errors*, with `L-SEE` facing 16 and 28 instances respectively, and `T-SEE` experiencing 11 and 9 instances.

Upon a more nuanced examination, especially after correcting for annotation errors, the performance landscape shifts. Initially, `T-SEE` appears to outperform `L-SEE` due to its lower error rates. However, this might also indicate a tendency of `T-SEE` to conform to the existing annotations, potentially overlooking unlabelled but present relations. This could imply that while `T-SEE` is more aligned with the given annotations, it may also be less inclined to explore beyond them, possibly fitting to annotation noise rather than capturing the full spectrum of semantic relations.

In summary, while `T-SEE` shows precision in alignment with the current ontology, `L-SEE`'s broader extraction attempts, despite higher initial error rates, may offer a more comprehensive understanding of the underlying semantic structures, especially when considering the corrected annotation context. This dichotomy highlights the balance between precision and recall in semantic event extraction and underscores the importance of continuous refinement in both methodologies to enhance their efficacy and reliability.

#### 6.3.1. Formatting and Ontology Errors

As indicated in Section 5.2.2, for 805 of the texts in the complete dataset *DBpedia-SEE*, `L-SEE` could not generate RDF triples due to formatting issues, including:

- Misformatted output: The LLM-generated JSON strings of 606 texts were not in proper JSON syntax and could not be processed.
- Non-existing event classes: In 401 cases, an event class was identified which is not part of the event ontology and the prompt. An example is the extraction of an event typed as `dbo:SyrianCivilWar`, while only `dbo:CivilWar` exists in the DBpedia ontology.
- Invalid properties: In 1,191 cases, a property was identified which is not part of the event ontology (e.g., `dbo:percentageOfPopularVote`, `dbo:delayReason`). Despite these being errors, they often demonstrate `L-SEE` capability to suggest relevant attributes for specific scenarios adaptively.

### 6.4. Effect of Text Characteristics on Semantic Event Extraction

To get a sense of `L-SEE` performance across a variety of syntactic and semantic phenomena, we dissected *DBpedia-SEE* into multiple subsets, each representing distinct text characteristics. The subsets are generated employing specific strategies, each tailored to highlight a particular aspect of the dataset, ranging from event co-occurrences to the complexity of the document structure.

#### 6.4.1. Text Characteristics

We employ a collection of strategies to generate meaningful subsets of the dataset, each aimed at isolating different factors that could influence `L-SEE`'s performance. Specifically, we ranked the entire dataset based on the presence and frequency of certain linguistic, syntactic, or semantic phenomena. From this ranking, we then selected the top 100 samples for each subset to focus our analysis on the most pronounced examples of each phenomenon.

**Semantic Diversity:** We assess samples for semantic diversity. The semantic diversity of a text is measured by the variety of verb phrases and their arguments, approximated by the count of unique verb lemmas in the text. Samples with high semantic diversity are chosen for this subset, aiming to test the model's understanding of varied semantic contexts and its ability to extract a broad range of event semantics.

**Sentence Length:** This strategy sorts the samples by the length of the text. Samples are then selected from the sorted list, prioritising those with the longest texts.

**Geographical Diversity:** Samples of this subset are generated based on the count of geographical entities identified by the *Spacy* NLP pipeline (i.e. "GPE" and "LOC" labelled entities) in each text. To assess the model's proficiency in dealing with texts containing diverse geographical references, we select samples with the highest counts of such entities.

**Temporal Event Distribution:** We identify texts with temporal expressions using the *Spacy* library and extract where they are most frequently occurring. As temporal expressions can be crucial for event understanding, this subset evaluates `L-SEE`'s capability to understand and integrate temporal information.

**Named Entity Diversity:** This subset focuses on the diversity of named entities. We again utilise the *Spacy* library to extract named entities and then sort and select samples with the widest range of entities. This subset tests the model's ability to accurately recognise and categorise entities in the context of events.

**Complex Sentence Structures:** Samples with intricate syntactical constructions are selected to challenge `L-SEE`'s parsing abilities, as complex structures can obscure event boundaries and relations, making extraction more difficult. This set is generated by measuring the depth of the syntactic parse tree of each text, with depth representing the maximum distance from any token to the root of the tree. Samples with the most complex sentence structures, i.e., the deepest parse trees, are selected for this subset.

In the following, we detail the outcomes of this analysis, demonstrating `L-SEE`'s efficacy and limitations across varying text characteristics.

#### 6.4.2. *Results*

L−SEE's performance was evaluated across the subsets using precision, recall, and $F_1$ scores for both event classification and relation extraction. Figures 14 and 15 show the results of this analysis, detailed in the following.

For comparison, we also include the full dataset performance in our analysis. The distinctly strongest relation extraction performance on the full dataset suggests that we have successfully sampled parts of our data that L−SEE finds difficult to deal with.

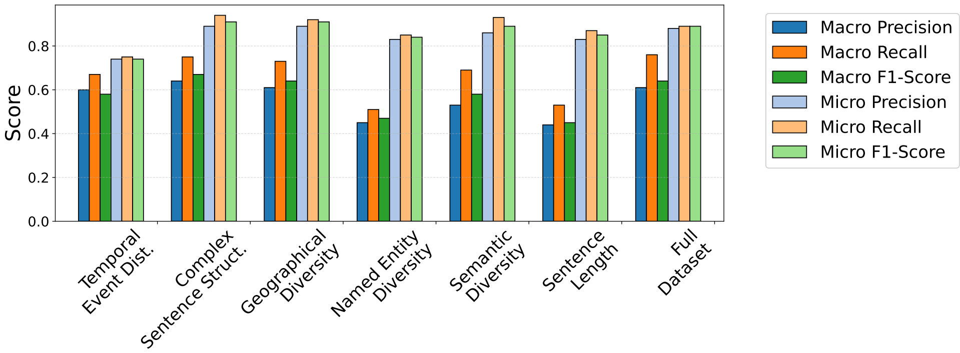


Fig. 14. L−SEE performance in event classification across various data subsets.

