EURECOM
Sophia Antipolis

SORBONNE UNIVERSITÉ

PHD THESIS

In Partial Fulfilment of the Requirements for the
Degree of Doctor of Philosophy from Sorbonne University
Specialization: Data Science

Semantic Extraction of Event Relations from Text with Knowledge Graphs

Youssra REBBOUD

Defended on 30/06/2025 before a committee composed of:

Reviewer	**Elena DEMIDOVA**,University of Bonn, Germany
Reviewer	**Enrico MOTTA**, Open University, UK
Examiner	**Paolo PAPOTTI**, EURECOM, France
Examiner	**Serena VILLATA**, INRIA, Sophia Antipolis, France
Thesis Director	**Maria ZULUAGA**, EURECOM, France
Thesis Co-Director	**Pasquale LISENA**, EURECOM, France
Thesis Co-Director	**Raphael TRONCY**, EURECOM, France

First and foremost, I would like to express my deepest gratitude to my supervisors, ***Pasquale Lisena*** *and* ***Raphaël Troncy****, for their invaluable guidance, insightful advice, and constant support throughout this journey.*

This work is lovingly dedicated to the memory of my dear mother, whose spirit continues to guide me every day.

To my family, both small and extended, for being my foundation and strength.

To my husband, for his love, patience, and unwavering support.

To my father, sister, and brother, for their endless encouragement and care.

To my friends, whose good spirit and joy made this journey brighter and far less boring.

This achievement is as much theirs as it is mine.

Acknowledgements

This work has been partially supported by the French National Research Agency (ANR) within the kFLOW project (Grant n°ANR-21-CE23-0028).

Abstract

This research tackles the scientific challenge of accurately extracting and modelling event flows from textual data, a capability essential for informed decision-making, historical reconstruction, and predictive analysis. We introduce **FARO**, an ontology that structures 25 refined relationships between events, harmonizing existent events data models, including refined causal types such as *direct causality*, *enablement*, *prevention*, and *intention*. This formalization supports richer semantic interpretations of textual narratives.

To enable robust extraction of such relations, we construct a large-scale annotated dataset of over 500,000 sentences, generated using Large Language Models (LLMs), commonsense knowledge from the `ATOMIC` knowledge graph, and generative-ai techniques. Leveraging this resource, we develop and evaluate an extraction framework capable of identifying fine-grained event relations, leveraging structured knowledge and contextual cues to capture complex event dynamics.

We validate the practical impact of our approach through two applications: (1) enhanced narrative generation via structured, semantically rich knowledge graphs, and (2) explainable fact-checking supported by causal reasoning. Our contributions provide both foundational resources and methodological advances for event-centric natural language understanding.

Abrégé

Cette recherche aborde le défi scientifique de l'extraction et de la modélisation précises des enchaînements d'événements à partir de données textuelles, une capacité essentielle pour la prise de décision éclairée, la reconstruction historique et l'analyse prédictive. Nous présentons **FARO**, une ontologie qui structure 25 relations affinées entre événements, en harmonisant les modèles de données événementielles existants, y compris des types causaux raffinés tels que la *causalité directe*, l'*activation*, la *prévention* et l'*intention*. Cette formalisation permet des interprétations sémantiques plus riches des récits textuels.

Pour permettre une extraction robuste de ces relations, nous construisons un jeu de données annoté à grande échelle de plus de 500 000 phrases, généré à l'aide de LLMs, de connaissances de sens commun issues du graphe de connaissances ATOMIC, ainsi que de techniques d'intelligence artificielle générative. En exploitant cette ressource, nous développons et évaluons un cadre d'extraction capable d'identifier des relations événementielles fines, en s'appuyant sur des connaissances structurées et des indices contextuels pour capturer la dynamique complexe des événements.

Nous validons l'impact pratique de notre approche à travers deux applications : (1) la génération de récits enrichis à l'aide de graphes de connaissances structurés et sémantiquement riches, et (2) la vérification automatique d'informations explicable, appuyée par un raisonnement causal. Nos contributions fournissent à la fois des ressources fondamentales et des avancées méthodologiques pour la compréhension du langage naturel centrée sur les événements.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence.

AWO the African wildlife ontology.

AWQ Activation-aware Weight Quantization.

CoT chain-of-thought.

CQ Competency Question.

EE Event Extraction.

ERE Event Relation Extraction.

FARO Facts and Events Relationship Ontology.

KG Knowledge Graph.

LLM Large Language Model.

LM Language Model.

LOV Linked Open Vocabulary.

NER Named Entity Recognition.

NIF NLP Interchange Format.

NLP Natural Language Processing.

OEA Overall Execution Accuracy.

OntoDT Generic Ontology of Datatypes.

PLM Pre-trained Language Model.

Pol Polarity.

RC Relation Classification.

RD Relation Detection.

SEM Simple Event Model.

SWO the Software Ontology.

TKG Temporal Knowledge Graph.

Chapter 1

Introduction

1.1 Motivation

Events play a fundamental role in shaping our lives, encompassing history, knowledge, impacts, and future developments, either through their individual existence or their relationships with one another. Our experience of the world is characterized by a continuous sequence of events, where newly observed events can be linked to one or more prior or future occurrences. These connections give rise to various types of relationships, such as cause-effect, relatedness, and co-occurrence in time or space.

Effectively capturing the chain of interconnected events surrounding specific phenomena can be highly complex yet essential for diverse applications, spanning from general public interest to specialized professional needs. Consider the economic implications of the United Kingdom's decision to leave the European Union, commonly known as Brexit. For instance, one may come across an article discussing Brexit, providing basic context or even linking to its definition. While this helps clarify the term itself, the broader implications and connections with other events might still be unclear. On one news page, Brexit could be presented as causing trade disruptions with the European Union. Elsewhere, another article might discuss how these trade disruptions subsequently led to inflationary pressures in the UK, explicitly highlighting significant rises in prices—such as housing increasing by 19.4%, transportation costs by 13%, and food prices by 8.6% as illustrated in Figure 1.1.

Thus, the ability to efficiently represent, retrieve, interpret, and analyze these event relationships is crucial, underscoring the importance of developing advanced methods and tools that support users in navigating complex informational landscapes.

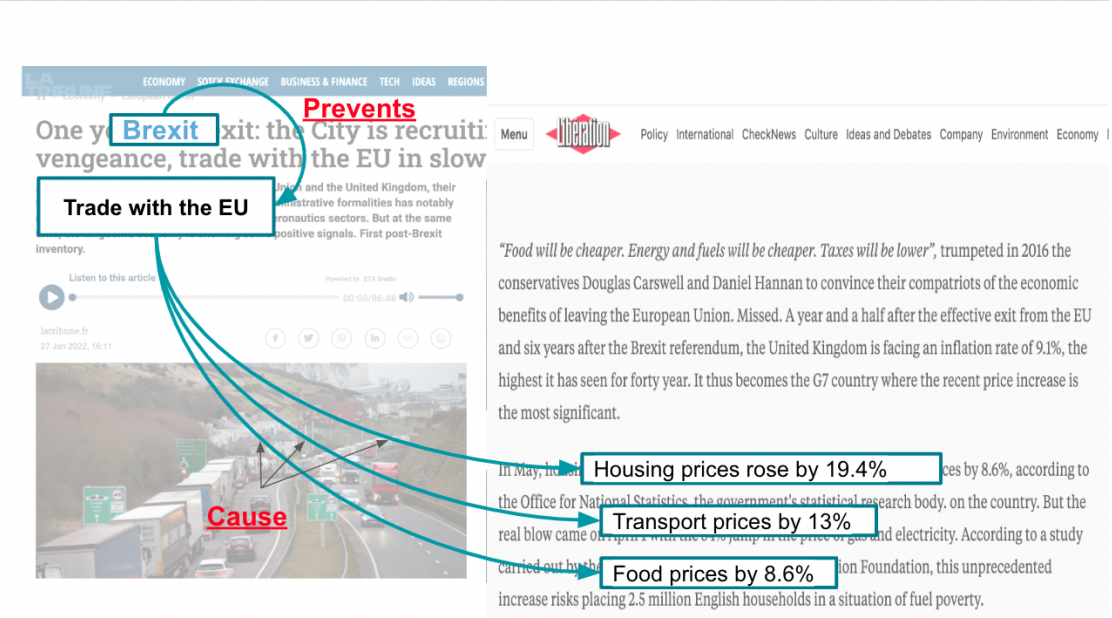

Figure 1.1: Brexit impact on the UK economy.
Image source: `https://shorturl.at/lxiV3`

Existing data models, such as *EventKG* [34], offer large-scale integration of events with temporal and spatial annotations, organizing them through hierarchical structures like sub and superevents and the 4W framework (Who, What, When, Where). However, despite capturing the mentioned relations, they often fall short of expressing how events influence one another. For instance, while it could record that Brexit happened in 2016 and another that inflation rose in the UK, these facts remain loosely connected unless explicitly linked through refined causal chains—such as Brexit *prevented* trades, which in turn *caused* economic repercussions. Without such semantically rich and fine-grained relationships, the full flow of events remains inaccessible to both humans and machines.

This highlights the need for a more **expressive** and **harmonized** model to represent event relationships, one that can bridge isolated data points into coherent narratives to navigate event flows with semantic precision.

This level of details is particularly valuable in contexts where the coherence, completeness, and interpretability of event flows are critical. For instance, in applications like narrative generation and fact-checking. In narratives generation tasks, existing models are limiting our ability to reconstruct meaningful narratives. Refined causal links may enrich the structure and meaning of the story, allowing for deeper insights into how and why events unfold. In fact-checking, the refined representations of causality potentially offer a more principled way to detect inconsistencies between claims and supporting evidence, going beyond surface-level textual similarity.

In the context of the *kFLOW*[1] project we had the chance to model, extract, and exploit fine-grained relationships between events in order to build semantically rich knowledge graphs that enable deeper understanding, reasoning, and explanation over evolving information.

1.2 Problem Statement and Research Questions

In the literature, various studies have explored event relationships, with **temporality** being the most commonly represented type [34, 79]. Other works have focused on the interaction between super-events and sub-events [63], as well as comparative relations [44]. While there has been increasing attention toward causality [130], there is currently no ontology that consolidates and systematically integrates the diverse event relationships described in existing research. Furthermore, their description of causality is either vague or, in cases where it is more refined, some relations are missing or lack proper definitions.

This rises our first research Question, **RQ1: How can existing event relation data models be harmonized into a unified ontology while ensuring maximal completeness, and semantic depth for causal relations?**

To build systems capable of reasoning about event dynamics such as understanding, predicting, or explaining event flows, it is essential to have high quality datasets that capture inter event relations with semantic precision. Without such data, it becomes difficult to train or evaluate models, especially for relations like refined causality that is missing in the literature.

While the Semantic Web provides methods and tools for representing facts in Knowledge Graphs (KGs), typically expressed in RDF, most existing KGs focus on isolated events. Some are tailored to event-centric knowledge [36], and Temporal Knowledge Graphs (TKGs) enrich these structures by associating each triple with temporal information. This enables the representation of event occurrences over time and supports tasks such as edge inference [50, 116]. However, despite their temporal expressiveness, TKGs are not well-suited for representing inter-event relationships, which limits their ability to capture the flow and interaction of events.

Existing datasets such as TimeBank [121], or EventStoryLine [16] offer a foundation for modeling event occurrences. Yet, they often lack fine-grained, semantically rich relations between events. Where such relations are included, they tend to be sparse, vaguely defined, or incomplete.

This reveals a key gap and leads us to our second research question: **RQ2: How can data be constructed, reannotated, reused, or improved to build a refined causality knowledge graph?**

[1] https://anr-kflow.github.io/

The extraction of these events, along with the relationships between them, has been a long-standing focus of interest within the research community under the task named Event Relation Extraction (ERE). Various scientific challenges have been proposed including the Event Nugget Detection task at TAC [78] and the Causality Identification task at CASE [113]. The existing literature has primarily studied temporal and sub-event relationships between events – such as in the TempEval initiative [118] – with recent attention turning towards causality [131]. However, the precise relations found limited space in research. This can be attributed to two factors: the lack of datasets representing these refined event relations, and the complexity of understanding these relations from text. **RQ3: Which approach should be use to extract refined causal relations from text?**

When considering the downstream applications of this research, examining the role of generated narratives can be particularly helpful. The generation and comprehension of narratives are increasingly shaped by advancements in Artificial Intelligence (AI). Language Models (LMs), such as BERT [26], GPT-3 [15], and the more recent ChatGPT (GPT-3.5)[2], have demonstrated remarkable capabilities in text generation and conversational tasks. However, these models, trained on vast and diverse datasets from undisclosed sources, exhibit inherent limitations, including knowledge gaps, inaccuracies, and societal biases [15, 27].

To address these challenges, KGs provide a structured and machine-readable representation of human knowledge, ensuring adaptability and reliability. The integration of KGs with AI-driven text generation has been widely explored in the literature [49], demonstrating their potential to enhance narrative construction through a more structured and knowledge-aware approach. However, while existing KGs are effective, they often represent only a limited set of relations, such as temporal dependencies, vague causality, or sub-super event relations. As a result, generated narratives from these KGs may lack semantic depth.

On the other hand, and despite recent advances in automated fact-checking, many existing systems struggle to provide transparent and interpretable explanations that align with human reasoning. A key challenge lies in the limited use of semantically precise event relations—such as *cause*, *prevent*, *intend*, and *enable*—which are crucial for understanding the consistency between claims and supporting evidence. Current explainability methods often overlook these nuanced relationships, leading to explanations that are either overly generic or fail to capture the underlying logic of the claim.

These two potential applications of our ERE system rises the following research questions, **RQ:4 Is the representation of fine-grained event relations beneficial for downstream tasks such as narrative generation and fact-checking?**

[2] https://openai.com/blog/chatgpt/

1.3 Contributions

This thesis presents the following key contributions:

- As an outcome of the literature review, we introduce **Facts and Events Relationship Ontology (FARO)**, to harmonize existing data models when it comes to event relations and be the most complete ontology.
- We provide a **dataset** of over 500,000 sentences annotated with five event relation types.
- We propose a **model** capable of accurately extracting fine-grained causality event relations from text, including direct causality, enabling, prevention, intention.
- The demonstration the effectiveness of our event relation extraction system in two downstream **applications**: enhancing the semantic understanding of **generative AI** and improving explainability in **fact-checking**

This manuscript is organized into three main parts.

Part I addresses our first research question. We begin by introducing and discussing our event-centric data model in Chapter 2. We then explore the role of Large Language Models (LLMs) in data engineering, in Chapter 3.

Part II focuses on the Natural Language Processing (NLP) aspects of our work. In Chapter 4, we present various strategies for constructing datasets enriched with refined event relations. Chapter 5 details our event relation extraction system, including the models and evaluation techniques used.

Part III showcases the downstream applications of our ERE system. In Chapter 6, we explore its use in enhancing narrative generation, and in Chapter 7, we demonstrate its contribution to explainable fact-checking through structured causal reasoning.

Finally we conclude and we provide future work research directions in Chapter 8.

Part I

Knowledge Engineering

Chapter 2

Representing Event Relations in Knowledge Graphs

Dynamic environments can be modeled as a series of events and facts that interact with each other, these interactions being characterised by different relations including temporal and causal ones. These have largely been studied in knowledge management, information retrieval or NLP, leading to several strategies aiming at extracting these relationships in textual documents. However, more relation types exist between events, which are insufficiently covered by existing data models and datasets if one needs to train a model to recognise them. In this chapter, we use semantic web technologies to design FARO, an ontology for representing event and fact relations. FARO allows representing up to 25 distinct relationships (including logical constraints), making it a possible bridge between (otherwise incompatible) datasets. We describe the modeling approach of this ontology resource.

2.1 Data Models: Ontologies and Datasets for Events and Relations

In the literature, several works have studied event relationships, the most common type of relationships being **temporality**. Fan et al. [29] identified 13 temporal relations – to be used in the context of 3D simulation –, including simultaneity (*equal*) and 6 other asymmetric (directed) properties, with their respective inverse – e.g. *before* / *after*. Equivalent relations are included in [44], with the addition of *Vagueness*. **Mereology** in the context of events – i.e. the interaction between sub-events and super-events – is also often represented [34,44,63,101,119]. Finally, the literature mentions other kinds of relation that we can group under the name of **contingency**. Wolf distinguishes the causality relations in four different concepts [130]:

- CAUSE: event A that leads to an event B;
- ENABLE: condition C to make an event B possible;
- PREVENT: event A that avoids an event B;

- DESPITE: event A did not succeed in avoiding an event B.

Hong et al. [44] designed one of the most complete event-event relationship classification, including 5 types (Inheritance, Expansion, Contingency, Comparison, Temporality) and 21 sub-types, with possible overlaps between classes. To the best of our knowledge, this is the only work including **comparative relations** which cover three kinds of relation types, such as *Opposition*, when two events are improbable to be both true ($parole \rightarrow sentenced$), *Negation*, when two events can be both true in different time slots, but not simultaneously (A is behind bars →A left). However, several relations between events are not accompanied by proper definitions, while still some relation types are missing.

Several ontologies have been published using semantic web technologies. While some of them do not include relations between events (e.g. LODE [106]), most of them include at least the concept of sub-events, such as in the *Event Pattern* [63], the *Event Ontology*[1], and the *Simple Event Model (SEM)*.

Event Model F is an ontology created to support the response in emergency events [101]. It includes three kind of event relationships: mereological, causal and correlation. Its *Justification* class enables to support the relationship with provenance – e.g. opinion, scientific law, etc. However, this is modeled by including classes – e.g. `EventCompositionSituation` and `EventCompositionDescription` – with the only purposes of connecting events and defining their roles. As a consequence, there are no direct links between the composite super-event and its components sub-events (same for cause-effect). Furthermore, only 1-1 relations are foreseen, so additional instances must be created for aggregating causes/effects. All this led to a complex model, hard to understand and to adopt.

One of the most popular models among libraries and cultural institutions is *CIDOC CRM* [28]. It is an event-centric model, in which everything is represented though the interlinking of events of creation, production, movement, destruction, etc. Among its properties, there are some which intend or allow to interlink events, instantiating temporal relations (e.g. `P176 starts before the start of`), mereological relations (`P9 consists of`), causal relations (`P17 was motivated by`), and even include intentionality (`P20 had specific purpose`).

It is evident from the literature the necessity to represent, next to proper events, also some *state* or *condition*, lasting in time. This concept has been modeled as a sub-class of event [52] or as a completely separate class [31]. Several datasets for the detection of events and event relations are available, focusing mostly on temporal relations or on pure causality. Temporal relations have been largely investigated since 2009 in the TempEval shared task [118], which used the standard TimeML format and the TimeBank corpus [87]. The latter has been extended in

[1] http://motools.sourceforge.net/event

CausalTimeBank [76] that follows the {CAUSE, ENABLE, PREVENT} model. In addition, events are marked as *factual* (happened), *counterfactual* (not happened) or *non-factual* (possibilities), while their relation can be *certain* or *uncertain*. On top of TimeML, the EventStoryLine dataset is proposed in [16], and includes the representation of causes and consequences in the context of `PLOT LINKs`, for tagging events that are relevant in a plot.

EventKG is a KG of harmonised and interlinked events extracted from several resources, such as Wikidata and YAGO [43]. It includes over 1,3 million events, linked to their spatial and temporal coordinates. Only the connection between sub-events and super-events is represented in this dataset. For instance, it includes events such as *"Covid-19 lockdowns"* and *"Covid-19 pandemic in UK"*, with no direct relation between[2]. In the medical field, the datasets CSci [134] and EurekAlert [135] have been annotated according to four levels of causal relation: no relationship (c0), causal (c1), conditional causal (c2), and correlational (c3).

Table 7.4 summarises these models and datasets, showing which kind of relations are included in each of them. In addition to those, it is important to mention CausalNet, a common sense graph of actions, with weights between them indicating the likelihood that they are in a cause-effect relation [74]. Finally, it is worth to cite CausalBank – including 314 million sentence-level cause-effect pairs – from which it has been generated the Cause Effect Graph, in which links between events are weighted based on their co-occurrence in the text [68].

The table shows clearly that none of the existing resources is able to represent the entirety of the possible relations, calling for a more complete data model that harmonizes all of them.

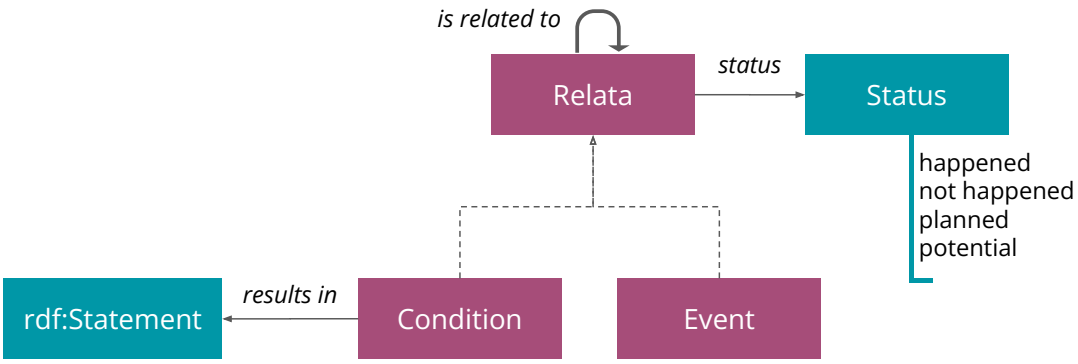

Figure 2.1: Core elements of the FARO ontology

	Fan [29]	Hong [44]	CIDOC CRM [28]	EventModelF [101]	Event Pattern [63]	SEM [119]	Wolff [130]	Event Ontology	TimeBank [118]	Causal TimeBank [76]	EventStoryLine [16]	EventKG [34]	CSci [134]	EurekAlert [135]	**FARO**
Temporal relations															
Before (after)	✓	✓	✓						✓	✓	✓				✓
Immediately Before / After									✓	✓	✓				✓
Equal / Simultaneous	✓	✓	✓						✓	✓	✓				✓
Meets (is met by)	✓	✓	✓						✓	✓	✓				✓
Overlaps / During	✓	✓	✓						✓	✓	✓				✓
Contains (is cont. by)	✓	✓	✓						✓	✓	✓				✓
Starts / Begins	✓	✓	✓						✓	✓	✓				✓
Finishes / Ends	✓	✓	✓						✓	✓	✓				✓
Vague		✓													
Mereological relations															
Sub-event (super-Event)		✓	✓	✓	✓	✓		✓				✓			✓
Re-emergence		✓													✓
Coreference		✓													✓
Variation		✓													✓
Confirmation / Ev. type		✓				✓									
Contingent relations															
Cause		✓	✓	✓			✓			✓	✓*		✓+	✓+	✓
Enable / Condition		✓					✓			✓-					✓
Prevent							✓			✓-					✓
Despite / Concession		✓					✓								✓
Correlation		✓		✓									✓	✓	✓
Intention / Purpose			✓												✓
Not cause													✓	✓	✓
Comparative relations															
Comparison		✓													✓
Conjunction / Similarity		✓													✓
Disjunction / Dissimilarity		✓													✓
Opposite		✓													✓
Negation / Alternative		✓													✓
Competition / Contrasting		✓													✓

Table 2.1: Event relation types supported by schemas/ontologies or present in datasets. In EventStoryLine (✓*), causal interpretation is possible through plot structure. CSci and EurekAlert (✓+) support conditional causation. Causal TimeBank has partial support (✓-).

2.2 FARO: Facts and Event Relations Ontology

FARO includes two different classes, *Condition* – transcendent, possibly can result in a RDF statement – and *Event* – immanent, following the categorisation in [100] – that are direct children of the more general class *Relata*, as in Figure 2.1. The latter is not intended to be directly use for instantiate entities, but is rather an abstraction layer for the other two main classes, allowing to define relations which connects indiscriminately any combination of them.

We found interesting to allow to define the *Status* of a *Relata* entity, to be chosen between four different options:

1. *happened* for sure at some moment in the past;
2. *not happened* for sure, we can exclude any happening of it in the future;
3. *potential*, meaning it is still uncertain if it will happen or not;
4. *planned*, sort of stronger potentiality, due to a will to this to happen.

This *Status* is intended to see an evolution in time, until it reaches either the *happened* or *not happened* status. We decided to leave possible to even define unforeseen statuses, apart to the four ones defined by the ontology.

Two *Relata* instances can be connected with a *is related to* property, which suggests general relatedness without further specification. The *is related to* property is further extended by 25 more specific properties, organised around four direct sub-classes of *is related to*. 2.2 illustrates the different classes for FARO:

Differently from other works, we decided to structure these properties hierarchically, in order to enable reasoning. This hierarchy has been realised following the definition of the individual relations. For the same purpose, we included logic constraints – such as `owl:cardinality` and `owl:propertyDisjointWith` – and further define property characteristics – using `owl:SymmetricProperty` and `owl:Transitive Property`. Please note that FARO is only intended to be used for representing the relationships between events, leaving the event description to be represented using other vocabularies or ontologies.

Looking a second time at Table 7.4, it is possible to appreciate that FARO is covering most of the listed relations, proposing itself as central ontology for the harmonisation of different data models. We decided to not include in our ontology the *Vague* temporal relation: even if valuable from the point of view of information extraction, these kind of properties are not

[2] We can logically imagine here that the spread the pandemic *caused* the lockdown, which is in its turn a measure for *preventing* the worsening of pandemic.

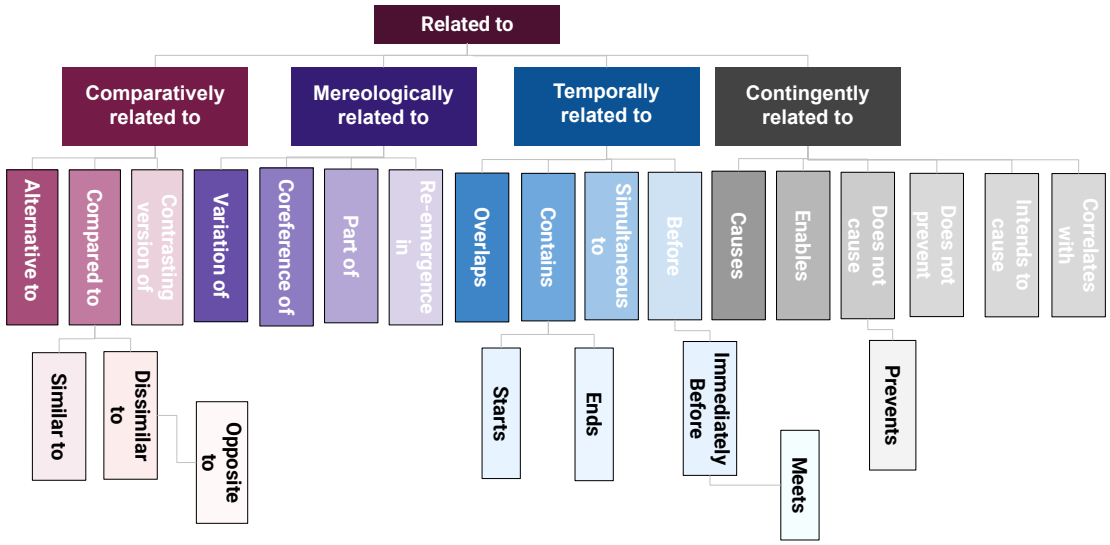

Figure 2.2: FARO Ontology Hierarchy

common in semantic web environments, where a more generic super-property can be used – in this case, *temporally related to*. Similarly, FARO is not including any *Confirmation / Event Type* property, because it can be expressed directly with an `rdf:type` statement. Alternatively, it is possible to use FARO in combination with other data models for event description – such as SEM [119], which allows typing events.

Not all event relationships involve just events. For instance, one may want to describe that being tall is helping a player to score in a basketball game. The player's height is of course not an event, but rather a *condition* which supported the happening of an *event*.

Relation between these events can go beyond vague causality to include more refined causal relations, we can categorize these relations under the super type *contingently related to* as described in 2.2. Although FARO covers 25 event relation types, harmonizing existing event-relation data models, we particularly focus here on relations beyond causality, as these are underrepresented in the literature. Such relations are essential for deeply understanding the semantics of event flows, which typically are only depicted with generalized or vague causality. In the following chapter, we will begin constructing datasets specifically designed around these underexplored relations.

Example direct-cause. "Ocean Drilling amp Exploration Co. will sell its contract-drilling business, and took a $50.9 million loss from discontinued operations in the third quarter **because** of the planned sale."

$$planned \xrightarrow{\textbf{causes}} sell\ ,\ planned \xrightarrow{\textbf{causes}} took\ ,\ planned \xrightarrow{\textbf{causes}} loss$$

Example intention. “Courtaulds PLC announced **plans** to spin off its textiles operations to existing shareholders in a restructuring to boost shareholder value."

$$spin \xrightarrow{\textbf{Intends to}} boost$$

Example prevention. “In addition to the estimated 45,000 Marines to ultimately be part of Operation Desert Shield, Stealth fighter planes and the aircraft carrier John F. Kennedy are also headed to Saudi Arabia to **protect** it from Iraqi expansionism."

$$headed \xrightarrow{\textbf{Prevents}} expansionism$$

Example enabling. “In addition, Courtaulds said the moves are logical because they will **allow** the textile businesses to focus more closely on core activities."

$$moves \xrightarrow{\textbf{Enables}} focus$$

Example does-not-cause. “He also rejected reports that his departure stemmed **from** disappointment the general manager’s post had not also led to a board directorship at the London-based news organization."

$$disappointment \xrightarrow{\textbf{NOT cause}} departure, post \xrightarrow{\textbf{NOT cause}} directorship$$

Example does-not-prevent. “Despite the heavy rain, the outdoor concert went ahead as scheduled."

$$rain \xrightarrow{\textbf{NOT prevent}} concert$$

Example correlates-with. “The increase in ice cream sales closely matched the rise in temperatures throughout the summer months."

$$ice\text{-}cream\ sales \xleftrightarrow{\textbf{correlates-with}} temperatures$$

The importance of these relations goes beyond a deeper understanding of event flows; it also enables the inference of further implicit relations. Figure 2.3 demonstrates a scenario where two relations co-occur within a single sentence. Specifically, it highlights how a single event can simultaneously fulfill two distinct roles—causing one event and preventing another.

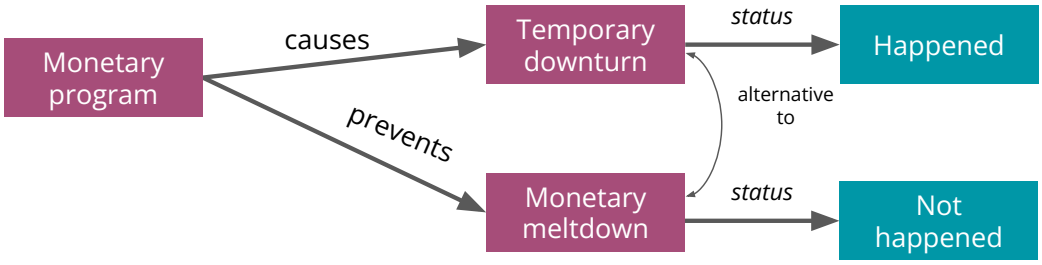

Figure 2.3: A relata causing an event and preventing another one, represented using FARO.

Consequently, we can infer that the two affected events represent alternative outcomes: one occurred, while the other did not, i.e., they can not happen at the same time.

example:

“A tight monetary program **caused** a temporary downturn but **prevented** a monetary meltdown". Figure 2.3 [3].

2.3 Conclusion

We introduce the *Facts and Events Relationship Ontology (FARO)*, a data model for representing events relationships in Knowledge Graphs. FARO has been implemented in OWL and publicly documented at https://purl.org/\gls{faro}/. In particular, we designed a structure which make possible to navigate through semantic links between events, exploring the flow of events backwards (searching for the causes or conditions of an event), forward (looking at consequences) or passing through other kind of connections. In other words, we want to make possible the creation of interconnected timelines of events, in which the connections between two consecutive points have explicit semantics. A such created graph would serve to improve the performance of downstream task (namely link prediction) and the explainability in decision making systems.

However two problems are left unsolved:

- We would to know what kind of questions is this model able to answer, which will be addressed in Chapter 3;
- The realisation of a graph using the FARO ontology needs a proper automation, that we will cover in Chapter 4 and Chapter 5.

[3]The text sample has been taken from https://economynext.com/sri-lanka-will-repay-bonds-holders-should-appreciate-efforts-made-cabraal-83785/. Last visited: 10/06/2022

Chapter 3

LLMs and Knowledge Engineering

Despite the growing adoption of ontologies in knowledge representation tasks, many widely used ontologies still lack formally defined Competency Questions (CQs). This is notably the case for the FARO ontology.

During the course of this thesis, LLMs have gained significant traction, drawing the attention of numerous researchers. Unlike the work presented in the previous chapter, we now shift our focus to the role of LLMs in knowledge engineering tasks—specifically, ontology modeling through CQ generation. Our objective is to assess whether LLMs are suitable for automatically generating meaningful and useful CQs that can guide ontology evaluation and use.

Through simple experiments, accessible to anyone via ChatGPT or similar tools, we observed that it is indeed possible to generate a basic OWL/RDF ontology structure using a prompt that briefly describes the targeted concepts. However, the reliability and scalability of this approach remain unexplored. This raises a critical question: To what extent can LLMs contribute to the knowledge engineering process alongside traditional methodologies such as CQ?