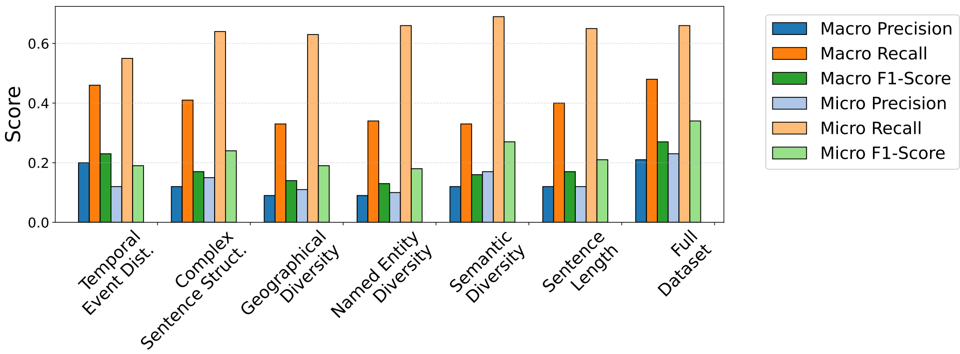


Fig. 15. L−SEE performance in relation extraction across various data subsets.

**Semantic Diversity** L−SEE exhibits moderate performance in the semantic diversity subset, with macro metrics reflecting challenges in consistently classifying a wide range of semantically varied events. However, micro metrics indicate better performance in frequent semantic contexts. The significantly lower performance in relation extraction highlights that there may be difficulties in mapping complex semantic relationships accurately.

**Sentence Length** The results suggest that longer sentences pose significant challenges, with lower macro metrics for event classification and even more pronounced difficulties in relation extraction. This indicates that L−SEE may have limitations in maintaining context and coherence over sentences.

**Geographical Diversity** L−SEE performs relatively well in event classification, suggesting a good grasp of geographical contexts. However, the lower relation extraction scores point to challenges in accurately extracting relationships when the diversity of geographical entities is high.

**Temporal Event Distribution** L−SEE displays reduced event classification accuracy in this subset. However, relation extraction metrics remain comparatively stable. Given the lower event classification performance, this stability in relation extraction, despite potential cascading errors from misclassifications, may indicate a relatively stronger inherent capability of the model in isolating and extracting relations in the context of temporally complex texts.

**Named Entity Diversity** We observe notable difficulties, highlighting `L-SEE`'s struggle with diverse named entities. The discrepancy between macro and micro metrics points to `L-SEE`'s poorer handling of less frequent event types when the diversity of entity mentions is high.

**Complex Sentence Structures** `L-SEE`'s strongest performance is observed in the subset with complex sentence structures, indicating effective parsing of intricate syntactic constructions. However, the lower relation extraction metrics suggest that while `L-SEE` can identify events within complex sentences, accurately extracting the relations remains challenging.

This analysis underscores `L-SEE`'s strengths in contextual integration and syntactic navigation but also points at significant areas for improvement. Future work should conduct further analysis focusing on how improved prompting strategies may help in robustness and accuracy across these diverse linguistic and contextual scenarios.

### *6.5. Discussion*

With our comparison of `T-SEE` and `L-SEE` in this article, we aim at a deeper understanding of the suitability of two different paradigms – transformer-based architectures and the use of LLMs – for semantic event extraction. Following our evaluation results and analysis, we identify five core phenomena to be considered when deciding between these paradigms:

**Mimicry of dataset characteristics:** Our analyses, e.g., in Table 5, Table 6 and Fig. 12, clearly demonstrate that the results of methodologies fine-tuned on the target datasets (`T-SEE`, `Text2Event` and `EventGraph`) are much more aligned to the expected RDF triples in the test sets than triples generated by an LLM (`L-SEE`). From this behaviour, we infer the following: (i) In a controlled setting where mimicking the characteristics of the training data is desired, transformer-based approaches are preferable over the use of LLMs. (ii) However, transformer-based approaches also mimic the flaws of the target datasets and knowledge graphs. For example, if a specific property is rarely used in an event knowledge graph but still valuable, `L-SEE` would identify it, while fine-tuned approaches might miss it. An example is the property `dbo:country` on the event class `dbo:Election`, which is only used in approximately $25\%$ of DBpedia's `dbo:Election` events.

**Distantly-labelled datasets:** Training a transformer-based architecture requires the availability of large training data, i.e., texts annotated with RDF triples. Therefore, we opted for the automated extraction of two new datasets. The use of distantly-labelled datasets without human annotations such as *DBpedia-SEE* and *Wikidata-SEE* for semantic event extraction or datasets for relation extraction [58–61] overcomes the issues of training data dimensionality but always comes with questions regarding dataset quality.[21] Specifically, we identified a large number of false positives when evaluating `L-SEE` (Table 6), resulting from incomplete knowledge graphs or faulty alignment between texts and RDF triples in the distant labelling process. Consequently, the evaluation of different approaches on a distantly-labelled dataset requires careful investigation of the outputs beyond solely providing scores of the evaluation metrics.

**Ontology Guidance:** We took care of carefully guiding both our approaches through our event ontologies. By fine-tuning a transformer-based architecture, adherence to the ontology can be enforced, e.g., by explicitly classifying into the event classes pertinent to the event ontology. For an LLM, in contrast, while we prompted for the specific event classes and properties, we still observed cases of invalid event classes or properties as discussed in Section 6.3.1. Also, our examples demonstrated cases of type misalignment and a misunderstanding of the semantic definition of a property (`dbo:secondLeader` in Example 4), demonstrating the need to control the outputs of an LLM. The improvement in the precision of LLM-based semantic event extraction is a major future direction for LLM-based semantic event extraction, e.g., through the provision of property descriptions within the prompt.

[21] Note that even human annotators frequently disagree when providing annotations for NLP tasks [62].

**Complexity:** Setting up a transformer-based architecture and its fine-tuning requires the availability of rich training data, computing and time resources. Setting up an LLM, in contrast, requires access to an LLM and careful prompt engineering, i.e., potentially easier-to-obtain resources.

**Real-world applications:** Given the capability of LLMs to adapt to different inputs and data characteristics, we assume that LLM-based approaches are well-suited under more complex, real-world conditions and to explore low-resource scenarios.

## 7. Related Work

Knowledge graphs have, as a form of structured human knowledge, drawn a lot of research attention from both academia and the industry [63]. With a great deal of event information worldwide, it is essential to bring entities and events together through event-centric knowledge representations [24], with event extraction and relation extraction being key technologies for accessing event knowledge [14].