3.1 Related Work

The recent surge in generative AI and the widespread adoption of LLMs in industrial and consumer applications—particularly in tasks like generating code from natural language—suggests that abstracting a domain into a formal representation from textual corpora is an achievable goal that could assist knowledge engineers. The knowledge engineering and semantic web communities are increasingly exploring LLMs for ontology and KG construction, tackling tasks such as creating views on heterogeneous data lakes [9], RDF triple and SPARQL query generation [30], named entity recognition and relation extraction [125], RML mapping creation [42], and schema or ontology matching [3, 33, 41]. Consequently, various stages of the knowledge engineering process are being revisited in the era of LLMs (e.g., LOT [86]). However, the

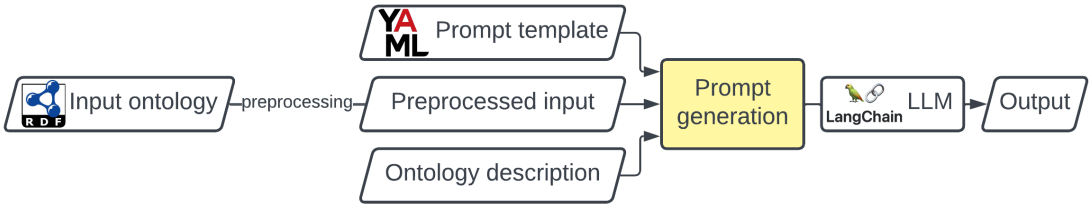

Figure 3.1: Workflow of the platform

systematic use of LLMs in these tasks requires further assessment, as results vary significantly depending on the underlying model and other influencing factors. Different works have so far investigated the performance of LLMs in classic tasks in the KG domain [4]. *SPIRES* [17] is a method that utilizes GPT-3 to produce structured data from an input text and schema. In [105], the authors use the Overall Execution Accuracy (OEA) to assess the performance of a LLM in converting questions to queries (SQL or SPARQL). The OEA is computed on an ad-hoc benchmark, where an execution is considered accurate if the query result matches the corresponding answer.

Several works address the usage and production of CQs. The study of patterns in competency questions [128] has inspired the realization of AgOCQs [8] in which CQs are automatically generated. The evaluation has been performed with an expert group, which highlighted the validity of the method. The patterns can be filled by *Glossary of terms* – which can be automatically extracted such as in ReqTagger [129] – or used to automatically generate SPARQL queries from CQs [12, 127].

3.2 Methodology

In this section, we provide details of our approach by focusing on the LLM-based data processing pipeline (Section 3.2.1) and on the prompt details (Section 3.2.2).

3.2.1 Implementation

To standardize and automate experiments, we developed a platform in Python, whose workflow is depicted in Figure 3.1. The platform relies on the LangChain framework[1] [19] to interact with various LLMs. Specifically, we integrated models from LangChain providers for Ollama, HuggingFace, and OpenAI into our workflow, allowing for querying within the same pipeline.

We make use of a prompt configuration in the form of a YAML file, including:

- the description of the task TD (for documentation purposes);

[1] https://python.langchain.com/

- the list of required input fields;
- the prompt template, in which placeholders are marked by curly brackets as in the documentation of LangChain, e.g. `{name}`, `{classes}`.

Additionally, each process can be further customized by specifying the LLM to use, the path of the input ontology, whether to include the ontology description in the prompt or not, and the number of required output results.

In order to avoid to ingest the full RDF representation in the prompt[2], we parse the ontology using RDFlib [62] and extract either:

- the list of class labels C;
- the list of property labels P;
- a summary schema of the interconnection of classes and properties S.

This schema S is represented as triples in the format (C_x, p_y, C_z), where $C_x, C_z \in C$ are class labels, and $p_y \in P$ is the label of an object property which has C_x as domain and C_z as range. An example taken from the FOAF ontology is `(foaf:Group, foaf:member, foaf:Agent)`. Please note that C_x and C_z are not necessarily two different classes, because the domain and range can coincide, e.g. in

`(foaf:Person, foaf:knows, foaf:Person)`. In the case of a data property $p_d \in P$, we include the triple $(C_x, p_d, \text{“}literal\text{"})$, e.g. in `(foaf:Person, foaf:lastName, "literal")`.

When the dimension of the ontology is large, it is processed in batches of 20 classes. In such a case, in each iteration, C is composed of a maximum of 20 classes, P includes all properties which have C as domain or range, and S encompasses all interconnections involving C and P.

3.2.2 Prompting

We primarily utilized three templates for our work. The first template outlines the classes within the ontology, the second includes both classes and properties, and the final template integrates the ontology's schema. Each of these templates encompasses:

- Task Description (TD): ‘`Generate a set of competency questions (CQ) which are relevant for the ontology called {name of ontology}`’.

[2]During some preliminary experiments, we realised that including the full ontology in Turtle format was producing a long prompt, which has shown to confuse the LLMs and produce hallucination.

- Ontology Description (*OD*): provides a general overview of the ontology and specifies the domain it belongs to, e.g., `'Odeuropa ontology represents' odours and their experiences from Cultural Heritage perspective.`
- Examples (*EXP*): examples of competency questions of the desired ontology, e.g., `'Which scents were linked to the idea of heaven in X period?'.`
- Notes(*N*): guidelines provided to the model for brevity and clarity, e.g., `'Do not include any text except the competency question'.`

Based on the prompt configuration technique described in Section 3.2.1, we propose to generate prompts for a given ontology with various features (Table 3.1) depending on the overall experiment goals and following best practice in prompt structuring.

Table 3.1: Prompt features as a function of the evaluation goal.
For the classes feature, the "*The {name} ontology has the following set of classes:*" is used in the prompt. For the "Properties" feature, it is the "*and the following set of properties:*" sentence. For "Schema" it is the "*The {name} ontology has the following schema*" sentence. "opt." stands for optional (i.e. w. and w.o definition).

Evaluation goal	Definition	Classes	Properties	Schema	Examples	Constraints
All classes	opt.	✓			*n*	✓
All classes + properties	opt.	✓	✓		*n*	✓
Logic	opt.			✓	*n*	✓

3.3 Experiments

In this section, we present the experiments conducted based on the method described in Section 3.2.

We first provide details of the dataset used in Section 3.5.1, then on the LLMs used in Section 3.3.2, and finally report on the evaluation results in Section 3.3.3.

3.3.1 Investigated Ontologies

For our experiments, we selected a subset of five ontologies (Table 3.2) with a publicly available implementation based on the following two criteria: *1)* these ontologies were modeled following explicitly the CQs methodology [95]; *2)* these ontologies have well-phrased CQs with associated Authoring Tests (ATs) in the form of SPARQL queries.

Candidate ontologies. The following ontologies fulfill the selection criteria discussed above:

- **DOREMUS** [2]: related to **music** and **cultural heritage** domains, the ontology comes with a documentation, competency questions, SPARQL queries and APIs and a large knowledge graph.

- **Polifonia** [25]: related to **music** and **cultural heritage**, the ontology enables to capture musical and historical knowledge. In addition to CQs and a comprehensive documentation, the authors provides a set of queries that we can leverage as ground truth and a knowledge graph.

- **DemCare** [59]: related to the **medical** domain, specifically tailored for dementia care and monitoring. Provided with CQs, a dataset, and well-structured documentation.

- **Odeuropa** [70]: related to **sensory experiences** and **cultural heritage**, focusing on olfactory experiences in historical contexts. Provided with CQs, documentation, queries and a dataset.

- **NORIA-O** [111]: related to the **IT** domain, and designed for network monitoring and performing **anomaly detection**. Provided with CQs, a documentation, queries, and a knowledge graph.

Once the subset was established, we created a dataset by recording a versioned copy of the ontologies' implementation, as well as their companion set of CQs and ATs. To generalize the approach described in Section 3.2 to all the ontologies of the subset, we normalized the representation of the CQs by storing them in a YAML data structure including – if relevant – the reference to the corresponding ATs. The dataset is publicly available in our repository, with annotation on the origin for each component of it and explanations on the normalization process.

Table 3.2: Subset of ontologies for the LLM4KE experiments.
Ontologies in our dataset, along with additional details such as the number of classes (#Classes) and properties (#Props), associated competency questions (CQ count), associated authoring tests (AT count), and a coverage measure (AT/CQ coverage) indicating the extent to which ATs are effectively defined and implemented for each CQ. For Polifonia, we count CQs from their "default group" and indicate "?" for the AT count as no obvious set of ATs was found. For Demcare, the CQ2SPARQLOWL [85] dataset served as a reference for building our dataset. For the remaining ontologies, the dataset was directly constructed from each project's repository.

Data-model	Ref.	Full ontology name or topic	#Classes	#Props	CQ count	AT count	AT/CQ coverage
DemCare	[59]	Dementia Ambient Care Ontology.	290	115	107	60	56%
DOREMUS	[2]	Music catalogues on the web of data.	218	705	58	30	52%
NORIA-O	[111]	IT networks and operations for anomaly detection and IT service management.	55	135	26	25	88%
Odeuropa	[70]	Odours and their experiences from a Cultural Heritage perspective.	13	10	74	74	100%
Polifonia	[25]	Polifonia Ontology Network (PON) for queries in the music domain.	247	299	194	?	0%

3.3.2 Investigated LLMs

We explored various LLMs options, including both open-source and proprietary models. For open-source models, we considered their performances according to the Hugging Face leaderboard,[3] in particular across three specific datasets, which we consider relevant for this research:

- ARC2018 [23] (AI2 Reasoning Challenge), a question-answering dataset;
- HellaSwag [136], created to challenge model common sense reasoning abilities;
- Winogrande [98], a dataset designed to evaluate commonsense reasoning capabilities in AI systems.

We selected these models based on their architectures, aiming to choose one from each architectural category. Each model was chosen for its superior performance within its respective architecture, as indicated by their positions on the leaderboard at the time of selection. Due to resource limitations, we have opted to confine our selection of open-source LLMs to those with a parameter count equal to or less than 13 billion. Table 3.3 summarises the used LLMs.

Table 3.3: Used LLMs for Experiments.
B refers to billion parameters.

Model	Architecture	Size (B)	Access Paradigm
DPO[4]	MixtralForCausalLM	12.9	Open-source
Solar[6]	LlamaForCausalLM	10.7	Open-source
UNA[8]	MistralForCausalLM	7	Open-source
Zephyrβ [10]	MistralForCausalLM	7	Open-source
GPT 3.5	Transformer Decoder	175	proprietary
GPT-4-0125-preview	Transformer Decoder	1500	proprietary

We have used `Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B`[4] (we refer to it as *DPO*), which is an instance of `FusionNet_7Bx2_MoE_14B` fine-tuned on the Truthy-DPO dataset[5].

Additionally, we leveraged `SOLAR-10B-OrcaDPO-Jawade`, which we shortcut to *Solar*, a fine-tuned version of `SOLAR-10.7B-Instruct-v1.0`[6] [57], finetuned on the `dpo pairs` dataset.[7] Furthermore, we have used `UNA-TheBeagle-7b-v1`[8], that we call simply *UNA*, a 7B LLM trained on The Bagel dataset.[9] On the other hand, we opted for zephyrβ[10] [117], because of its performance that surpassed Llama2 70B [115] on different benchmarks.

[3] https://huggingface.co/spaces/HuggingFaceH4/open_\gls{llm}_leaderboard
[4] https://huggingface.co/yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B
[5] https://huggingface.co/datasets/jondurbin/truthy-dpo-v0.1
[6] https://huggingface.co/bhavinjawade/SOLAR/-10B/-OrcaDPO/-Jawade
[7] https://huggingface.co/datasets/Intel/orca_dpo_pairs
[8] https://huggingface.co/fblgit/UNA-TheBeagle/-7b/-v1
[9] https://huggingface.co/datasets/jondurbin/bagel-v0.3
[10] https://huggingface.co/HuggingFaceH4/zephyr/-7b/-beta

Moreover, we included in our study API-only access models, and in particular the GPT series from OpenAI[11]. We used both GPT3.5[12] and GPT4 [81].

3.3.3 Results

To perform the evaluation of our approach, we utilize the dataset presented in Section 3.5.1 and consider the CQs provided by the authors of each ontology as the ground truth. We compare the output CQs from the LLMs (CQ_o) to each CQ in the ground-truth (CQ_{gt}) and consider a CQ_o as valid if it is sufficiently similar to at least one CQ_{gt}. For the similarity score, we use cosine similarity between the embeddings of CQ_o and CQ_{gt} computed using SentenceBERT [94]. We define a threshold θ above which we consider a CQ_o valid (Eq. 3.1):

$$x \in CQ_o^{valid} \Leftrightarrow x \in CQ_o \wedge \exists \{y \in CQ_{gt} : \text{cosine similarity}(y, x) > \theta\} \tag{3.1}$$

with $CQ_o^{valid} \subset CQ_o$. We then compute the precision $P = \frac{\text{number of } CQ_o^{valid}}{\text{number of } CQ_o}$ of each experiment.

The results of the experiments are reported in Table 3.4, using a threshold of $\theta = 0.6$, chosen empirically for better showing the differences between the models. As a first outlook, we observe that the precision scores are generally low. From the perspective of the LLMs, Zephyr consistently shows the best scores across a majority of ontologies with at least two different modalities, with the exception of some experiments on Odeuropa (in particular with only classes) and NORIA-O (classes and properties) where UNA performs better. For Odeuropa, this can be due to the fact that the dimension of Odeuropa is lower that the used batch size, and it is consequently included entirely in the prompt; reducing the batch size to 5, improves the results of Zephyr for Odeuropa to 0.90 (C), 0.91 (P) and 0.70 (S). Future work will investigate the effect of the batch size on the different LLMs and ontologies.

From the perspective of prompt features, we observe that providing examples (few-shot) generally leads to better precision (compared to zero-shot), although not always.

Even though the absolute scores are generally quite low, it should not be concluded that the generated CQs are irrelevant. In fact, the generation process may have resulted in new competency questions that can be a valuable addition to the ground truth dataset. To properly evaluate the relevance of these competency questions, an expert panel should be involved, which will be the focus of future work. Due to variations in the number of classes among the ontologies in our dataset (Table 3.2), it is important to note that the LLMs used in the experiments may have been queried more frequently for certain ontologies and less frequently for others, because of the subdivision in batches.

[11] https://openai.com/
[12] https://platform.openai.com/docs/models/gpt-3-5-turbo

Table 3.4: The precision scores for the experiments, reporting the LLM name, the number of included exemplary CQs and, for each ontology, the modality {C = all classes, P = classes and properties, S = summary schema}

Ontology	→	DOREMUS			DemCare			Odeuropa			Polifonia			NORIA-O		
LLM	**Ex**	C	P	S	C	P	S	C	P	S	C	P	S	C	P	S
GPT3	0	0.02	0.01	0.01	0.15	0.14	0.00	0.00	0.00	0.10	0.08	0.08	0.20	0.00	0.00	0.03
	3	0.04	0.01	0.04	0.17	0.13	0.00	0.90	0.30	0.00	0.20	0.30	0.32	0.00	0.03	0.03
GPT4	0	0.00	0.00	0.02	0.14	0.23	0.01	0.20	0.50	0.30	0.21	0.24	0.30	0.00	0.03	0.00
	3	**0.10**	0.11	0.11	**0.21**	0.17	0.01	0.40	0.90	0.90	0.32	0.32	0.32	0.03	0.03	0.00
dpo	0	0.00	0.00	0.00	0.04	0.08	0.00	0.70	0.30	0.00	0.05	0.09	0.11	0.00	0.00	0.00
	3	0.03	0.04	0.01	0.15	0.13	0.04	0.75	0.82	**1.00**	0.22	0.22	0.22	0.04	0.06	0.00
solar	0	0.00	0.00	0.00	0.08	0.06	0.00	0.20	0.00	0.20	0.07	0.04	0.12	0.00	0.03	0.00
	3	0.00	0.12	0.07	0.11	0.17	0.00	0.30	0.30	0.30	0.20	0.22	0.24	0.04	0.00	0.03
una	0	0.00	0.03	0.05	0.10	0.10	0.00	0.50	0.00	0.64	0.08	0.05	0.10	0.03	0.00	0.00
	3	0.09	0.15	0.12	0.20	0.24	**0.27**	**1.00**	0.70	**1.00**	0.34	**0.38**	0.33	**0.31**	**0.07**	0.00
zephyr	0	0.01	0.01	0.00	0.05	0.09	0.00	0.90	**1.00**	0.00	0.16	0.08	0.15	0.00	0.00	0.00
	3	0.03	**0.58**	**0.56**	**0.21**	**0.33**	0.00	0.40	0.00	**1.00**	**0.36**	**0.38**	**0.34**	0.00	0.00	**0.20**

A first qualitative assessment let us notice that the configurations obtaining the lower scores have some common characteristics: the strict reuse of class and property labels instead of periphrasis, the inclusion of the ontology name in the output CQ, the presence of generic connections between concepts (“involve”, “influence”, “associate”, “relate”) instead of semantically meaningful ones. Future work will investigate possible patterns with the help of domain experts.

3.4 Application of LLMs to generate CQs for FARO ontology

Having demonstrated that LLMs, despite certain limitations, can achieve good precision in generating CQs, particularly in few-shot settings. we decided to explore their ability to generate CQs for the *FARO* ontology.

Our evaluation began with a zero-shot setup, followed by a few-shot scenario in which we provided the model with three example CQs specifically tailored to *FARO*. These examples aim to reflect the types of event and condition relationships captured by the ontology:

- Which events were made possible or enabled by the outcome of the Brexit vote?
- Which events were considered alternatives to the UK exiting the EU?
- Which events were prevented from happening due to the lockdown?

In our experiments on FARO, we employ *Zephyr*, as it demonstrated superior performance in our previous evaluations. Additionally, while earlier experiments utilized an older version of GPT-4, we now include tests with the updated GPT-4o model to assess potential improvements in performance brought by this newer version. Below, we present a qualitative assessment of the generated CQs, focusing on their quality, specificity, and domain diversity.

Zero-Shot Prompting

Across all zero-shot scenarios, we observe that the LLMs tend to generate very generic CQs. For instance, GPT-4 produced the following example:

> *Which statement results from a given condition?*

This pattern of overly broad and vague questions was consistent across all tested models, indicating that without additional context or guidance, LLMs struggle to align generated CQs with the specific semantics of the FARO ontology.

few-shot - Classes Only

When prompting with ontology classes alone, none of the models produced high-quality or meaningful CQs. The output remained vague and lacked alignment with the structure or purpose of FARO. Examples include:

> *How are relata involved in the transition from one event to another?*
> *What conditions must be met for a statement to be considered true?*
> *Which statuses are most commonly reported in relation to a significant event?*

few-shot - Summary Schema, Classes and Properties

Introducing a logical scheme or the properties together with classes into the prompt substantially improved the relevance and coherence of generated CQs.

For instance, GPT-4 produced accurate questions, although with limited topical diversity. Most of the generated CQs centered around the example it was given—Brexit:

> *Which events resulted in the literal statement of post-Brexit economic uncertainty?*
> *Which events intentionally cause political disruption after the Brexit vote?*
> *Which events do not cause the expected public backlash in the Brexit scenario?*
> *Which conditions result in statements about the UK's exit from the EU?*

The same behaviour was seen with Zephyr: with logical scheme summary prompting or the inclusion of properties with classes, it showed more structure but lacked topical diversity and often repeated question patterns. Many questions were highly influenced by the provided examples.

On the other hand, GPT-4o demonstrated noticeable improvement in both accuracy and domain diversity. It generated well-formed and meaningful questions across a range of topics:

Which conditions result in the declaration of a national emergency?
What statements are the outcomes of severe weather conditions?
Which events intentionally aim to cause economic growth?
What events are temporally related to a global pandemic?

Which conditions re-emerge in cyclic economic patterns?
What relata are related to the technological advancements in the 21st century?
Which events are temporally related to the rise of social media platforms?
What statements were intended to cause a change in consumer behavior?

Overall, these experiments highlight the importance of context and example design in guiding LLMs to produce meaningful and ontology-aligned CQs. While zero-shot setups consistently yield generic outputs, using a scheme summary prompting and including booth classes and properties significantly improves both relevance and coherence. Among the tested models, GPT-4o showed the best balance between accuracy and topical diversity.

3.5 Towards a More Comprehensive Benchmark

Addressing the gap in assessing LLM performance for reverse engineering an ontology, it is essential to evaluate their effectiveness across other subtasks to gain a comprehensive understanding of their capabilities and limitations in knowledge engineering. This highlights the need for a deeper investigation into the characteristics of ontologies that impact the accuracy of LLM-generated responses and vice versa.

In this section, we further detail the elements of this benchmark: the use cases and tasks to be evaluated (Section 3.5.2), the curated datasets (Section 3.5.1), and the evaluation metrics for improving LLM performance (Section 3.5.3).

3.5.1 Benchmark Data

In addition to the availability of CQs that we discussed before for selecting the ontologies, we propose to consider the following criteria to include ontologies in the benchmark:, comprehensive documentation of the ontology, FAIR-related scores attached to the ontology, SPARQL queries demonstrating the usage of the ontology and/or SHACL [58] shapes constraining its usage. Furthermore, we advocate for a selection that covers diverse domain of discourses. From a conceptualization point of view, the selected ontologies should exhibit different structure (e.g. relatively flat ontologies such as schema.org[13] versus nested models such as FRBR [47])

[13] https://schema.org/

and make use of well-known ontology design patterns (e.g. event-based modeling like CIDOC-CRM[14]). Although not all ontologies provide the requisite information for every criteria, they should still be suitable for evaluation in specific tasks.

Domain of Discourse. The benchmark should cover as many domains as possible to be able to draw the line between the performance of LLMs and a given domain. Examples include: general purpose ontologies, IT and sensors, creative industries and medias, cultural heritage and museums, healthcare and medicine, biology and life sciences, education and e-learning, e-commerce and retail, finance and banking, and legal sectors.

We plan to incorporate additional ontologies from other domains using well-known catalogs such as Linked Open Vocabulary (LOV) [110] and the Industry Portal [6]. We can also rely on [85] which provides a dataset of competency questions for different ontologies and domains (e.g. the African wildlife ontology (AWO) [55] , the Software Ontology (SWO) [48], the Generic Ontology of Datatypes (OntoDT) [82]) together with their translation into SPARQL queries.

3.5.2 Benchmark Tasks

Conceptualize an ontology. This task aims to conceptualize an ontology given a set of CQs and a domain scope. A variant of it can start from a partial ontology, and the task of the LLM is to complete the ontology by adding the missing classes and properties. In this case, we will evaluate the LLM's performance in accurately completing an ontology based solely on the upper-level structure. This will help understanding the depth to which the LLM can reach with and without CQs that could assist in clarifying user intent.

Generate Competency Questions. This task focuses on generating CQs given an ontology or specific parts of it, similar to the approach outlined in [93]. The performance of LLMs will be evaluated according to the prompts being used and the nature of the information provided, such as a general description of an ontology, taxonomical branches, or even the entire ontology. The impact of each type of input will be analyzed in order to assess the generalization capabilities of the model and its limitations related to the domain, the structure or the size of the ontology. Additionally, this task aims to identify best practices for prompt formatting to enhance communication between human language and LLMs.

Produce the ontology documentation. In this task, the LLM should produce a human-readable documentation of the ontology, emphasizing on its main classes and properties. This can be further expanded in generating useful API calls, e.g. following the SPARQL Transformers

[14] https://cidoc-crm.org/

approach [69]. This can be achieved by either inputting the entire ontology or using a chain-of-thought (CoT) [126] approach, in case of complex ontologies.

Implement queries corresponding to CQs. This task will assess the LLM's ability to generate autonomously relevant queries given the ontology structure and the user intent expressed with competency questions.

Verbalize Knowledge Graph Excerpts. The task involves generating human-readable interpretations of a dataset, using an ontology as a guide for the LLM to structure the information. The goal is to go beyond the verbalization of all possibles triples (subject, verb, object) and to generate paragraphs that summarize the graphs.

3.5.3 Evaluation Metrics and Process

In this section, we outline the process for evaluating and improving LLM performance using a factorial experiment design based on the following factors:

1) *Prompting strategies:* presence or absence of partial knowledge of the competency questions, taxonomy, and documentation in the LLM's context, depending on the specific task similar to an ablation study.

2) *Data instance:* presence or absence of instances from a knowledge graph structured by a given ontology to guide the LLM.

The evaluation process will be iterative, refining the integration of the knowledge graph with the LLM at each step and assessing performance improvements along the way. Multiple iterations will be conducted and the results will be analyzed using statistical methods to quantify progress. To facilitate comparison between different models or methods, we propose to rely on a CI/CD-enabled pipeline based on the tools developed in [93], with performance results tracked using a leaderboard.

Table 3.5 summarizes the evaluation techniques intended for each of the benchmark tasks (Section 3.5.2).

- **Semantic Similarity.** This is typically implemented as a cosine similarity between vectors embedding a ground truth sentence and a generated response from a LLM. SentenceBERT is generally used for evaluating the CQ generation and ontology documentation tasks.

Task	Evaluation Techniques
Conceptualize an ontology	Ontology Evaluation Criteria, Logical Consistency.
Generate Competency Questions	Semantic Similarity.
Produce the ontology documentation	Semantic Similarity, between the generated documentation and the existing definition of classes and properties).
Implement queries corresponding to CQs	Structure Comparison.
Verbalize Knowledge Graph Excerpts	Fluency and Coherence.

Table 3.5: Benchmark Tasks and Evaluation Techniques

- **Ontology Evaluation Criteria.** Using an existing ontology as the gold standard, we can assess the *accuracy, completeness, and conciseness* of the generated ontology [88]. This serves the tasks of ontology generation and ontology enrichment. However, *adaptability, clarity, and computational efficiency* are not addressed in this research, as they depend on the ground truth ontology.

- **Logical Consistency.** This enables us to validate the semantic formalization of an ontology, typically using tools such as the Hermit reasoner [107].

- **Structure Comparison.** When evaluating the generation of queries, this measure will compare the structure of the generated query with a ground truth query. We can leverage the RTED algorithm, which calculates the Tree Edit Distance (TED) [83] for this purpose.

- **Fluency and Coherence.** When verbalizing and summarizing KG excerpts (instance data guided by an ontology), this metric will assess the fluency (grammatical correctness) and adequacy (referring to the accurate integration of triples [53]) of the generated text.

3.6 Conclusion

To explore what types of questions the FARO ontology can answer, we first evaluated the ability of LLMs to generate competency questions across a range of ontologies. This formed part of a broader investigation into how LLMs can support knowledge engineering tasks, particularly in the context of ontology development.

We designed a methodology and implemented a data processing pipeline involving six LLMs, four prompting strategies (zero-shot, few-shot with classes,few-shot with classes and properties, and few-shot with schema summaries), and five diverse ontologies. These initial experiments helped us identify effective prompting conditions, highlighting the importance of including examples and relationship information in prompts to improve the quality of generated CQs. Interestingly, we also observed that overly detailed prompts could reduce

model performance in some cases, suggesting a need for balanced prompt design.

From these results, we selected the most promising models—Zephyr and GPT-4—for further testing on the FARO ontology. We also included GPT-4o, which was not available during the initial experiments, to assess its performance.

Qualitative analysis unveiled that while zero-shot and class-only prompts generally produced vague or poorly aligned questions, schema-informed prompting led to significantly better results. In particular, GPT-4o demonstrated the strongest performance, generating accurate, semantically coherent, and topically diverse CQs. Examples include: “Which events intentionally aim to cause economic growth?” and “What statements were intended to cause a change in consumer behavior?”

Encouraged by these findings, we extended our work beyond CQs generation and proposed a benchmark to evaluate the ability of LLMs in a broader range of knowledge engineering tasks. This benchmark aims to systematically assess how LLMs can contribute across different stages of ontology-based system development, from data modeling to application-oriented reasoning.

Part II

Natural Language Processing of Event Relations

Chapter 4

Constructing an Event Relation Dataset

The construction of an event relation dataset is a crucial step in advancing automatic event relation extraction methods, particularly for fine-grained causal reasoning. This chapter introduces the construction of a dataset centered on a subset of event relations from the FARO ontology—namely *Cause*, *Intend*, *Prevent*, *Enable*, and *Not Cause*. These relations were selected due to their limited representation in existing literature and their critical role in capturing nuanced semantic distinctions beyond simple causality. Given the limitations in existing datasets, we extend and re-annotate two well-established resources, TimeBank and EventCausality by incorporating additional event relation types, ensuring compatibility with the TimeML annotation framework. To further enhance data coverage and balance, we leverage LLMs for data augmentation and integrate commonsense knowledge from sources such as The Atlas of Machine Commonsense (ATOMIC) [99]. The resulting dataset serves as a foundation for training and evaluating models in event relation extraction, with applications in causal reasoning, narrative generation, and fact-checking. This chapter details the dataset construction process, annotation methodology, augmentation strategies, and the final structured representation as a knowledge graph.

4.1 Initial Dataset Construction

In this section, we describe a dataset that includes some of the relations described in FARO, focusing on the contingent relations. we believe that a first version of a multi-relation event dataset is crucial to start designing new automatic methods for extracting them. Note that this is the first dataset incorporating *Intend*, and differentiating between *Cause*, *Prevent*, and *Enable*. These relations have been chosen because poorly represented in available datasets.

We developed this dataset by extending and re-annotating two existing datasets with new event relations types, namely intention, enabling, prevention, and explicit negation of causality. The choice of the datasets were based on their format (TimeML), which was convenient for

extending it with other relation link.

- TimeBank [122], published by Brandeis University, providing 183 English news articles with over 27,000 event and temporal annotations about events, times and temporal links between events and times. The annotation respects the TimeML 1.2.1 specification.
- EventCausality [80], the dataset comes with causal and temporal annotations on 25 news articles collected from CNN7, giving at the end 1.3k events, 3.4k temporal links and 172 causal relations between events.

Both selected datasets are represented using the TimeML format [87], which we kept it as a base. This format enables to annotate events in the text and to declare possible connections between them using one among:

- `TLINK`, a temporal relation between events (or between an event and a time expression). Ex: "John **left (ei1)** 2 days *before (s1)* the **attack (ei2)**" →

```
<TLINK eventInstanceID="ei1" signalID="s1"
relatedToEvent="ei2" relType="BEFORE" magnitude="t1" />
```

- `ALINK`, a relationship between an "aspectual" event (events that add a notion about an action whether it begins, finishes, continues, etc.) – normally represented by phrasal verbs, e.g. *start to*– and its argument event: initiation, continuation, etc. Ex: John **started (ei5)** to **read (ei6)** →

```
<ALINK eventInstanceID="ei5"
relatedToEventInstance="ei6" relType="INITIATES" />
```

- `SLINK`, refers as a Subordination Link, which is used for contexts introducing relations between two events, or an event and a signal. Ex. "John **said (ei2)** that he **taught (ei3)** on Monday." →

```
<SLINK   eventInstanceID="ei2"
subordinatedEventInstance="ei3" relType="EVIDENTIAL" />
```

While TimeBank uses all 3 types of links, EventCausality instantiates explicit TLINK relation tags, with causal links are represented separately in another file – not following TimeML, so hard to re-use in other dataset. We kept the temporal links and we enriched it by new event relation tags.

4.1.1 A generic relation link: RLINK

Following the experience described in [76] with the addition of the causal link CLINK, we extended TimeML with a new relation type RLINK, which we designed as a generalisation of the existing ones (TLINK, ALINK, CLINK), and enriched the previously described datasets accordingly. RLINK – or relation link – is a description of a generic relationship between two events, that can be further specified. A RLINK instance has 4 attributes as following:

- Link Identifier (lid) represents an ID for the relation, unique at the document level;
- Relation type (relType) refers as the type of relation between two events or the predicate of the triple, which can be one of the property of FARO, e.g. *Cause*, *Prevent*, etc.;
- Event instance Identifier (eventInstanceID) is the relata with the role of subject of the triple;
- Related event instance Identifier (relatedEventInstance) is the relata with the role of object of the triple.

Example. "Subcontractors will be offered a **settlement (ei264)** and a swift **transition (ei265)** to new management is expected to avert an **exodus(ei268)** of skilled workers from Waertsilae Marine's two big shipyards."

```
<RLINK eventInstanceID="ei264"
lid="l42" relType="prevention" relatedEventInstance="ei268" />

<RLINK eventInstanceID="ei265"
lid="l43" relType="prevention" relatedEventInstance="ei268" />
```

4.1.2 Candidate Generation

We re-annotated each of the mentioned datasets applying a semi-automatic procedure, based on expression matching as first step, followed by a manual check to validate the extracted annotations.

First, we collected a set of potential signal words for each of the 5 studied relations. We searched in the text these signals and extracted the sentences containing them, which we consider potential candidates. Each candidate sentence is dispatched according to the number of possible event pair combinations of relata that can construct the relation, among all the already annotated events for that specific sentence in the original datasets. In other words, we

created a table in which each line contains a unique combination of two events, the signal word, the document id and the full sentence, as in Table 4.1.

Event1	eid1	Event2	eid2	signal	Annotation	DocumentID	Sentence
settlement	e44	expected	e14	avert	0	$wsj_0187.tml$	Subcontractors will ...
...	...	...	...	...	0	$wsj_0187.tml$	Subcontractors will ...
settlement	e44	exodus	e46	avert	1	$wsj_0187.tml$	Subcontractors will ...
...	...	...	...	...	0	$wsj_0187.tml$	Subcontractors will ...
transition	e45	exodus	e46	avert	1	$wsj_0187.tml$	Subcontractors will ...

Table 4.1: Table of the candidate pairs for a specific relation type (prevention), with manual annotation (1 = correct, 0 = wrong).

In the following, we detail the strategy applied for the signal collection and the extraction for each relation type, together with some examples.

Causality.

We adopted the manually defined causal signals and causal verbs in [77], in which causal signals are nominal phrases that express causality (e.g: because of, in order to, as a result of). However, causal verbs are a set of verbs representing the act of causing, such as: cause, bribe, push, etc. The first automatic selection results in 1790 candidate causal relation for TimeBank dataset, and 697 for EventCausality dataset. After dispatching, we ended up with 9658 and 1205 possible event pair causal relation for TimeBank and EventCausality datasets respectively.

Intention.

To capture intention, we manually created a list of possible intention signals (e.g: want, plan, aim). Additionally, we adopted another set of events as signals taken from the TimeBank dataset belonging to the class I-action. I-action (Intentional action), is an argument for those events that express an action of intention to do something.

As a result of automatic intention signals matching, we got 412 candidate expression for holding intention for TimeBank and 154 for EventCausality dataset. However, after extracting all possible event pair combinations, we ended up with 4028 and 230 intention candidate expression for TimeBank and EventCausality datasets respectively.

Prevention.

We integrate prevention signals as defined in [77], in which are initially included into the causal verbs list and claimed to express prevention, e.g: block, bar, deter, etc. After the exploitation of these signals, we could extract 120 and 25 candidate expression, which lead to 988 and 53 event pair combination from TimeBank and EventCausality respectively.

Enabling.