### *7.1. Event Knowledge Graphs*

Event knowledge graphs represent knowledge about happenings with societal impact in an event ontology and interlink them with connected entities [24]. We distinguish between two types of event representations as follows:

- Named events: The predominantly entity-centric information of popular cross-domain knowledge graphs such as DBpedia, YAGO, and Wikidata represent events as *named events* such as “Brexit” and “World War II”. Named events are also the core component of EventKG [64], a multilingual event-centric temporal knowledge graph, part of the Open Event Knowledge Graph [65] that integrates event-related data sets from multiple application domains. GDELT [44] and ICEWS are two datasets of global political events encoded using the CAMEO framework [66], i.e., not in RDF.
- Unnamed events: Works that address *unnamed events* specifically deal with the identification of texts describing events and with the semantic annotation of these texts. For example, Rospocher et al. [22] build knowledge graphs from news articles, and Zhang et al. [67] develop a large-scale English event knowledge graph extracted from several sources such as reviews, news, and social media. For the task of event modelling, [68] proposes a weakly-supervised approach to extract event relation tuples from text and build an event knowledge base, not focusing on event-entity relations.

All event knowledge graphs require the availability of an event ontology, with popular examples including *LODE* [6], the *Simple Event Model* [7] and more as discussed by Pyriani et al. [69]. Relevant patterns for event representation are presented in [70, 71], focusing on the spatio-temporal extent of events, the role of their participants and recurring events. In this article, we extracted event ontologies from their vocabularies to allow the population of the well-established cross-domain knowledge graphs DBpedia and Wikidata.

With `T-SEE`, we aim to bring together the complementary strengths of the Semantic Web and NLP perspectives by performing event extraction that can be adapted to different event ontologies.

### *7.2. Event Extraction*

Event extraction (EE) is a critical task in constructing and populating entity-centric knowledge graphs, with recent advancements significantly diversifying the methodologies employed [30, 46, 72]. Traditional approaches have relied on sentence-level pipelines for extracting event triggers and their corresponding argument roles [52, 73, 74], employing sequence-to-structure generation paradigms like *Text2Event* [46] and multi-task frameworks such as *DyGIE++* [56], which utilise contextualised embeddings and dynamic span graph updates. Other studies have extended the scope to document-level EE [75, 76] or ventured into open-domain EE without predefined event classes [77, 78], which, while broadening the applicability, faces challenges due to the absence of a well-defined event ontology.

Innovations in the field have introduced contrastive pre-training frameworks like *CLEVE* [27], which capitalise on large unsupervised datasets and their semantic structures to enhance EE's efficacy, demonstrating marked improvements in both supervised and unsupervised settings. Similarly, *EventGraph* [47] has presented a joint framework that conceptualises events as graphs, facilitating the simultaneous detection and extraction of multiple events and their intricate interrelations, thereby achieving state-of-the-art results in event trigger and argument role classification.

*Deepstruct*, on the other hand, tries to leverage the structural understanding capabilities of language models through task-agnostic pretraining, allowing for zero-shot knowledge transfer across a wide array of structure prediction tasks and setting new benchmarks on numerous datasets [57]. With *DEGREE* [53], authors propose a data-efficient, generation-based model for EE that capitalises on semantic guidance from manually designed prompts and the joint prediction of triggers and arguments, showcasing robust performance in low-resource settings.

A notable shift in EE methodology is the adoption of a question-answering paradigm [52], which mitigates the prevalent issue of error propagation seen in conventional approaches by facilitating end-to-end argument extraction, including for roles not encountered during training. Following this line, *QGA-EE* [54] has refined the QA-based approach by integrating context-aware question generation, thus accommodating multiple arguments for identical roles and surpassing prior single-task models in performance metrics.

In light of the new methodologies and progress in event extraction, the research community has also focused on the specific subtasks of event extraction. For example, with *PAIE* [79], authors devise a prompt tuning approach to document-level event argument extraction similar to the already established question-answering paradigm in event extraction work. Older work on event argument extraction, such as *HMEAE* [18], a hierarchical approach to argument extraction utilising concept correlation among argument roles, have, in turn, inspired approaches such as *DEGREE* that aim to resolve issues such as poor handling of the encoding of the labels semantics and other weak supervision signals. The other subtask of event extraction, event detection, has also received attention with the *DRC* framework [80] trying to compete with trigger-based models as a way of exploring methods of event detection robust to less annotated real-world domains, an area we examine in our work as well.

These developments reflect a broader trend towards more adaptable, efficient, and comprehensive models for event extraction, underlining the field's evolution towards leveraging advanced language model capabilities and innovative problem-solving frameworks.

### *7.3. LLM-based Information Extraction*

The field of Information Extraction (IE) has traditionally relied on rule-based and statistical methods to extract structured information from text. However, the emergence of Large Language Models (LLMs) has opened up new avenues for tackling IE tasks with remarkable capabilities in understanding and generating natural language. This section reviews recent advancements in using LLMs for IE, particularly focusing on unstructured information extraction and event extraction.

*General Information Extraction with LLMs*

A few years ago, LLMs were still in their early stages of development, with limited capabilities for tackling complex tasks like information extraction. While early works explored LLM-based approaches for IE (e.g., [81]), these models faced challenges due to limited model capacity, data inefficiency, and limited adaptation. However, significant advancements in recent years have addressed these challenges, driven by the rise of the transformer architecture [33] enabling long-range dependencies. Large-scale pre-training pushed things further with BERT [35] and GPT-3 [82], allowing LLMs to learn general language understanding capabilities and adapt to specific IE tasks through fine-tuning with smaller labelled datasets. Finally, the growing availability of powerful computing resources like GPUs and TPUs [83] has enabled the training of larger and more complex LLM models, further enhancing their ability to handle complex information extraction tasks.

*Unstructured Information Extraction with LLMs*

In 2022, Dunn et al. showed how a pre-trained LLM can extract structured information from scientific abstracts [32]. In 2023, Polak et al. [84] expanded on the early promises of unstructured information extraction with *ChatExtract*, demonstrating that a significant amount of up-front effort, expertise, and coding may be fully automated using an advanced conversational LLM. By leveraging prompts and follow-up questions, *ChatExtract* achieves high accuracy and efficiency in extracting materials data, showcasing the potential of LLMs for automated knowledge extraction from scientific literature.