For this event relation, we defined a list of verbs that alert the existence of enabling, such as authorize, warrant, entitle, etc.. We extended this list with enable signals as defined in [77], e.g: help, permit, empower, etc. to guarantee a high coverage. As a result, we obtained 41 and 17 candidate expression and 328 and 16 candidate event pairs combination for TimeBank and EventCausality datasets respectively.

Not Causality.

To extract the explicit not cause relation, we rely on the previously extracted causal relations, in which, we first naively pick those expression having both negation and causality at the same time, than manually validate the right ones. Consequently, we obtained 230 and 124 candidate expression and 1640 and 255 candidate event pairs from TimeBank and EventCausality datasets respectively.

4.1.3 Manual Assessment

The described process extracted a long list of candidate relations, most of them being incorrect and to be filtered out. The structure in Table 4.1 has been then used by two fluent English speakers annotators, which manually checked the candidate sentences. The process is summarised in the following steps:

1. Each annotator reads and annotates 300 lines for each type of relation.
2. On this preliminary annotation, we compute Cohen's kappa inter annotator agreement (IAA) [56] between the two annotations.
    - If the IIA does not show a substantial agreement (> 0.6), the annotators meet, check the contrasting annotations and agree on a strategy. Then, 300 different lines are chosen and the process goes back to point 1.
    - Otherwise, we progress to next point.
3. The annotation is completed for the rest of the datasets, each annotator taking a unique

portion.

During annotation, only relations with precised relata have been considered as correct, while others have been marked as not correct. The annotation process relied on an IAA = 0.7112, which is considered a substantial agreement. Table 4.2 shows the initial stats of our annotated dataset.

In the following example, the signal word is marked in **bold**, events have been marked using *italic*, but only the underlined ones have been considered part of relationships of type *Prevent* by the annotators in Table 4.1.

Example. "Subcontractors will be offered a *settlement* and a swift *transition* to new management is *expected* to **avert** an *exodus* of skilled workers from Waertsilae Marine's two big shipyards, government officials *said*."

Relation type	Cause	Intend	Prevent	Enable	Not-Cause
Number of relations	283	44	13	18	3

Table 4.2: Total number of relations validated by annotators for each relation type. These relations are present in the released Event Relation dataset.

Noticing the imbalance in the initial dataset, we decided to conduct additional annotations. To address this, we utilized a large corpus of English-language newswire texts from the French news agency AFP [1], comprising over 2 million articles. Our goal was to extract missing sentences for the under-represented event relations using the signals described in the previous section. This process resulted the following improvement: after manually evaluating X (to check) sentences, we were able to increase the number of examples for the *enable* relation from 18 to 100 and for the *prevent* relation from 13 to 81. As shown in Table 4.3 After this improvement, the new stats are the following:

Relation type	Cause	Intend	Prevent	Enable	Not-Cause
Number of relations	283	44	81	100	3

Table 4.3: Final number of relations validated by annotators for each relation type after including candidates from the AFP dataset.

[1] https://www.afp.com/en

4.2 Data Augmentation with LLMs

The initial dataset described above has two major limitation: its size and the large unbalance between relation types. In this section, we describe our efforts for overcoming these limitation using augmentation techniques with LLMs.

Our data augmentation strategy for expanding the dataset is based on the automatic generation of sentences using a prompt-based model. Using the right prompt as input, the model would provide new synthetic sentences for enriching the dataset.

We use the GPT-3 language model [15], and more precisely the GPT-3.5 *text-davinci-003* variant as described in the OpenAI documentation.[2] We are interested in generating sentences that involve events and relationships between them, particularly those related to prevention, intention, and enabling.

4.2.1 Prompt-based Sample Generation of Sentences

When designing the prompt utilized to generate synthetic examples for a specific relation type, we include:

1. the definition that the FARO ontology assigns to that relation type;
2. a subset of relevant examples from the dataset.

We consider a sequence of words Xi = [x_1, x_{t1}, ..., $_{t2}$, x_n], representing an event relationship occurring between two Relata, of a specific relation type ER_x. The words x_{t1} and x_{t2} respectively represents in the text the two Relata which are the subject and the object of the relations. The definition of the relation type *definition(ER_x)* is taken from the FARO ontology.

The selection of the prompt is done after a series of attempts. For sentences generation, we started by leveraging only the task description in the prompt. Therefore, the generated sentences where too short and basic, while we need realistic and longer sentences, similarly to those in the *Original Dataset*.

Table 4.4 demonstrates an effort to prompt the model to produce sentences that showcase connection between events with the desired relation type, but the resulting answer falls short of meeting our intended expectations.

The prompt text to generate sample sentences including relations of type ERx is written as the following:

[2] https://platform.openai.com/docs/guides/completion

Table 4.4: Example of prompting attempts that fell short of producing the desired results

Prompt	Answer	Limitation
Give me an event that enables the happening of other event.	One event that enables the happening of another event is a person's decision to take an action. For example, a person's decision to get up and walk across a room enables the person to arrive at their destination.	• Prompt had no examples from the existing dataset. • Answer was too simple. • Did not describe a real-world scenario. • Explanation was given instead of an actual expression.
Describe a situation where an event is an intention to cause another event, for example: Companies such as Microsoft or a combined worldcom MCI are trying to monopolize Internet access.	An example of an event intended to cause another event is when companies such as Microsoft or a combined worldcom MCI attempt to monopolize Internet access. By controlling the majority of the market, these companies can dictate the terms and prices of access, potentially limiting consumer choice and driving up costs...	• Explanation was given instead of an actual expression.

Prompt(ERx) = *definition(Event)* + *definition(ER_x)* + *request(ER)* + *examples(ERx)*

This prompt definition concerns prevention and intention relations. In the context of *enabling* relation, we include the definition of a condition as follows:

Prompt(ER_{enable}) = *definition(Event)* + *definition(Condition)* + *definition(ER_{enable})* + *request(ER)* + *examples(ERx)*

where *request(ER)* refers to the task description that is given to the language model along with the definitions and *examples(ERx)* are randomly-selected examples from the existing dataset which will be used to iteratively expand and reformulate the dataset.

Example: Prompt used to generate sentences with event relation of type *Enabling*

definition(Event)	An event is a possible or actual event, which can possibly be defined by precise time and space coordinates.
definition(Condition)	A condition is the fact of having certain qualities, which may trigger events.
definition(ER_{enable}A)	The *enables relationship* connects a condition or an event (trigger1), with an other event (trigger 2) it is contributing to realize as an enabling factor.
request(ER)	Give me very long political example sentences following these examples and give me each sentence in one line.
examples(ER_{enable})	...

Note that the original dataset was re-annotated based on Timebank [118] and Event Causality dataset [80], both of which are derived from news articles. This makes the majority of the sentences falling within the political domain. Therefore, introducing the word political in the prompt is to ensure that the generated sentences were coherent and consistent with the original dataset domain.

4.2.2 Prompt-based Event Trigger Annotation

Similarly to sentence generation, we leverage definitions of events to the prompt, adding few examples illustrating the right position for event triggers for each relation types. The prompts have been chosen to acquire the most similar sample pattern to facilitate parsing.

For an event relationship ERx including prevention or intention, the prompt for selecting their event trigger words is designed as follow:

Prompt Event Triggers ERx= definition(Event) + definition(ER_x) + $request_{trig}$(ERx, sentence, x_{t1}, x_{t2})

where the last element is the description of the task of retrieving event triggers from the text. This request takes the following shape:

> If in this sentence <TEXT OF THE SENTENCE> is present an expression with a <RELATION TYPE> relationship between <x_{t1}> (trigger1) and <x_{t1}> (trigger2), what would be the trigger1 and trigger2 in these sentences? Give me only one single word for each trigger an only two triggers per sentence. Put each pair between parentheses in a separate line.

For event relations of type *enable*, the definition of the condition is modified in the following way:

Prompt Event Triggers ER_{enable}= definition(Event) + definition(Condition) + definition(ER_{enable}) + $request_{trig}$(ER_{enable}, sentence, x_{t1}, x_{t2})

Example: Prompt used to generate event triggers with event relation of type *Prevention*

definition(Event)	An event is a possible or actual event, which can possibly be defined by precise time and space coordinates.
definition(Condition)	A condition is the fact of having certain qualities, which may trigger events.
definition(ER_{enable})	The *prevent relationship* connects an event (trigger1) with the event (trigger 2) for which is the cause of not happening.
request(ET)	If in this sentence *"Subcontractors will be offered a settlement and a swift transition to new management is expected to avert an exodus of skilled workers from Waertsilae Marine's two big shipyards, government officials said."* is present an expression with a *prevention* relationship between *settlement* (trigger1) and *exodus* (trigger2), what would be the trigger1 and trigger2 in these sentences? Give me only one single word for each trigger and only two triggers per sentence, put each pair between parentheses in a separate line.

4.2.3 Manual Validation

We use these methods and we generate 600 sentences with each of the relations.

To guarantee the accuracy of the generated set of samples and their appropriate event triggers, we manually validate each synthetic sentence, ensuring its adherence to the given definition.

Overall, 90.77% of all generated sentences were correctly representing an event relation of the requested type. After removing the wrong samples from the dataset, we proceed checking the correctness of their extracted event triggers for the remaining correct sentences.

The generated events triggers were not consistent in term of their patterns from one generation to another. For this reason, an additional parsing step was needed. For doing that, we identified the different textual patterns, processed and categorized these patterns by removing irrelevant words such as '(trigger 1)', and retaining only the precise word or sequence of words that represent the essential part of the event. We were able to identify roughly 12 different patterns. Some examples are reported in Table Table 4.5.

Table 4.5: Three of the different textual patterns which GPT-3 was returning in output for the Event Triggers selection.

Pattern Number	**Event Triggers**
0	Entitles, Buy
1	Approval (trigger1), Acquire (trigger2)
2	"Trigger1 (military): success Trigger2 (diplomatic): risks"

After this processing, we validated the correctness of the two trigger words, measuring an accuracy of 75.15% for trigger-1 and 66.82% for trigger-2. Sentences with wrong triggers were not eliminated from the dataset, but instead manually fixed. Table Table 4.6 shows the detailed accuracy scores for each relation type.

We merged the synthetic data with our original dataset, resulting into a larger and more diverse dataset. after duplicates cleaning we ended up with 1228 new sentences – with relative event triggers – making a total 1891 sentences.

Test Set

The test set of the dataset is taken purely from the news dataset. After investigation we have found it a bit polluted since few sampels from the test set were used as seeds for generation. In order to address the issue of similarity among certain test samples, we have removed sentences from the test set based on a similarity assessment conducted using SentenceBERT

Relation types	Intention	Prevention	Enabling	Total
Correct Generated Sentences(%)	93.82	97	81.5	90.77
Correct ET1 (%)	75.13	81.83	68.5	75.15
Correct ET2 (%)	73.47	77	50	66.82
Number of Checked Examples	600	600	600	1800

Table 4.6: Percentage of Correct Sentences and Event Trigger Words with GPT-3

embeddings [94] and cosine similarity (threshold of 90% similarity). We compensated for the reduction of around 21% by incorporating additional sentences that contained the necessary relations. To achieve this, we manually examine 4130 sentences of the AVeriTeC dataset [103], which consists of news data, in order to extract 216 sentences annotated with fine-grained causal relationships. These sentences are solely used on the test set.

The statistics of this new dataset – latter in the text named the *Augmented Dataset* – are reported in Table 4.7.

Relation Type	Original Dataset	Augmented Dataset
Prevent	81	500
Enable	100	450
Intend	42	459
Cause	268	268
No-Relation	172	172
total	663	1849

Table 4.7: Augmented Dataset Statistics

However, some limitations have also been observed: the enhanced dataset is still largely imbalanced, with the less represented classes (*no relation* and *direct cause*) having around a third of samples of the most represented ones.

To tackle the imbalance issue of previous dataset regarding the *directly cause* relation type, we explored additional pre-existing datasets containing causal relations. Specifically, we incorporated causal examples from the Causal News Corpus (CNC) [114], a dataset consisting of 3,417 annotated sentences, from which, 1,811 sentences contain cause-effect annotations, while the rest are labeled as non-causal. After analyzing several examples from the dataset, we mapped them to the direct causality relation defined in the FARO ontology.

4.3 Data Augmentation with Common Sense

In addition to re-use existing datasets, we propose to improve event relation coverage by leveraging existing common sense knowledge bases and generating additional examples using LLMs. Our strategy involves extracting structured knowledge from sources such as ATOMIC and augmenting it with LLM-generated data to fill gaps, particularly for under-represented relations such as *enabling* and *prevention*. Additionally, we incorporate *negative sampling* to introduce the *no-relation* type, helping the model to recognize cases where no event relation exists. Our method consists of the following key steps:

- We extract relevant event relations from ATOMIC, a large-scale common sense knowl-

Subject	Relation	Object	Mapped relation
PersonX looks before you leap.	xIntent	to be cautious	intends-to-cause
PersonX looks towards PersonY	xWant	to greet PersonY	intends-to-cause
PersonX loses 15 pounds	xEffect	has more energy	causes

Table 4.8: Example Triples from the ATOMIC Dataset with FARO Mappings

edge graph.

- Recognizing that ATOMIC lacks certain key event relations such as *enabling* and *prevention* we expand the dataset by generating new examples with LLMs. This process involves:

  - Using an LLM to generate diverse sentences corresponding to the missing relations.
  - Iteratively refining the dataset by feeding these examples back into an LLM to increase variety and ensure broad domain coverage.

The following subsections provide a detailed explanation of each step.

Leveraging Common Sense Data from ATOMIC

ATOMIC is a large-scale knowledge graph designed to enhance deep learning models' ability to perform *if-then* reasoning and reason about familiar events by leveraging crowd-sourced knowledge extraction. It contains over 877k inferential knowledge tuples that describe commonsense situations. Unlike traditional approaches that rely solely on taxonomic information from corpora, ATOMIC provides examples of everyday situations that are considered common sense.

Among those, we are interested on types that can be mapped to the FARO relations. We observe that some relations – *(o|x)Want/ xIntent* and *(o|x) Effect*[3] – overlap with the definition of the direct cause and intention relationships (Table 4.8). Consequently, we extract these examples from ATOMIC and include them in our dataset under the *intend* and *cause* relation types. Furthermore, since ATOMIC lacks event relations such as *enabling* and *prevention*, we expand the dataset by generating new examples, as described in the next section.

[3] In ATOMIC, the prefix 'x' typically represents the person or entity which is the subject of an action. For instance, 'xIntent' signifies the intention of person X, the entity initiating the action. On the other hand, the prefix 'o' stands for 'others,' indicating the impact or perspective from the viewpoint of those affected by the action. For example, 'oEffect' denotes the effect of person X's actions on others, capturing the consequences or observed outcomes of their behavior.

Augmenting Common Sense Knowledge for Event Relation Extraction

We make use of a LLM to generate diverse sentences corresponding to the relations *enabling* and *prevention*. Additionally, we apply an iterative refinement process where the generated examples are fed back into the LLM for further augmentation, increasing variety and ensuring broad domain coverage. The prompt that we have used is given in Figure 4.1, noting that for the first iteration, we omit the examples part only.

To ensure high-quality generated data, we evaluate multiple LLMs. The selection process is based on a manual review of the initial outputs, assessing their coherence, diversity, and adherence to expected relation types. The best-performing model is then used for subsequent data generation and refinement.

Task Description

Objective:
Generate augmented sentences containing two events with a **RelationType** relationship.

Definition: *Relations and Events*

Task Instructions:

- Sentences must depict common-sense scenarios.
- Each sentence should include two events.
- The generated examples should adhere to the provided structure.
- Do not generate more than the requested number of sentences.

Output Format:

Output:
Generate examples with **RelationType** relations following the specified format.

Figure 4.1: Prompt structure for generating commonsense examples for a given relation type

We evaluated three models known for their state-of-the art performance: **Llama2** [115], **Zephyr** [117], and **Truthful-DPO-TomGrc FusionNet**. Truthful-DPO, based on Mixtral, a sparse mixture of experts model (SMoE) developed by MistralAI, was the top-performing model on the Hugging Face leaderboard at the time of development[4]. A manual review of the first 20 outputs show that Zephyr and Truthful-DPO produced accurate annotations, whereas Llama2 struggled for correctly identifying spans of text and was therefore excluded from further experimentation.

[4]https://huggingface.co/spaces/Mikelue/yunconglong-Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B

To help the model to identify the cases where no event relation exists, we introduce **negative samples**. This is achieved by restructuring the dataset into event triples and randomly swapping either the subject or the object in a way that invalidates the original relation. This approach ensures that the model learns to distinguish between valid and spurious event relations.

Manual Assessment

To assess the quality of the generated data, we have manually reviewed 100 generated examples for each relation type in order to identify systematic errors. Specifically, we filter out incorrect annotations by removing patterns that frequently lead to miss-classifications, such as sentences containing contrastive conjunctions or contradiction-related terms. For example, while instructing the LLM to generate a sentence for the causal relationship “enable”, we obtained the following generated sentence: <ARG0>Introducing salt</ARG0> into boiling water, <ARG1>it prevents</ARG1>. which makes use of the term “prevents” yielding a contradiction. This sentence is filtered out.

4.4 Combined Dataset

We finally constructed a combined dataset that combine sentences from both the news datasets and the common sense datasets in order to evaluate whether integrating common sense knowledge enhances fine-grained causality extraction or not. This combination allows us to assess the impact of augmenting real-world textual data with generalizable causal patterns. To ensure a balanced representation of causal relations, we maintain an equal number of samples per relation type across both sources (news and commonsense). Specifically, for each relation present in the news dataset, we include an equivalent number of examples from the commonsense dataset. This combined dataset allows us to examine whether introducing structured common sense knowledge improves the model ability to capture causal dependencies beyond what is observed in news texts alone.

Table 4.9 summarizes the final dataset statistics after augmentation and duplicate cleaning.

4.5 Modeling the Extracted Event Relations in a Knowledge Graph

In order to foster application also in other fields, we shaped our dataset also as the Knowledge Graph (KG), following common Semantic Web principles.

For the sake of example, let’s consider the following sentence:

Category	Dataset	Total	Cause	Enable	Prevent	Intend	No-rel.
News Data	Original Data	663	268	100	81	42	172
	Synthetic Data	1,228	0	350	419	459	0
	CausalNews Corpus (CNC)	3,316	1,710	0	0	0	1,606
Common Sense	ATOMIC	315,173	82,242	0	0	146,588	86,943
	Synth. Common Sense	205,884	0	65,485	53,456	0	86,943
Total		526,264	84,321	66,025	54,067	147,189	175,664
Combined dataset		6792	3520	814	948	944	566
Test dataset		632	351	89	52	40	100
	including AVeriTeC	216	133	46	26	11	0

Table 4.9: Dataset Statistics

The spokesman said that the proposed guidelines caused Crossland to revise its business objectives

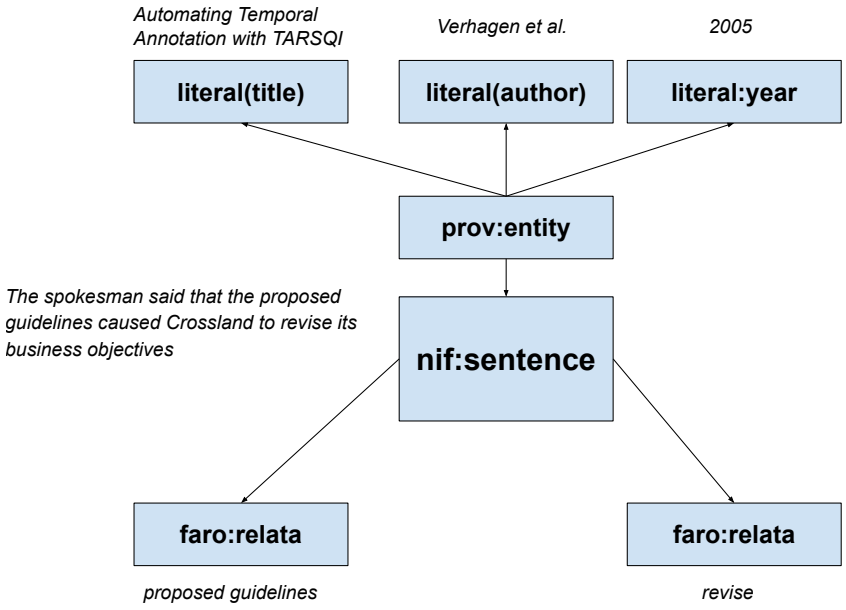

Figure 4.2: Knowledge Graph Schema

The schema of the Knowledge Graph is reported in 4.2, in which it is possible to appreciate a group of entities. The group is composed by the sentences of the news articles, their provenances, and the described event relation. For the former, we employed the NLP Interchange

Format (NIF)[5] [40]. The event relation is instead reported using the previously mentioned FARO ontology. To ensure traceability of our data, each sentence has an associated provenance. We achieve this by assigning a provenance node (*prov:Activity*) to every sentence, which links the extracted knowledge to its original source.

In this design, sentences (*nif:sentence*) can reveal in the text (*nif:word*) events interconnected by a relation (e.g., *faro:causes*). To ensure traceability of our data, each sentence is assigned a provenance, linking extracted information to its original source. This is achieved by associating each sentence with a provenance node (*prov:Activity*) that includes metadata such as the title, authors, and publication year of the source.

The provenance information is assigned based on the *source* of the data:

- Sentences extracted from news datasets are linked to their original sources, either TimeBank, EventStoryLine, or AFP , depending on their origin. Those sourced from the *Causal News Corpus (CNC)* are specifically referenced from [114] to acknowledge their original authors.
- Sentences derived from the *ATOMIC knowledge graph* are distinguished from sentences generated by language models. While ATOMIC sentences are linked directly to their original work, those generated by models like Zephyr or GPT-3 are explicitly attributed to software using the (`prov:SoftwareAgent`) property.
- Sentences labeled as *no-relation* are synthetically created by swapping subjects and objects from existing sentences. Such sentences are also attributed to software as their provenance.

To ensure consistency and transparency, each provenance node contains additional metadata:

- The *dcterms:title* property stores the dataset or paper title.
- The *dcterms:creator* property lists the authors of the dataset or publication.
- The *fabio:hasPublicationYear* property records the year of publication.

By structuring provenance in this way, we ensure that each extracted relation maintains a clear lineage, improving transparency, validation, and reproducibility in downstream applications.

We published this KG using FAIR principles, including link disambiguation, in a public SPARQL endpoint and through dumps in Turtle format available on the open-source repository[6].

[5] https://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html
[6] https://github.com/ANR-kFLOW/knowledge-graph/tree/main/dump/KG

4.6 Conclusion

In this chapter, we have outlined the construction of a novel event relation dataset that focuses on capturing a range of causal and contingent relations, extending existing datasets such as TimeBank and EventCausality. Through the introduction of RLINK and the incorporation of event relations such as Cause, Intend, Prevent, Enable, and Not Cause, we have expanded the capacity for fine-grained causal reasoning. Our approach combines semi-automatic annotation, manual validation, and large language model-based augmentation to overcome the challenges of dataset imbalance and limited coverage of specific relations. By integrating commonsense knowledge from ATOMIC and leveraging LLMs for synthetic data generation, we have created a robust and diverse dataset. This dataset, structured as a knowledge graph, provides a valuable resource for further research in causal event extraction, narrative generation, and automated fact-checking systems. We have create a KG out of this dataset having all the sentences of the news articles and the extracted FARO relations between them together with the origins of these sentences using the provenance property.

This dataset will serve as the foundation for training an ERE system, which we will be described in details in the following chapter.

Chapter 5

Event and Event Relation Extraction from Text

Event Relation Extraction from text remains a challenging task in the information extraction community, particularly when dealing with fine-grained causal relationships between events such as *directly cause*, *enable*, *prevent*, or *intend*. In this chapter, we introduce a novel model capable of extracting these refined causal relations from text. We explore and compare various techniques for relation classification and event extraction, both as independent tasks and within a multitask learning framework. Additionally, we evaluate the effectiveness of zero-shot and few-shot prompting for different LLMs in comparison to Pre-trained Language Models (PLMs). Our experiments demonstrate the added value of commonsense knowledge and show that PLMs trained on news data enriched with commonsense reasoning outperform LLMs for this task.

5.1 Causal Event Relation Extraction: Literature Review and Gap Analysis

Various techniques have been employed to extract event relations from texts, including supervised, unsupervised, semi-supervised, and distant supervision methods. These approaches primarily focus on extracting specific event relations, such as causality, temporality, and coreference [72]. Auto-regressive sequence-to-sequence models, such as REBEL, classify event relations by extracting triples *(subject, relation, object)* from text [46]. LLMs have also shown promise in this domain [45]. For example, the Flan-T5 model combined with chain-of-thought prompting and few-shot learning using GPT-3.5 outperformed the REBEL baseline for relation extraction on the CoNLL04 dataset [124]. Additionally, LLMs have been applied to causal relationship extraction from tabular data [71]. Recent advances have used PLMs to improve event relation extraction. [104] has combined data augmentation using GPT-3.5 with a RoBERTa-based multi-layer tagging approach to identify multiple causal relations in a single sentence. This method achieved top performance in the Event Causality Identification Shared

Task 3 at CASE 2023 [114].

However, existing approaches typically focus on coarse-grained or generic causal links and have not been applied to the extraction of refined causal relations —such as *prevent*, *enable*, or *intend*—as formalized in our data model. This gap motivates the need for dedicated models capable of capturing such nuanced semantic distinctions to be be able to apply them in downstream application.

5.2 Approach

Event relation extraction can be decomposed into three interconnected subtasks:

(1) **Relation Detection (RD)** involves identifying whether a causal relation exists or not between two events. This task can be framed as a binary classification problem;

(2) **Relation Classification (RC)** involves sequence classification, where sentences are classified into one of the target relations. The set of relations will be in our case: *cause*, *enable*, *prevent*, *intend*, or *no-relation*;

(3) **Event Extraction (EE)** is the process of span detection, which precisely identifies the spans of text that represent the subject and the object of the relation, referred to as *event1* and *event2*.

We explore three strategies for Event Relation Extraction. (1) The first strategy decomposes the task into the three subtasks – training and testing each subtask separately – with the aim of assessing whether this decomposition reduces complexity and enhances performance or not. (2) The second strategy adopts a multitask learning framework, where a single model jointly learns all three subtasks in an end-to-end manner, leveraging shared representations. (3) The third strategy relies on prompting available Large Large Models. This latter strategy does not require any training nor fine-tuning step, but few-shot prompting needs a handful of examples.

5.2.1 Fine-Grained Causality Extraction as Three Separate Subtasks

In this strategy, the three subtasks are handled separately.

Relation Detection. This task is modeled as a binary sequence classification problem that aims to decide whether a particular sentence contains a fine-grained causal event relation or not. This serves as a preliminary step as subsequent processes will only be executed when a relation is detected. This approach minimizes unnecessary computational costs and mitigates

potential sources of confusion. We categorize the sentences in our dataset into positive examples (class 1) if they contained relations such as cause, enable, prevent, or intend, and negative examples (class 0) otherwise. For this task, we trained a simple transformer-based binary classifier.

Relation Classification. This task constitutes a sequence classification problem with five distinct categories. For the relation classification, we trained a transformer-based model that receives a sentence as input, trained to classify the sentence into one of five classes: *cause*, *enable*, *prevent*, *intend* or *no-relation*, which are returned in output after a linear activation module.

Event Extraction. The detection of the events composing the relation is cast as a token classification problem. The spans are annotated following the BIO[1] tagging scheme [91]. We provide below an example including a causal relations between the subject *prolonged drought* and the object *severe water shortage*:

"The(O) **prolonged**(B-C) **drought**(I-C) across(O) the(O) region(O) resulted(O) in(O) **severe**(B-E) **water**(I-E) **shortages**(I-E) and(O) crop(O) failures(O), leading(O) to(O) economic(O) hardship(O) for(O) local(O) farmers(O)."

We trained again a transformer-based model to predict the appropriate BIO tag for each token.

For these three subtasks, we experiment with both BERT and RoBERTa [140]. For relation detection and relation classification, we used the pre-trained version of the two models for sequence classification.[2] For the event extraction sub-task, we leverage a version of the models specifically pre-trained for the Named Entity Recognition (NER) task.[3] These models are fine-tuned on our dataset for fine-grained causality extraction.

During inference, we first use the relation detection model as a filter to exclude cases where no causal relation exists. Furthermore, we also train and test our model on detecting these cases (*no relation*) for the relation classification and event extraction subtasks. This allows us to evaluate their effectiveness in handling potential false positive leakage from the initial filter coming out of the relation detection model.

[1] BIO = *Beginning, Inside, Outside*

[2] https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#transformers.BertForSequenceClassification, https://huggingface.co/FacebookAI/roberta-large

[3] https://huggingface.co/dslim/bert-base-NER,https://huggingface.co/51la5/roberta-large-NER

Require: Input tokens T, batch size B, max seq length L, relation classes R, BIO labels N
Initialize: Encoder, classification heads (RD, RC, EE)
Initialize: Loss functions $\mathscr{L}_{RD}$, $\mathscr{L}_{RC}$, $\mathscr{L}_{EE}$
Encode input: $H \leftarrow \text{Model}(T)$
Extract pooled output P and sequence output S
Relation Detection:
$RD\_logits \leftarrow \text{Linear}(P, 2)$
Compute loss: $\mathscr{L} \leftarrow \mathscr{L}_{RD}(RD\_logits, labels_{RD})$
Initialize Logits:
$RC\_logits \leftarrow \mathbf{0}_{(B,R)}$
$EE\_logits \leftarrow \mathbf{0}_{(B,L,N)}$
for $i \leftarrow 1$ to B **do**
    if $\arg\max(RD\_logits[i]) = 1$ **then**
        $RC\_logits[i] \leftarrow \text{Linear}(P[i], R)$
        $EE\_logits[i] \leftarrow \text{Softmax}(\text{Linear}(S[i], N))$
    else
        Assign $RC\_logits[i, 'no-relation'] \leftarrow 1.0$
        Assign $EE\_logits[i, :, 0] \leftarrow 1.0$
    end if
end for
Compute additional losses:
$\mathscr{L} \leftarrow \mathscr{L} + \mathscr{L}_{RC}(RC\_logits, labels_{RC}) + \mathscr{L}_{EE}(EE\_logits, labels_{EE})$
Return: $\{\mathscr{L}, relation\_logits, type\_logits, span\_logits\}$

Figure 5.1: Multi-Head RoBERTa pseudo-code for Causality Extraction

5.2.2 Fine-Grained Causality Extraction as an End-to-End Pipeline

In this strategy, a single model is trained using multi-task learning to predict a complete triplet as output given an input text. This approach enables the model to handle all three sub-tasks within a unified architecture. We experiment with two models: sequence-to-sequence (*seq2seq*) architectures (encoder-decoder) and classification architectures (encoder only).

seq2seq architectures have demonstrated strong performance in relation extraction tasks by converting raw text into structured representations [46]. These models typically employ an encoder-decoder framework, where the input text is processed to generate structured triplets representing causal relationships.

Example input:

> "The government's swift action to impose a lockdown **prevents** the rapid spread of COVID-19 among the population."

Output:

> **<triplet>** impose a lockdown **<subj>** spread of COVID-19 **<obj>** prevent

REBEL is an auto-regressive seq2seq model based on BART [66], composed of an encoder and a decoder layer, which has achieved state-of-the-art performance in general relation extraction. REBEL processes raw text and returns as output a triplet *(Subject, Relation, Object)*. This approach is similar to a natural language translation, but the translation concerns text to triplets rather than language to language.

Alternatively, we use a classification-based approach to treat causality extraction as a multi-task classification problem. We designed a model architecture with three classification heads to handle the three different subtasks:

1. **Relation Detection Head**: a binary classifier determines whether a causal relation exists in the given input text.
2. **Relation Classification Head**: a multi-class classifier predicts the specific type of causal relation (out of five possible categories).
3. **Event Extraction Head**: a token-level classifier applies BIO tagging to identify the spans of text corresponding to cause and effect within the text.

If the first classifier predicts that no causal relation exists, the model assigns the label *no-relation* to the relation classification head and labels all tokens with *O* in the BIO tagging scheme. Otherwise, the model proceeds with the other two classification heads to predict the relation type and extract relevant events. The pseudo-code of this approach is shown in Figure 5.1.

We use RoBERTa that has shown effectiveness in generic causality extraction and event extraction tasks [64, 104]. Both REBEL and RoBERTa have never – in previous work – been explicitly trained or evaluated for fine-grained causality extraction, where relations are more semantically precise and aligned with a specific relational schema.

5.2.3 LLMs as Relation Classifiers and Event Extractors

LLMs have demonstrated strong performance across various NLP tasks, including language understanding, text classification, and information retrieval [18]. Their effectiveness is driven by training on massive datasets and the application of advanced learning techniques such as self-supervised learning, fine-tuning, instruction tuning, reinforcement learning, and in-context learning, enabling them to generalize across a wide range of tasks [138]. In this work, we have assessed the performance of some LLMs on fine-grained causality extraction task and compare their performance with other pre-trained models that are discussed above.

We have considered both zero-shot and few-shots prompting. Given a sentence from our dataset, along with the definitions of each relation we aim to extract, we prompt the LLMs to

extract the subject, object, and relation from the text as illustrated in Figure 5.2. The output of the LLMs is parsed to extract the subject, object, and relation from a given text. This process is repeated along all the test set. The implementation is represented in Figure 5.3 and makes use of the LangChain framework[4] [19] for easy interaction with different LLMs.

In our experiment, we compare the performances of both a closed model (GPT-4o[5]) and an open weights model (Zephyr-7B-beta-AWQ[6]). The latter is a variant of the Zephyr-7B-beta model[7] quantized with Activation-aware Weight Quantization (AWQ) . This lighter version of the model allows for faster inference and requires less memory.

Prompt: You are an expert in fine-grained causality extraction. Your task is to extract the subject, object, and relation from the given sentence. The relation must be one of the following: *cause*, *enable*, *prevent*, *intend*, or *no_relation* (if none of the refined causal relations apply).

Relation Definitions: The definitions of **cause**, **intend**, **enable**, and **prevent** are based on the FARO ontology.