In the same year, Wei et al. proposed *ChatIE*, a multi-turn QA framework for zero-shot information extraction demonstrating good performance across a number of datasets, three tasks, and two languages [55]. Li et al. systematically analysed *ChatGPT* across seven detailed information extraction tasks [85] including event extraction. The authors show that while *ChatGPT* underperforms in standard IE tasks compared to BERT-based models, it excels in OpenIE settings, as confirmed by human evaluators. However, a notable concern is the model's overconfidence in its predictions, leading to calibration issues. This is further confirmed in the comprehensive survey by Liu et al. [86], in which the authors evaluate the capabilities and applications of *ChatGPT* (versions 3.5 and 4) against the backdrop of current state-of-the-art models in natural language processing. The paper highlights *ChatGPT*'s advancements in large-scale pre-training, instruction fine-tuning, and reinforcement learning from human feedback, which collectively enhance its adaptability and performance across a myriad of NLP tasks.A detailed comparison of *ChatGPT* with existing state-of-the-art models reveals that while *ChatGPT* excels in multitask learning and shows promising results in some NLP tasks, it falls short in multilingual capabilities and specialised tasks when compared to dedicated models. Moreover, stability and consistency emerge as areas where *ChatGPT* does not yet match the performance levels of state-of-the-art models, which could impact its reliability in critical applications.

#### *7.3.1. LLM-based Event Extraction*

LLMs have recently been utilised for the task of event extraction. In general, as already mentioned, Li et al. [85] evaluate the performance of *ChatGPT* on a number of information extraction tasks, revealing an increasingly worse performance as the complexity of the evaluated task increases, where the worst performance is reported on the task of event extraction.

A comparison between LLMs and traditional methods have been conducted on several tasks related to EE: In [87], authors explore prompt-based learning with *GPT-4* for detecting factual events in literary narratives. The study concludes that while BiLSTM with BERT embeddings excels in event detection within literary texts, *GPT-4* shows promise in prompt-based learning approaches, particularly in few-shot settings. Sharif et al. [88] conducted an in-depth analysis of *ChatGPT*'s performance on the task of characterising information-seeking events, where *ChatGPT* underperformed compared to transformer models like *XLNet*, especially in domain-specific contexts requiring extensive knowledge.

Zhan et al. introduce *GLEN* [89], a large-scale general-purpose event detection dataset that significantly expands the ontology of event types. While *InstructGPT* underperformed compared to other baselines in their experiments, the authors attribute this to the limited input length and lack of fine-tuning, with only $57.8\%$ of generated event types matching the ontology, similarly to our observations (Section 6.3.1). In 2024, Zhang et al. present *ULTRA* [90], a framework utilising hierarchical modelling and pairwise refinement for document-level event argument extraction.

While the early attempts at utilising LLMs for the complex task of event extraction have shown mixed results, with LLMs often underperforming in comparison to traditional methods, especially in domain-specific contexts, there is a clear trajectory of improvement. As LLMs continue to evolve, gaining the ability to handle larger context windows and as researchers refine their prompting techniques — such as breaking down the task into simpler subtasks as demonstrated in `L-SEE` — the gap between LLMs and traditional methods is expected to narrow. The advancements in hierarchical modelling, pairwise refinement, and modules like LEAFER [90] for argument span refinement indicate the potential for LLMs to improve and catch up to traditional event extraction methodologies in the near future.

#### *7.3.2. LLM-based Knowledge Graph Population*

The use of LLMs for the population of knowledge graphs has also been explored recently. For example, Mihindukulasooriya et al. experimented on ontology-driven triple extraction from sentences [91], while Yao et al. performed instruction tuning for the tasks of triple classification, relation prediction and entity link prediction [92]. In another innovative approach, *AutoKG* leverages a multi-agent-based approach employing LLMs and external sources for KG construction and reasoning [93]. Zhang et al. propose *KoPA*, which ingests entity and relation embeddings into LLMs [94].

These papers about LLM-based information extraction present a glimpse into the rapidly evolving field of LLM-based IE. While promising results have been achieved, further research is needed to address challenges such as factual correctness, bias mitigation, and adapting LLMs to specific domains and tasks. As research progresses, LLMs are set to play a key role in the future of information extraction, enabling efficient and accurate knowledge extraction from vast amounts of unstructured text data.

## 8. Conclusion

In this article, we compared two paradigms for semantic event extraction: Fine-tuning transformer-based architectures as exemplified by our approach `T-SEE` and prompting Large Language Models (LLMs), exemplified by our approach `L-SEE`.

Both approaches consist of two main steps: event classification and relation extraction, where `T-SEE` frames event classification as a multi-label classification task, and conducts relation extraction with a span prediction transformer model. `L-SEE` provides an LLM with two different prompts which include the event classes and properties in the target event ontology.

In our evaluation, we first introduced two new datasets for semantic event extraction. Then, we compare `T-SEE` and `L-SEE` to two state-of-the-art baselines, with `T-SEE` outperforming or matching them and setting a new benchmark for transformer-based methods in semantic event extraction. Finally, we specifically focused on the different characteristics of `T-SEE` and `L-SEE`, highlighting `T-SEE`'s adaptation to the precise characteristics of the training and test data, while `L-SEE` performs clearly worse on the test data. However, our subsequent analysis revealed its capability of extracting relevant knowledge that is often overlooked by distantly-labelled datasets.

Consequently, we derive a set of phenomena to be regarded when performing semantic event extraction, including the role of distantly-labelled datasets and the event ontology.

In future work, we plan to further improve `T-SEE` and `L-SEE`, e.g., by bringing event classification, relation extraction and other tasks like named entity recognition even closer together in joint multi-task learning frameworks and to extend them to encompass multilingual and document-level semantic event extraction. In addition, we aim to enhance metrics and datasets, allowing a fair comparison between semantic event extraction methods employing transformer-based architectures and LLMs.

## References

[1] S. Latif, S. Agarwal, S. Gottschalk, C. Chrosch, F. Feit, J. Jahn, T. Braun, Y.C. Tchenko, E. Demidova and F. Beck, Visually Connecting Historical Figures Through Event Knowledge Graphs, *Computing Research Repository* (2021).

[2] S. Gottschalk and E. Demidova, EventKG+BT: Generation of Interactive Biography Timelines from a Knowledge Graph, in: *The Semantic Web*, 2020.

[3] S. Gottschalk and E. Demidova, EventKG - the Hub of Event Knowledge on the Web - and Biographical Timeline Generation, *The Semantic Web* (2019).

[4] R. Porzel, M. Pomarlan, L. Spillner, J.A. Bateman, T. Mildner and C. Santagiustina, Narrativizing Knowledge Graphs, in: *International Workshop on Artificial Intelligence Technologies for Legal Documents*, 2022.

[5] T. Souza Costa, S. Gottschalk and E. Demidova, Event-QA: A dataset for Event-centric Question Answering over Knowledge Graphs, in: *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 3157–3164.