Task Instructions: Extract the **subject**, **object**, and **relation** for the following sentence.
Sentence: `"input_sentence"`

Output Format: `Subject: <extracted_subject>, Object: <extracted_object>, Relation: <extracted_relation>`

Important Guidelines:

- The relation must be one of: *cause*, *enable*, *prevent*, *intend*, or *no_relation*.
- Extract the **actual words** from the sentence for both **subject** and **object**.
- **DO NOT** use placeholders like '<subject>', '<object>', or '<relation>' in the output. Always provide extracted values from the input sentence.
- If the sentence does not contain any of the four refined causal relations, output: `Relation: no_relation`.

Examples: `examples`

Figure 5.2: Fine-grained causality extraction prompt

5.3 Experiments

This section presents an analysis of the model performance across different datasets and strategies, focusing on relation detection, relation classification, and event extraction. Table 5.1 presents the experiment results.

[4] https://python.langchain.com/
[5] https://openai.com/index/gpt-4o-and-more-tools-to-chatgpt-free/
[6] https://huggingface.co/TheBloke/Zephyr-7B-beta-AWQ
[7] https://huggingface.co/HuggingFaceH4/Zephyr-7b-beta

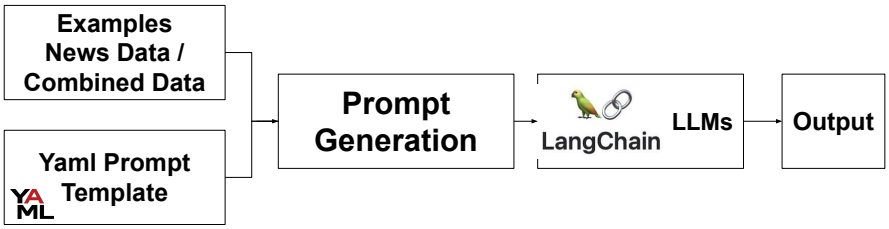

Figure 5.3: Workflow of Event Relation Extraction with LLMs

Dataset	Strategy	Model	Relation Detection			Relation Classification			Event Extraction			Average F1
			P	R	F1	P	R	F1	P	R	F1	
News	End-to-End	REBEL	0.86	0.42	0.56	0.75	0.60	0.65	0.58	0.65	0.59	0.60
		RoBERTa	0.98	0.98	**0.98**	0.79	0.71	0.74	0.19	0.20	0.20	0.64
	Separate	BERT	0.94	0.85	0.89	0.8	0.76	**0.77**	0.58	0.64	0.61	0.76
		RoBERTa	0.86	0.94	0.89	0.71	0.77	0.73	0.66	0.67	0.66	0.76
Combined	End-to-End	REBEL	0.72	0.51	0.60	0.77	0.74	0.75	0.74	0.68	**0.70**	0.68
		RoBERTa	0.99	0.96	**0.98**	0.75	0.80	**0.77**	0.19	0.20	0.20	0.65
	Separate	BERT	0.93	0.91	0.92	0.68	0.74	0.7	0.56	0.64	0.60	0.74
		RoBERTa	0.93	0.92	0.92	0.71	0.76	0.73	0.67	0.61	0.64	**0.763**
News	LLM	GPT4 (0-shot)	0.18	0.85	0.29	0.54	0.37	0.33	0.37	0.23	0.23	0.29
		Zephyr (0-shot)	0.19	0.79	0.31	0.31	0.32	0.23	0.25	0.25	0.24	0.26
		GPT4 (2-shot)	0.45	0.83	0.59	0.52	0.62	0.53	0.41	0.43	0.42	0.51
		GPT4 (4-shot)	0.40	0.97	0.57	0.55	0.64	0.54	0.46	0.44	0.45	0.52
		Zephyr (2-shot)	0.38	0.66	0.48	0.42	0.51	0.44	0.29	0.28	0.27	0.39
		Zephyr (4-shot)	0.17	0.98	0.29	0.28	0.22	0.10	0.23	0.21	0.20	0.19
Combined	LLM	GPT4 (2-shot)	0.34	0.88	0.49	0.49	0.58	0.46	0.36	0.35	0.35	0.43
		GPT4 (4-shot)	0.28	0.92	0.43	0.50	0.55	0.44	0.34	0.33	0.32	0.39
		Zephyr (2-shot)	0.22	0.94	0.36	0.41	0.38	0.30	0.29	0.29	0.29	0.31
		Zephyr (4-shot)	0.16	0.95	0.27	0.46	0.22	0.11	0.30	0.30	0.30	0.23

Table 5.1: Combined performance across subtasks with Precision (P), Recall (R) and F1-score (F1) and an average F1-score on the cleaned test set derived from [92]

5.3.1 Comparison of End-to-End vs. Separate Strategies

For both the news and combined datasets, in 66.66% of the cases, the models trained using the end-to-end strategy outperform those trained on separate tasks. This confirms the benefit of shared parameter learning. For relation classification on news data alone, the performance of both approaches remains comparable. Conversely, for EE, the separately trained RoBERTa model outperforms its end-to-end counterpart, indicating that the shared parameter approach was less effective for this task.

5.3.2 Impact of Common Sense Knowledge Integration

Integrating commonsense knowledge into the dataset (i.e. using the combined dataset) leads to notable improvements, particularly for end-to-end models. In 70% of cases, models trained on the combined dataset performed either better or in par with their counterparts, with 67% of these cases showing improved performance. Notably, in the remaining cases, the performance

drop was minimal, with a maximum decline of only 0.07%.

For instance, training REBEL on the combined dataset enhances its RD F1-score by 4%, RC by 10%, and EE by 11%. Similarly, RoBERTa, in an end-to-end setting, shows a slight improvement in RC while maintaining stable performance across other tasks.

In contrast, models trained separately benefit primarily in RD, while their performance in other subtasks remains unchanged or only marginally improves when trained on news data alone. This suggests that commonsense knowledge enhances the understanding of event relations, but its impact varies based on the learning strategy. It also implies that models explicitly trained on individual subtasks may already capture implicit commonsense knowledge, limiting the benefits of additional augmentation.

5.3.3 LLMs for Fine-Grained Causality Extraction

The performance of GPT-4 and Zephyr varies significantly across different tasks and settings. In general, GPT-4 consistently outperforms Zephyr in all settings, achieving higher F1-scores in RD, RD, and EE. Adding a few-shot context improves performance across most subtasks. The 4-shot setting provides a notable boost, with GPT-4 reaching an F1-score of 0.52 on the News dataset and 0.43 with 2-shot on the Combined dataset. Zephyr also shows improvements but still lags behind GPT-4.

We observe that LLMs do not benefit from the commonsense data. This may be due to the fact that these models are trained on vast amounts of data and already capture essential commonsense knowledge. Consequently, incorporating additional commonsense examples reduces the proportion of news data examples, potentially leading to a decline in performance. Notably, for both 2-shot and 4-shot settings, we randomly select examples from both commonsense and news datasets.

Compared to PLMs, RDs under-perform significantly in structured extraction tasks. The highest-performing fine-tuned models reach an average F1-score of 0.76 (News) and 0.763 (Combined), while GPT-4, even with 4-shot prompting, only attains 0.48 (News) and 0.38 (Combined).

Breaking down performance by subtask, we observe distinct trends across models:

- **Relation Detection**: RoBERTa achieves the highest **F1-score of 0.98** in both datasets using the end-to-end approach, but BERT and RoBERTa in the separate strategy remain competitive with scores around **0.89–0.92**.
- **Relation Classification**: The best performance is observed with the combined dataset using RoBERTa in an end-to-end manner reaching **0.77** and also with BERT on the news

Dataset with the same performance. In this subtask, most of the models benefit from the inclusion of common sense knowledge.

- **Event Extraction**: The best-performing model (REBEL, end-to-end) improves by **11%** when using the combined dataset, suggesting that common sense augmentation is beneficial for event extraction.

5.4 Platform and API for Event Relation Extraction

Our goal is to perform event relation extraction from textual data, focusing on four semantically precise event relations: *Direct-Cause, Enable, Intend, and Prevent*. We converted this pipeline into a Web API and we developed a front-end user interface to make our system accessible to a broader audience, to demonstrate the capabilities of our model in action, and to facilitate hands-on interaction with fine-grained event relation extraction. This also enables researchers and practitioners to test custom examples, explore model outputs, and better understand how causal reasoning is handled in context.

First, the model filters out sentences that do not have a causal event relation. The sentences containing a causal relation are further processed by the RC module to determine which type of event relation is in the sentence from the four relations. Finally, the EE module extracts the subject and the object of the event relation in a given sentence. Figure 5.4 illustrates these modules.

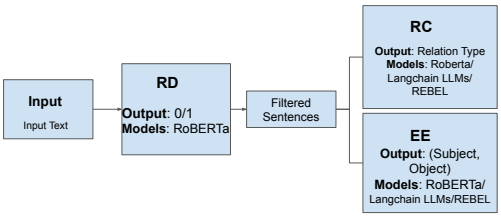

Figure 5.4: The ERE pipeline workflow

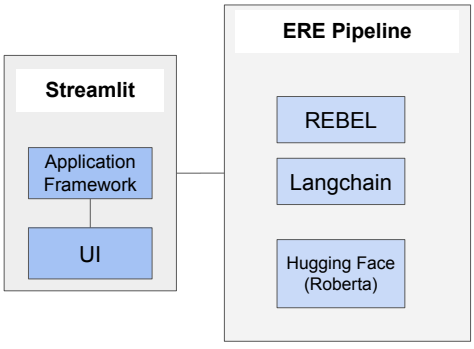

Figure 5.5: Streamlit UI and Application Framework

The front-end application enables users to experiment with the capabilities of the models. This application is developed using Streamlit[8], which acts as both the web application and the web API as shown in Figure 5.5. The Streamlit application receives input from the user and passes it along to the Python pipeline via a configuration file. The user can write his own

[8]https://streamlit.io/

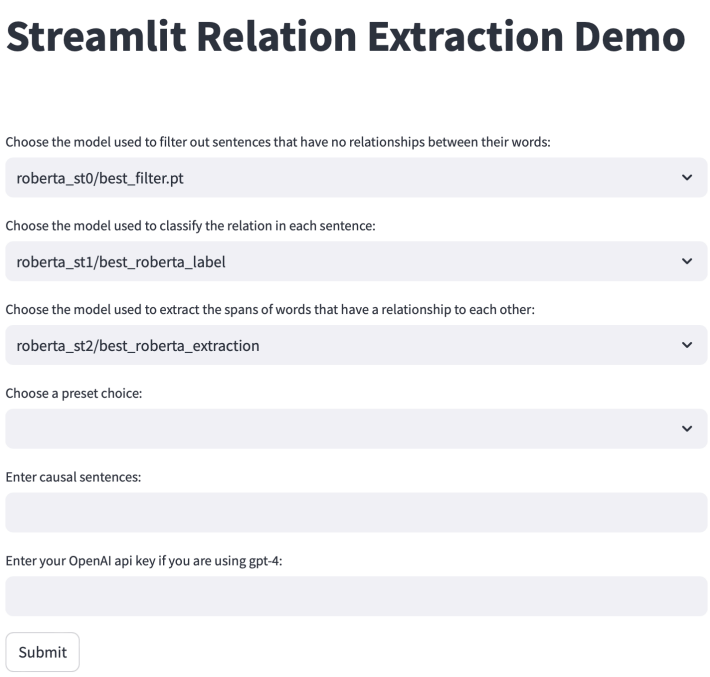

Figure 5.6: Users can configure each step of the event relation extraction pipeline by selecting models for sentence filtering, relation classification, and span extraction. They can choose between preset or custom inputs, and provide an OpenAI API key if using GPT-4. The pipeline runs upon submission.

text or use a preset input. After the user makes his choice of the model used for each task, the inference running in the back-end will be produced.

The output returned to the user will include his original text used to produce the inference with highlights of subject and object of the extracted event relation. Next to the highlights are labels indicating what part of the span it is (subject or object) together with the event relation type.

The classification for the relation can either be: cause, intend, prevent, enable, or other. Other refers to when the model producing the classifications gives a non-standard response. Figure 5.6 shows a screenshot of the demo.

5.5 Conclusion

We thoroughly study the event relation extraction task with a focus on fine-grained causal relations that extend beyond traditional categories. We introduce the first model capable of accurately extracting these refined causalities from text. key take-away:

- The **end-to-end** approach is generally more effective than the separate approach, particularly for relation detection and classification;
- **Common sense knowledge improves performance**, especially for end-to-end models and event extraction;
- **RoBERTa performs best overall**, achieving the highest average F1-score (0.763) when trained separately on the combined dataset;
- **Event extraction remains challenging**, but augmentation with common sense knowledge provides a noticeable improvement;
- **LLMs lag behind fine-tuned PLMs**, with GPT-4 and Zephyr showing potential in few-shot settings but still under-performing compared to models like RoBERTa and BERT in structured information extraction tasks.

To support experimentation and foster accessibility, we developed an interactive pipeline, made publicly available through a Streamlit-based interface. This demo allows users to visualize the event relation extraction process in an intuitive and accessible format. Basically, users can input custom sentences and receive automatically generated causal inferences, enabling hands-on exploration of our system's capabilities.

We believe that the described method can be highly beneficial for a series of downstream tasks and applications. This will be the focus of next part of this thesis.

Part III

Applications

Chapter 6

A Knowledge Graph-Based Storytelling Approach

Narratives play a crucial role in shaping perceptions, beliefs, and decision-making processes. While contemporary pre-trained language models have demonstrated remarkable capabilities in text generation and question-answering, they remain limited in knowledge coverage and are susceptible to societal biases. This work aims to bridge these gaps by integrating knowledge graphs into narrative construction. Rather than solely relying on fundamental aspects such as the 4W (who, what, when, where) and general relationships, our approach incorporates fine-grained semantic relations, capturing nuanced causal dynamics—such as an event preventing, intending to cause, causing, or enabling another event. By leveraging state-of-the-art methods to predict these relations, we show that automatically generated narratives can achieve greater grammatical coherence and semantic accuracy.

To further enhance the depth and richness of generated narratives, we propose expanding the WebNLG dataset [32], originally limited to 4W relations, by incorporating refinecd causal relations form our dataset. This integration enables the generation of more linguistically sophisticated text, embedding richer causal and semantic structures to improve narrative coherence and informativeness.

6.1 Related Work: Storytelling and Narratives

Narratives stand at the heart of our societal fabric, serving our understanding and facilitating the exchange and preservation of knowledge, and cultural heritage. These narratives filter through our everyday lives, appearing in diverse forms such as commercials, political campaigns, news broadcasts, literary works, television shows, and more, each with its unique purpose and significance. What makes narratives truly captivating is their transformative power: they possess the ability to shape our perceptions, instill beliefs, and steer our choices and actions [35]. Consequently, the quest to innovate in the realm of complex narrative generation holds the potential to usher in a new era of AI systems that are intricately attuned to

human sensibilities. Building upon the profound role of narratives in our society, it becomes evident that our means of narrative generation and comprehension are intertwined with the capabilities of modern AI. Pre-trained large language models, exemplified by models such as BERT , and GPT-3, have showcased remarkable progress in text generation, and conversational tasks. Yet, these models, shaped by training on extensive datasets drawn from undisclosed and diverse sources, bear intrinsic limitations, including knowledge gaps, inaccuracies, and societal biases [27]. Their challenges in maintaining semantic coherence and capturing long-term dependencies within text generation further underscore the need for innovation in narrative crafting [67, 75].

A *narrative graph* [13] incorporates two main components: the individual representation of events, including the “four W" aspects (*who*, *what*, *when*, *where*) and the interconnection of these events through temporal and causal relationships. Since no KG existed with the refined causal relations, none was used for narratives generation.

KG summarization entails an initial step of information retrieval and selection. To acquire the essential nodes for event description, an effective approach involves ranking techniques that assign significance to nodes based on the relationships they possess. Various methods can be used such as entity ranking, relationship ranking, and semantic document ranking [51]. [14] proposes a system that can identify relevant information needed to build a narrative graph, by using an informed graph search traversal strategy. To determine which information is considered 'relevant' the method uses filters to prune the search space with respect to the Simple Event Model (What, Who, Where, When).

On the other hand, different methods for generating texts from knowledge graphs have been proposed. In [132], triples are extracted to fine-tune a GPT-2 model [89], making the model dependent on the input triples. A similar approach is introduced in [96], involving BART and T5 [90]. This approach obtained state-of-the-art performances on the AGENDA dataset [60] but not on the WebNLG dataset. Both found that PLMs work well on unordered representations of the graph. JointGT [54] uses BART and T5, and exploits new pre-training methods to explicitly preserve the input graph's structural information. JointGT outperforms the other mentioned technique on WebNLG, which might indicate that including the topology of the graph lead to better results. A different approach [73] uses a transformer encoding structure to encode both the global information and the local topology information, and feeds a transformer to decode and generate text. However, this did not work as well as the previously mentioned technique [96], which used a PLM without encoding. This might indicate that PLMs obtain better results than self trained transformer models.

6.2 Dataset

In this section, we present the datasets that we used to train our method:WebNLG and the portion of our dataset referred to as the FARO dataset in the remainder of this paper, for ease of reference. At the time of writing this section, **only a small portion of the FARO dataset was available**, making it comparatively smaller than the larger dataset used.

For evaluation, we use two evaluation datasets: the FARO test set and the ASRAEL KG [97]. ASRAEL is a knowledge graph that includes various event-related articles and their interconnections, including the 4W relations.