[6] R. Shaw, R. Troncy and L. Hardman, LODE: Linking open descriptions of events, in: *The Semantic Web: Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings 4*, Springer, 2009, pp. 153–167.

[7] W.R. Van Hage, V. Malaisé, R. Segers, L. Hollink and G. Schreiber, Design and Use of the Simple Event Model (SEM), *Journal of Web Semantics* **9**(2) (2011), 128–136.

[8] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: *The Semantic Web Conference*, 2007.

[9] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez and D. Vrandečić, Introducing Wikidata to the Linked Data Web, in: *International Semantic Web Conference*, 2014.

[10] A. Li, X. Wang, W. Wang, A. Zhang and B. Li, A Survey of Relation Extraction of Knowledge Graphs, in: *Web and Big Data International Workshops*, 2019.

[11] S. Ji, S. Pan, E. Cambria, P. Marttinen and P.S. Yu, A Survey on Knowledge Graphs: Representation, Acquisition and Applications, *Computing Research Repository* (2020).

[12] S. Shirai, D. Bhattacharjya and O. Hassanzadeh, Event Prediction using Case-Based Reasoning over Knowledge Graphs, in: *The Web Conference*, 2023.

[13] G. Stoica, E.A. Platanios and B. Póczos, Improving Relation Extraction by Leveraging Knowledge Graph Link Prediction, *Computing Research Repository* (2020).
[14] W. Xiang and B. Wang, A Survey of Event Extraction from Text, *IEEE Access* (2019).
[15] J. Zheng, F. Cai, W. Chen, W. Lei and H. Chen, Taxonomy-aware Learning for Few-shot Event Detection, in: *The Web Conference*, 2021.
[16] S. Mehta, M.R. Islam, H. Rangwala and N. Ramakrishnan, Event Detection Using Hierarchical Multi-aspect Attention, in: *Web Conference*, 2019.
[17] S. Li, H. Ji and J. Han, Document-level Event Argument Extraction by Conditional Generation, *arXiv preprint arXiv:2104.05919* (2021).
[18] X. Wang, Z. Wang, X. Han, Z. Liu, J. Li, P. Li, M. Sun, J. Zhou and X. Ren, HMEAE: Hierarchical Modular Event Argument Extraction, in: *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 2019.
[19] Y. Ma, Z. Wang, Y. Cao, M. Li, M. Chen, K. Wang and J. Shao, Prompt for extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction, *arXiv preprint arXiv:2202.12109* (2022).
[20] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin and J. Zhou, MAVEN: A Massive General Domain Event Detection Dataset, in: *Empirical Methods in Natural Language Processing*, 2020.
[21] S. Ebner, P. Xia, R. Culkin, K. Rawlins and B. Van Durme, Multi-Sentence Argument Linking, in: *Association for Computational Linguistics*, 2020.
[22] M. Rospocher, M. Van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger and T. Bogaard, Building Event-centric Knowledge Graphs from News, *Journal of Web Semantics* (2016).
[23] K. Guo, D. Diefenbach, A. Gourru and C. Gravier, Wikidata as a Seed for Web Extraction, in: *The Web Conference*, 2023.
[24] S. Guan, X. Cheng, L. Bai, F. Zhang, Z. Li, Y. Zeng, X. Jin and J. Guo, What is Event Knowledge Graph: A Survey, *IEEE Transactions on Knowledge and Data Engineering* (2022).
[25] F. Hamborg, C. Breitinger and B. Gipp, Giveme5w1h: A Universal System for Extracting Main Events from News Articles, *arXiv preprint arXiv:1909.02766* (2019).
[26] Y. Zhou, Y. Chen, J. Zhao, Y. Wu, J. Xu and J. Li, What the Role is vs. What Plays the Role: Semi-Supervised Event Argument Extraction via Dual Question Answering, in: *AAAI*, 2021.
[27] Z. Wang, X. Wang, X. Han, Y. Lin, L. Hou, Z. Liu, P. Li, J. Li and J. Zhou, CLEVE: Contrastive Pre-training for Event Extraction, *arXiv preprint arXiv:2105.14485* (2021).
[28] Linguistic Data Consortium, *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 2005.
[29] A.M. Davani, L. Yeh, M. Atari, B. Kennedy, G. Portillo-Wightman, E. Gonzalez, N. Delong, R. Bhatia, A. Mirinjian, X. Ren et al., Reporting the Unreported: Event extraction for Analyzing the Local Representation of Hate Crimes, *arXiv preprint arXiv:1909.02126* (2019).
[30] R. Xu, T. Liu, L. Li and B. Chang, Document-level Event Extraction via Heterogeneous Graph-based Interaction Model with a Tracker, in: *Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2021.
[31] P. Kaliamoorthi, A. Siddhant, E. Li and M. Johnson, Distilling Large Language Models into Tiny and Effective Students using pQRNN, *arXiv preprint arXiv:2101.08890* (2021).
[32] A. Dunn, J. Dagdelen, N. Walker, S. Lee, A.S. Rosen, G. Ceder, K. Persson and A. Jain, Structured Information Extraction from Complex Scientific Text with Fine-tuned Large Language Models, *arXiv preprint arXiv:2212.05238* (2022).
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention Is All You Need, *Advances in Neural Information Processing Systems* (2017).
[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, Focal Loss for Dense Object Detection, in: *international conference on computer vision*, 2017.
[35] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv:1810.04805* (2018).
[36] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, in: *International Conference on Learning Representations*, 2019.
[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, *arXiv preprint arXiv:1907.11692* (2019).
[38] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov and Q.V. Le, Xlnet: Generalized Autoregressive Pretraining for Language Understanding, *Advances in Neural Information Processing Systems* (2019).
[39] R. Hoekstra and P. Groth, PROV-O-Viz-understanding the Role of Activities in Provenance, in: *Provenance and Annotation of Data and Processes*, 2015.
[40] L. Guzman-Nateras, V. Lai, A. Pouran Ben Veyseh, F. Dernoncourt and T. Nguyen, Event Detection for Suicide Understanding, in: *North American Association for Computational Linguistics*, 2022.
[41] Y. Luan, L. He, M. Ostendorf and H. Hajishirzi, Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction, in: *Empirical Methods in Natural Language Processing*, 2018.
[42] T. Ohta, Y. Tateisi, J.-D. Kim, H. Mima and J. Tsujii, The GENIA corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain, in: *Human Language Technology Conference*, Citeseer, 2002.
[43] A. Pouran Ben Veyseh, M.V. Nguyen, F. Dernoncourt and T. Nguyen, MINION: a Large-Scale and Diverse Dataset for Multilingual Event Detection, in: *North American Chapter of the Association for Computational Linguistics:*, 2022.
[44] K. Leetaru and P.A. Schrodt, GDELT: Global Data on Events, Location, and Tone, in: *ISA Annual Convention*, 2013.
[45] M. Li, R. Xu, S. Wang, L. Zhou, X. Lin, C. Zhu, M. Zeng, H. Ji and S.-F. Chang, Clip-event: Connecting Text and Images with Event Structures, in: *Conference on Computer Vision and Pattern Recognition*, 2022.

[46] Y. Lu, H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao and S. Chen, Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction, in: *Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2021.

[47] H. You, D. Samuel, S. Touileb and L. Øvrelid, EventGraph: Event Extraction as Semantic Graph Parsing, *arXiv preprint arXiv:2210.08646* (2022).

[48] F. Li, W. Peng, Y. Chen, Q. Wang, L. Pan, Y. Lyu and Y. Zhu, Event Extraction as Multi-turn Question Answering, in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 829–838.

[49] X. Liu, H.-Y. Huang, G. Shi and B. Wang, 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)*, 2022, pp. 5216–5228.

[50] H. Huang, X. Liu, G. Shi and Q. Liu, Event Extraction with Dynamic Prefix Tuning and Relevance Retrieval, *IEEE Transactions on Knowledge and Data Engineering* (2023).

[51] S. Liu, Y. Li, F. Zhang, T. Yang and X. Zhou, Event Detection Without Triggers, in: *North American Chapter of the Association for Computational Linguistics*, 2019.

[52] X. Du and C. Cardie, Event Extraction by Answering (Almost) Natural Questions, in: *Empirical Methods in Natural Language Processing*, 2020.

[53] I. Hsu, K.-H. Huang, E. Boschee, S. Miller, P. Natarajan, K.-W. Chang, N. Peng et al., DEGREE: A Data-efficient Generation-based Event Extraction Model, *arXiv preprint arXiv:2108.12724* (2021).

[54] D. Lu, S. Ran, J. Tetreault and A. Jaimes, Event Extraction as Question Generation and Answering, *arXiv preprint arXiv:2307.05567* (2023).

[55] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang et al., Zero-shot Information Extraction via Chatting with Chatgpt, *arXiv preprint arXiv:2302.10205* (2023).

[56] D. Wadden, U. Wennberg, Y. Luan and H. Hajishirzi, Entity, Relation, and Event Extraction with Contextualized Span Representations, in: *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019.

[57] C. Wang, X. Liu, Z. Chen, H. Hong, J. Tang and D. Song, DeepStruct: Pretraining of Language Models for Structure Prediction, *arXiv preprint arXiv:2205.10475* (2022).

[58] X. Han, T. Gao, Y. Lin, H. Peng, Y. Yang, C. Xiao, Z. Liu, P. Li, J. Zhou and M. Sun, More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction, in: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 745–758.

[59] B. Goodrich, V. Rao, P.J. Liu and M. Saleh, Assessing the Factual Accuracy of Generated Text, in: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 166–175.

[60] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou and M. Sun, DocRED: A Large-Scale Document-Level Relation Extraction Dataset, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 764–777.

[61] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest and E. Simperl, T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples, in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[62] K. Mai, T.-H. Pham, M.T. Nguyen, T.D. Nguyen, D. Bollegala, R. Sasano and S. Sekine, An Empirical Study on Fine-grained Named Entity Recognition, in: *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 711–722.

[63] S. Ji, S. Pan, E. Cambria, P. Marttinen and P. Yu, A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[64] S. Gottschalk and E. Demidova, EventKG: a Multilingual Event-centric Temporal Knowledge Graph, in: *Extended Semantic Web Conference*, 2018.

[65] S. Gottschalk, E. Kacupaj, S. Abdollahi, D. Alves, G. Amaral, E. Koutsiana, T. Kuculo, D. Major, C. Mello, G.S. Cheema et al., OEKG: The Open Event Knowledge Graph, in: *CLEOPATRA Workshop co-located with the 30th The Web Conference*, 2021.

[66] D.J. Gerner, P.A. Schrodt, O. Yilmaz and R. Abu-Jabr, Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for the Analysis of Foreign Policy Interactions, *International Studies Association* (2002).

[67] H. Zhang, X. Liu, H. Pan, Y. Song and C.W.-K. Leung, ASER: A Large-scale Eventuality Knowledge Graph, in: *The Web Conference*, 2020.

[68] W. Yao, Z. Dai, M. Ramaswamy, B. Min and R. Huang, Weakly Supervised Subevent Knowledge Acquisition, in: *Empirical Methods in Natural Language Processing*, 2020.

[69] R. Piryani, N. Aussenac-Gilles and N. Hernandez, Comprehensive Survey on Ontologies about Event, in: *ESWC Workshops on Semantic Methods for Events and Stories (SEMMES@ ESWC 2023)*, Vol. 3443, ceur-ws. org, 2023, pp. 1–15.

[70] V.A. Carriero, A. Gangemi, A.G. Nuzzolese, V. Presutti et al., An Ontology Design Pattern for Representing Recurrent Situations., in: *WOP (Book)*, 2021, pp. 166–182.

[71] A. Krisnadhi and P. Hitzler, A Core Pattern for Events, *Advances in Ontology Design and Patterns* **32** (2017), 29–38.

[72] X. Du, A.M. Rush and C. Cardie, GRIT: Generative Role-filler Transformers for Document-level Event Entity Extraction, in: *European Chapter of the Association for Computational Linguistic*, 2021.

[73] J. Liu, Y. Chen, K. Liu, W. Bi and X. Liu, Event Extraction as Machine Reading Comprehension, in: *Empirical Methods in Natural Language Processing*, 2020.

[74] S. Yang, D. Feng, L. Qiao, Z. Kan and D. Li, Exploring Pre-trained Language Models for Event Extraction and Generation, in: *Association for Computational Linguistics*, 2019.

[75] S. Zheng, W. Cao, W. Xu and J. Bian, Doc2EDAG: An End-to-End Document-level Framework for Chinese Financial Event Extraction, in: *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019.

[76] D. Lou, Z. Liao, S. Deng, N. Zhang and H. Chen, MLBiNet: A Cross-Sentence Collective Event Detection Network, in: *Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 2021.

[77] X. Liu, H.-Y. Huang and Y. Zhang, Open Domain Event Extraction Using Neural Latent Variable Models, in: *Association for Computational Linguistics*, 2019.

[78] D. Rusu, J. Hodson and A. Kimball, Unsupervised Techniques for Extracting and Clustering Complex Events in News, in: *EVENTS*, 2014.

[79] Y. Ma, Z. Wang, Y. Cao, M. Li, M. Chen, K. Wang and J. Shao, Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction, in: *The Association for Computational Linguistics*, Association for Computational Linguistics, 2022.

[80] J. Zhao and H. Yang, Trigger-free Event Detection via Derangement Reading Comprehension, *CoRR* (2022).

[81] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, Deep Contextualized Word Representations, in: *The North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[82] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., Language Models are Few-shot Learners, *Advances in neural information processing systems* (2020).

[83] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers et al., In-datacenter Performance Analysis of a Tensor Processing Unit, in: *international symposium on computer architecture*, 2017.

[84] M.P. Polak and D. Morgan, Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering, *Nature Communications* (2024).

[85] B. Li, G. Fang, Y. Yang, Q. Wang, W. Ye, W. Zhao and S. Zhang, Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness, *arXiv preprint arXiv:2304.11633* (2023).

[86] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu et al., Summary of Chatgpt-related Research and Perspective Towards the Future of Large Language Models, *Meta-Radiology* (2023), 100017.

[87] C. Kirti, A. Chattopadhyay, A. Anand and P. Guha, Deciphering Storytelling Events: A Study of Neural and Prompt-Driven Event Detection in Short Stories, in: *2023 International Conference on Asian Language Processing (IALP)*, IEEE, 2023, pp. 106–111.

[88] O. Sharif, M. Basak, T. Parvin, A. Scharfstein, A. Bradham, J.T. Borodovsky, S.E. Lord and S.M. Preum, Characterizing Information Seeking Events in Health-Related Social Discourse, *arXiv preprint arXiv:2308.09156* (2023).

[89] S. Li, Q. Zhan, K. Conger, M. Palmer, H. Ji and J. Han, GLEN: General-Purpose Event Detection for Thousands of Types, in: *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[90] X.F. Zhang, C. Blum, T. Choji, S. Shah and A. Vempala, ULTRA: Unleash LLMs' Potential for Event Argument Extraction through Hierarchical Modeling and Pair-wise Refinement, *arXiv preprint arXiv:2401.13218* (2024).

[91] N. Mihindukulasooriya, S. Tiwari, C.F. Enguix and K. Lata, Text2kgbench: A Benchmark for Ontology-driven Knowledge Graph Generation from Text, in: *International Semantic Web Conference*, Springer, 2023, pp. 247–265.

[92] L. Yao, J. Peng, C. Mao and Y. Luo, Exploring Large Language Models for Knowledge Graph Completion, *arXiv preprint arXiv:2308.13916* (2023).

[93] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen and N. Zhang, LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities, *arXiv preprint arXiv:2305.13168* (2023).

[94] Y. Zhang, Z. Chen, W. Zhang and H. Chen, Making Large Language Models Perform Better in Knowledge Graph Completion, *arXiv preprint arXiv:2310.06671* (2023).

[95] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: A Core of Semantic Knowledge, in: *The Web Conference*, 2007.

[96] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf and H. Hajishirzi, A General Framework for Information Extraction using Dynamic Span Graphs, in: *North American Chapter of the Association for Computational Linguistics*, 2019.

[97] J. Liu, L. Min and X. Huang, An Overview of Event Extraction and its Applications, *arXiv preprint arXiv:2111.03212* (2021).

[98] S.T.R. Rizvi, D. Mercier, S. Agne, S. Erkel, A. Dengel and S. Ahmed, Ontology-based Information Extraction from Technical Documents., in: *ICAART*, 2018.

[99] K. Guo, D. Diefenbach, A. Gourru and C. Gravier, Wikidata as a Seed for Web Extraction, in: *The Web Conference*, 2023.

[100] C. Walker, S. Strassel, J. Medero and K. Maeda, ACE 2005 Multilingual Training Corpus, *Linguistic Data Consortium, Philadelphia* (2006).

[101] Z. Li, S. Zhao, X. Ding and T. Liu, EEG: Knowledge Base for Event Evolutionary Principles and Patterns, in: *Chinese National Conference on Social Media Processing*, 2017.

[102] S. Abdollahi, S. Gottschalk and E. Demidova, EventKG+ Click: A Dataset of Language-specific Event-centric User Interaction Traces, in: *CLEOPATRA Workshop co-located with the 17th Extended Semantic Web Conference*, 2020.

[103] Q. Ning, B. Zhou, Z. Feng, H. Peng and D. Roth, CogCompTime: A Tool for Understanding Time in Natural Language, in: *Empirical Methods in Natural Language Processing*, 2018.

[104] M.D. Ma, J. Sun, M. Yang, K.-H. Huang, N. Wen, S. Singh, R. Han and N. Peng, EventPlus: A Temporal Event Understanding Pipeline, in: *North American Chapter of the Association for Computational Linguistics*, 2021.

[105] H. Wang, H. Zhang, M. Chen and D. Roth, Learning Constraints and Descriptive Segmentation for Subevent Detection, in: *Empirical Methods in Natural Language Processing*, 2021.

[106] P. Rajpurkar, R. Jia and P. Liang, Know What You Don't Know: Unanswerable Questions for SQuAD, in: *Association for Computational Linguistics*, 2018.

[107] L. Qiu, H. Zhou, Y. Qu, W. Zhang, S. Li, S. Rong, D. Ru, L. Qian, K. Tu and Y. Yu, QA4IE: a Question Answering based Framework for Information Extraction, in: *International Semantic Web Conference*, 2018.
[108] X. Li, F. Yin, Z. Sun, X. Li, A. Yuan, D. Chai, M. Zhou and J. Li, Entity-Relation Extraction as Multi-Turn Question Answering, in: *Association for Computational Linguistics*, 2019.
[109] A. Delpeuch, OpenTapioca: Lightweight Entity Linking for Wikidata, in: *Wikidata Workshop co-located with International Semantic Web Conference*, 2020.
[110] B. Xu, Y. Zhang, J. Liang, Y. Xiao, S.-w. Hwang and W. Wang, Cross-lingual Type Inference, in: *International Conference on Database Systems for Advanced Applications*, 2016.
[111] M.D. Ward, N.W. Metternich, C. Carrington, C. Dorff, M. Gallop, F.M. Hollenbach, A. Schultz, S. Weschle et al., Geographical Models of Crises: Evidence from ICEWS, *Design for Cross-Cultural Activities* (2012).
[112] J.L. Martinez-Rodriguez, A. Hogan and I. Lopez-Arevalo, Information Extraction Meets the Semantic Web: a Survey, *Semantic Web* (2020).
[113] P.-L.H. Cabot and R. Navigli, REBEL: Relation Extraction by End-to-end Language Generation, in: *Empirical Methods in Natural Language Processing*, 2021.
[114] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu and M. Sun, OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction, in: *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019.
[115] K. Verspoor, J. Cohn, S. Mniszewski and C. Joslyn, A Categorization Approach to Automated Ontological Function Annotation, *Protein Science* (2006).
[116] G. Rossiello, F. Chowdhury, N. Mihindukulasooriya, O. Cornec and A. Gliozzo, KnowGL: Knowledge Generation and Linking from Text, *arXiv preprint arXiv:2210.13952* (2022).
[117] T. Liu, Y.E. Jiang, N. Monath, R. Cotterell and M. Sachan, Autoregressive Structured Prediction with Language Models, in: *Empirical Methods in Natural Language Processing*, 2022.
[118] Z. Zhong and D. Chen, A Frustratingly Easy Approach for Entity and Relation Extraction, in: *North American Chapter of the Association for Computational Linguistics*, 2021.
[119] D. Ye, Y. Lin, P. Li and M. Sun, Packed Levitated Marker for Entity and Relation Extraction, in: *Association for Computational Linguistics*, 2022.
[120] J. Chen, H. Lin, X. Han and L. Sun, Honey or Poison? Solving the Trigger Curse in Few-shot Event Detection via Causal Intervention, in: *Empirical Methods in Natural Language Processing*, 2021.
[121] M. Tong, B. Xu, S. Wang, M. Han, Y. Cao, J. Zhu, S. Chen, L. Hou and J. Li, DocEE: A Large-scale and Fine-grained Benchmark for Document-level Event Extraction, Association for Computational Linguistics, 2022.
[122] T. Ge, L. Cui, B. Chang, Z. Sui, F. Wei and M. Zhou, Eventwiki: A Knowledge Base of Major Events, in: *International Conference on Language Resources and Evaluation*, 2018.
[123] J. Ellis, J. Getman, D. Fore, N. Kuster, Z. Song, A. Bies and S.M. Strassel, Overview of Linguistic Resources for the TAC KBP 2015 Evaluations: Methodologies and Results., in: *Tac*, 2015.
[124] J. Zheng, F. Cai, W. Chen, W. Lei and H. Chen, Taxonomy-aware Learning for Few-Shot Event Detection, in: *The Web Conference*, 2021.
[125] S. Zhou, B. Yu, A. Sun, C. Long, J. Li and J. Sun, A Survey on Neural Open Information Extraction: Current Status and Future Directions, in: *International Joint Conference*, 2022.
[126] V. Perot, K. Kang, F. Luisier, G. Su, X. Sun, R.S. Boppana, Z. Wang, J. Mu, H. Zhang and N. Hua, LMDX: Language Model-based Document Information Extraction and Localization, *arXiv preprint arXiv:2309.10952* (2023).
[127] N. Heist, S. Hertling, D. Ringler and H. Paulheim, Knowledge Graphs on the Web-An Overview., *Knowledge Graphs for eXplainable Artificial Intelligence* (2020), 3–22.

# Appendix A. Prompts of `L-SEE`

This appendix shows the prompts used by `L-SEE` as described in Sections 4.1 and 4.2.

## A.1. Prompt for Event Classification

Fig. 16 shows the LLM prompt template we use for `L-SEE`'s event classification as described in Section 4.1.

> LLM Prompt for Event Classification
>
> **Prompt:**
> *Your task is to analyse the sentence and classify events that are in the sentence.*
> *An event is identified by an action or a mention of an event.*
> *You will only consider events that are likely to have their own Wikipedia page.*
>
> *For example, in the sentence "John married Mary in Paris on 12th December 2020 during the Parisian Unrests", the events are "marriage" and "unrests". However, "married" is not to be considered an event as it is unlikely to have its own Wikipedia page.*
>
> *You are to select event types from the following list of event types and return it as a list of strings of event types:*
>
> *{All Event Classes C}*
>
> *Note: The events that you should identify are links in Wikipedia, they may not be referred to directly by name in the sentence but a specific word or phrase in the sentence may link to the event. E.g., in "Senator McCain also got 10% higher approval rating compared to 2010", 2010 is a link to the event "United States Senate elections, 2010" even though it is not mentioned directly in the sentence.*

Fig. 16. Illustration of a structured prompt provided to the LLM for event classification.

### *A.2. Prompt for Relation Extraction*

Fig. 17 shows the LLM prompt template we use for `L-SEE`'s relation extraction as described in Section 4.2.

**LLM Prompt for Relation Extraction**

**Prompt:**

*Your task is to extract the properties of the events that are in a given sentence and their values. You will only consider properties that are likely to be associated with the given event classes. Extract the properties of the events and return a JSON object with the event classes as the keys and the properties as the values.*
*The property values can be dates, entities, or quantities. If there is no specific value for a property, you must not include it in the JSON object.*
*The extracted property values must fit their respective property types. For example, if the property is "date", the value must be able to be formatted as a date (e.g. "12th December 2019" or "2019" in the case of a year).*
*Similarly, if the property is "location", the value must be a location. If there are multiple values for a property, you must include all the values in a list.*

*Consider the following example:*

*Sentence:*
*John married Mary on the first day of the start of the COVID-19 pandemic, on 12th December 2019. It was only a few days later that in the winter of 2019, the German-French War destroyed the cities of Paris and Berlin.*

*Event classes and their potential properties:*
*- Pandemic: city, startDate*
*- MilitaryConflict: city, date, participant*

*Output:*

```
{
    "Pandemic": {
        "startDate": ["12th December 2019"]
    },
    "MilitaryConflict": {
        "city": ["Paris", "Berlin"],
        "date": ["2019"]
    }
}
```

*This is your task:*

*Sentence:*
*{Text t}*

*Event classes and their potential properties:*
{Event classes $V_t.C$ and properties $V_t.P$}

Fig. 17. Illustration of a structured prompt provided to the LLM for property extraction.