Before our evaluation, ASRAEL lacked precise semantic relations. Therefore, we had to extract these relations from the event articles (linked to the KG) to conduct the assessment. We enhanced the ASRAEL KG with these extracted additional relations (similarly to the ones in FARO), resulting in a more dense and comprehensive knowledge graph. To achieve this objective, we used REBEL to extract events and relations (*cause*, *enable*, *prevent*, and *enable*). Furthermore, we leverage an existing event co-reference resolution model [11] to perform the task within the KG. This model creates clusters of mentions, computes similarity scores for each cluster, merges those with the highest score, and repeats this process until the score fell below a defined threshold, which we empirically set to 0.95. This clustering process resulted in a graph primarily composed of clusters with a single mention, which are due to not finding a similar match. According to our manual assessment, the algorithm matched correctly a large number of syntactic matches, which makes it trustworthy. In total, we successfully clustered 45,031 mentions, with 36,057 being unique. The resulting narrative graph[1] provides a RDF representation of event co-references and relationships, enriched with ontologies such as NLP Interchange Format (NIF) [2], SEM and FARO to describe the relations between triples, further enhancing the context and meaning of our KG.

6.3 Knowledge graph summarization

Knowledge Graph summarization comprises two tasks: the selection of pertinent information from the KG, and the text generation based on the extracted data.

6.3.1 Relevant Information Selection

A SPARQL query has been written to extract the essential nodes, such as persons, places, and times, crucial for narrative construction from a main event within the ASRAEL KG. This query

[1] https://github.com/ANR-kFLOW/KG2Narrative/blob/main/Data/graphs/final_generated/eag_complete_merged.ttl

[2] https://persistence.uni-leipzig.org/nlp2rdf/

prioritizes the selection of events involving the 4W nodes with higher frequencies of incoming edges. Mentions are selected similarly; the larger the cluster of co-referent mentions (formed by the event co-reference model) is, the higher the priority of said cluster. Since we face a limitation on the number of input tokens of the text generation model, up to three mentions are selected from the same cluster.

The quality of the output depends largely on the quality the output of previous steps (relation extraction and co-reference resolution). Future work aims to enhance the accuracy of both these tasks and explore methods for identifying indirectly linked relevant nodes to selected events.

6.3.2 Text Generation from Knowledge Graphs

As anticipated in Section 6.1, using a PLM instead of training a language model from scratch can lead to better results. Furthermore, incorporating the graph's topology into the model has been shown to generate better natural text. JointGT [54] incorporates both of these characteristics, hence, we adopted this method. The authors pre-trained this model on the KGText dataset [20], consisting of 7 million graph-text pairs extracted from English Wikipedia dump.[3] It includes around 1.8 million entities and 1,210 relations.

The WebNLG dataset does not contain any of the FARO relations. Therefore, we fine-tuned the model on a merged dataset, combining the WebNLG and FARO dataset, as in Table 6.1 without making changes to the model itself. The creation of this combined dataset involves the following multi-step process. Initially, entities and their respective encodings are extracted from the WebNLG dataset. Subsequently, entities from the FARO dataset are encoded utilizing the extracted encodings from WebNLG. Finally, the resulting encodings and their relations are integrated into the original WebNLG dataset, thereby producing the combined dataset.

Table 6.1: Sizes of the datasets used for training and evaluating the JointGT model.

Dataset	**Train**	**Val**	**Test**
WebNLG	12,876	1,619	1,600
FARO	1,800	201	108
Combined	14,676	1,820	1,708

The model undergoes fine-tuning on the WebNLG dataset. We refer to the original model as *base model*, and the model fine-tuned on the combined dataset as *combined model*.[4]

[3] https://dumps.wikimedia.org/

[4] The model was replicated using the same parameters from the original paper, except for the batch size lowered due to memory constraints. The parameters are *Learning rate*: 0.000025, *Batch size*: 4, *Epochs*: 10, *Optimizer*: Adam. *Early stopping*: 10 epochs

6.4 Results

6.4.1 Quantitative analysis

Table 6.2: The performance metrics of the best performing model on their corresponding validation and test set – either WebNLG or the combined set. Both models are evaluated also on the FARO test set.

Model	Dataset	BLEU	METEOR	ROUGE	Step	Epoch
Base (WebNLG)	Val	0.6642	0.4727	0.7558	22400	6
	Test	0.6529	0.4681	0.7535	-	-
	FARO test	0.0	0.0565	0.1299	-	-
Combined	Val	0.6368	0.4543	0.7468	36000	9
	Test	0.6101	0.4409	0.7260	-	-
	FARO test	0.0477	0.0877	0.1949	-	-

Table 6.2 provides crucial insights into the model's performance, measured by the BLEU, METEOR, and ROUGE metrics. BLEU emphasizes precision, indicating how accurately the generated text aligns with the reference text. On the other hand, ROUGE focuses on recall, gauging the extent to which the reference text is captured in the generated output. METEOR combines elements of both precision and recall, and its effectiveness can be further enhanced by incorporating improved word matching strategies. ROUGE suggests a high level of alignment with reference texts in conveying information, while BLEU shows minor word deviations from references. The lower METEOR score might stem from alignment nuances in score calculation. Notably, the base model's test performance closely mirrors the results outlined in the original JointGT paper. The model that was trained on the combined dataset performed slightly worse for all three metrics than the model that was trained on the base WebNLG data. This can be explained by two considerations. First, it is evident in Table 6.2 that tests on FARO have very low performances. Secondly, the FARO dataset only accounts for a relatively small proportion in the combined dataset (Table 6.1). To better understand the reasons, a qualitative analysis is proposed in the next section.

6.4.2 Qualitative analysis

We examine instances from WebNLG and FARO datasets to analyze the base and combined model's performance. Observing Tables 6.3 and 6.4, the text generated by the combined dataset-trained model appears more semantically robust. The base model's generated text for FARO triples (Table 6.3, column *Base generated*) is notably brief, often mirroring the triples with semantic inaccuracies. Conversely, the combined model produces more coherent and accurate sentences in the same dataset (column *Combined generated*), maintaining triple direction. However, it's important to note that while the generated content respects triple order and semantic accuracy, it may still have limitations in altering the original label's content.

Table 6.3: Sample of the FARO test-set and the generated output of the base and combined model.

Triple	**Label**	**Base generated**	**Combined generated**
(offer, cause, reimburse)	(The directors said if Messrs. Drabinsky and Gottlieb mail an offer to shareholders by Nov. 22, it will reimburse them a maximum of C$8.5 million for expenses related to a bid.)	The cause of the offer is to reimburse .	The company has also announced that it will offer a new credit facility to small businesses, in an effort to reimburse them for the cost of capital expenditures.

Table 6.4: Sample of the WebNLG Test-set and the generated output of the base model.

Triple	**Label**	**Generated**
(3Arena, owner, Live Nation Entertainment), (Dublin, is part of, Republic of Ireland), (3Arena, location, Dublin), (Dublin, is part of, Leinster)	(The owner of 3Arena , Dublin , Leinster , Republic of Ireland is Live Nation Entertainment .), (Dublin is part of Leinster and a city in the Republic of Ireland . Dublin is also home to the 3Arena which is currently owned byLive Nation Entertainment.)	3Arena is located in Dublin , Leinster , Republic of Ireland and is owned by Live Nation Entertainment .

We also get a sight why the quantitative results are slightly worst for the combined model. The WebNLG data (Table 6.4) contains multiple triples per instance, giving more information about the text, and contains multiple labels. The FARO data (Table 6.3) contains only one triple per instance, together with one target sentence (label). Therefore, the model has less information about what to generate, and less chances to match the target label. Looking at the FARO input triples and the target label, it can be seen that the relationship (predicate) is often not explicitly represented by a particular word in the target sentence (implicit relation), making the evaluation with matching words harder. We provide additional insights in the appendix.

User Evaluation on ASRAEL To evaluate the system's performance, seven events from the ASRAEL dataset have been selected based on several criteria: values for the 4W properties, linking to a minimal number of articles, etc. The two largest (in terms of having the most articles) events in ASRAEL having all of the 4W properties are selected for evaluation: *"Operation Breaking Dawn"*, and *"2021 storming of the United States Capitol"*. The rationality behind this is to ensure that the information selection method is challenged by having an extensive amount of information to choose from. Among the remaining events in ASRAEL that include information about the place and time, five additional events are selected, bringing the total to seven.

The information selection method is used to select time, place, actor, and up to three mentions from the seven selected events. The base and combined models are used to generate text from the selected information. This information per event can be found in the appendix, together with the generated text. A manual evaluation was needed due to the absence reference text for automated metrics. Three annotators with a proficient level of English fluency determined which text was better for each event, by using either "win", "lose", or "tie", assessing *fluency* (grammatical correctness) and *adequacy* (correct integration of triples). This method aligns with the approach in [54]. Majority voting determined the winner or equality between models, followed by a non-parametric *sign test* at a significance level of $\alpha = 0.05$ to establish superiority. The non-parametric statistical sign-test is used to compare data. It assesses whether the median difference between observations differs significantly from zero, providing a p-value that indicates the probability of observing the given difference or a more extreme difference if the null hypothesis (no difference) were true. The significance level, denoted by alpha α, is a predetermined threshold set at 0.05, against which the p-value is compared to determine statistical significance. Results of this annotation are accessible in Table 6.5.

The combined model produces better fluent text than the base model in 71.4% of the cases. The non-parametric "sign test" was performed to measure a significant difference in the fluency of the text. With a p-value of 0.11, no significant difference was found. The same was

done to gauge the text's adequacy. With a p-value of 0.25, no significant difference was found.

Table 6.5: Fleiss' Kappa (κ) indicates perfect, and moderate agreement between annotators. The wins, losses, and ties when comparing the combined model against the base model are indicated in percentages. No model was significantly better than another with a significance level of 0.05.

Model	Fluency			κ	Adequacy			κ
	Win %	Lose %	Tie %		Win %	Lose %	Tie %	
Combined vs Base	71.4	14.3	14.3	1.0	28.6	0.0	71.4	0.6

Table 6.6: BLEU, METEOR, and ROUGE scores per model on the generated text from the article.

Model	**BLEU**	**METEOR**	**ROUGE**
Combined	0.1681	0.2081	0.3622
Base	0.1874	0.2273	0.3738

Table 6.7: Fleiss' Kappa (κ) indicates substantial agreement between annotators. The wins, losses, and ties when comparing the combined model against the base model are indicated in percentages. The combined model was significantly better than the base model in generating adequate sentences.

Model	Fluency			κ	Adequacy			κ
	Win %	Lose %	Tie %		Win %	Lose %	Tie %	
Combined vs Base	33.3	16.7	50.0	0.73	58.3	8.3	33.3	0.61

User Evaluation on an Manually Annotated Event To demonstrate whether the obtained results are consistent independently from the quality of the information extraction output, we decided to perform a user evaluation on a single article (sample), which has been manually annotated by handcrafting the resulting subgraph. This subgraph has been processed with both the combined and base model, and then evaluated using either “win", “lose", or “tie", in the same way as described in the previous section. The percentage of wins, losses and ties for the combined model, together with the Fleiss' kappa are reported in Table 6.7. The combined model has been assigned more wins for producing fluent and adequate text. The non-parametric “signed test" is applied to test if this is significant, again, with a significance level of 0.05. With a p-value of 0.34, no significant difference is found in generating more fluent texts between models. With a p-value of 0.04, a significant difference is found in generating more adequate sentences by the combined model, compared to the base model.

BLEU, METEOR, and ROUGE metrics have been computed using the sentences from the article as “reference label". These scores are detailed in Table 6.6. This illustrates that the base model performs slightly better than the model that was trained on the combined data. A reason for this could be formulated by looking at the generated texts, which can be found in

the appendix. More often than the combined model, the base model will output parts of the triple without taking the relationship between them into account. This will result in a badly formed sentence, but higher metrics, since more triples are incorporated. This is also reflected in the scores in Table 6.7, where the combined model is commonly noted for producing more fluent texts. Furthermore, the scores in Table 6.6 (computed on a single annotated article) are much lower then those computed on the whole WebNLG test set (Table 6.2). This outcome could be expected, considering that some of the triples extracted from the article are not, or to a limited extend, present in the original WebNLG data used to pre-train the JointGT model.

6.4.3 Conclusion

We studied in this chapter how to build complex narratives in the form of graphs of events, generating text with good level of complexity and semantic richness, expecting the system to generate answers beyond only *What* (the event), *Who* (the actor), *Where* (the location), and *When* (the time).

We enhanced the WebNLG dataset through the incorporation of the FARO dataset, aimed at refining the semantic depth of event relations. The expanded dataset now encompasses intricate relations including causality, prevention, intention, and enabling. From qualitative analysis, we can state that training on precise event relations produces more complete generated sentences, while no statistically significant difference was observed on fluency. Future work will experiment on more data to draw final conclusions. Our information selection from the graph focuses solely on the main event, disregarding pertinent details from interconnected events. Additionally, the data used for fine-tuning differs from the original dataset in terms of triple counts and instances, potentially impacting model evaluation.

Chapter 7

Fact Checking with Knowledge Graphs

In fact-checking, a common reason to reject a claim is the presence of erroneous cause-effect relationships between described events. Current automated fact-checking methods lack dedicated refined causal-based reasoning, potentially missing a valuable opportunity. In this section, we proposed a first approach to involve event relationships in a verdict predictor for fact checking. Our methodology combines an event relation extractor, semantic similarity computation and a set of if-then logical rules to detect contradictions or equivalences between the events in a claim and those in the evidence tested on two fact-checking datasets.

7.1 Related Work: Explainable Fact-Checking

Various end-to-end fact-checking systems have been developed. Hassan et al. [39] created a system that assesses claim veracity using keyword searches for evidence and knowledge bases, relying on traditional features like part-of-speech tags and sentiment for a 3-way classification experimenting with classifiers such as random forests and SVMs. In contrast, others employ deep neural networks (DNNs) for evidence selection and natural language inference, marking early examples of explainable fact-checking [108]. Popat et al. [84] use attention mechanism as a way to extract the most important words in an evidence as an explanation. The explanation gathered from the aforementioned ways is often not comprehensive [61]. Other methods use mining rules and knowledge graphs, which provide more comprehensive explanations, but that face coverage and scalability issues. Finally, other methods rely on summarization [10].

Models for textual entailment are widely used in verdict prediction [37]. These models are primarily deep learning black-box models, whose strong performance is counterbalanced by the lack of any explainability, making it difficult for humans to understand the logic behind the predictions. To address this problem, several works have been conducted in this area, focusing on attention mechanisms, summarization, or rule mining [61]. However, all of these lack precise causal explanations.

Recently, the *CheckWhy* dataset for fact-checking has been released [109]. This dataset includes 19,596 samples, mostly generated using generative models with some manual validation, from which 36% involve event causality. The dataset has been tested with LLMs, and its effectiveness in combination with any rule-based approach is yet to be proven. Causality in fact-checking has been also studied in [112], demonstrating the causal deductive reasoning capabilities of LLMs, while in [21], causal graphs and counterfactual reasoning is applied to the task. All these approaches are restricted to what we refer to as *direct causality*, or rely on loosely defined causal links, which often results in vague explanations.

7.2 Reasoning Rules

In this section, we introduce relation-based reasoning rules to deduce the most probable verdict for a given claim, knowing the evidence. These rules are intended to be found: 1. between events in the claim; 2. between events in each evidence; 3. between one event in the claim and one event in the evidence (or vice-versa).

Throughout this section, we use four placeholders – A, B, C, and D – to represent events or entities that can be related by "cause," "enable," "intend," , "prevent", or "no-relation" eventually. We will consider in the claim the events A and B and their relationship "$A \rightarrow B$". Similarly, we will consider in the evidence the events C and D and their relationship "$C \rightarrow D$". We also consider the degree of similarity between events and how that affects overall alignment or misalignment between a claim and its evidence. Below, we outline four key scenarios: *Logical Alignment*, *Logical Misalignment*, *Causal Loops*, and *Cherry-picking Scenarios*.

Logical alignment

This scenario is verified if the claim and the evidence include the same (or similar, or transitively-linked) events, which are also connected by the same relation. In the claim $A \rightarrow B$, and in the evidence $C \rightarrow D$. There is logical alignment if the relation is the same and:

- C is similar to A and D is similar to B;
- in absence of such similarities, we find a possible relation between A and C and/or between B and D which offer partial support by transitivity. In other words, while similarity between events provides a clear pathway to alignment, a direct causal connection can also strengthen the claim in cases where event similarity is not established.

If at least one of the previous cases is verified, we can conclude that the evidence *supports* the claim.

Example. In the following claim-evidence pair:
Claim: Sumo wrestler Toyozakura Toshiaki committed match-fixing, ending his career in 2011 that started in 1989.
Evidence: He was forced to retire in April 2011 after an investigation by the Japan Sumo Association found him guilty of match-fixing.

Let the events be denoted as follows: **Notation:**

A : *Committed match fixing*

B : *Ending*

C : *Investigation*

D : *Retire*

Relationships: From the claim and evidence, we have the following relationships:

1. $A \xrightarrow{\text{causes}} B$ (*Direct cause from the claim*)
2. $C \xrightarrow{\text{causes}} D$ (*Direct cause from the evidence*)
3. $A \xrightarrow{\text{causes}} C$ (*Direct cause linking claim and evidence*)
4. $B = D$ (*Equivalence between Retire and Ending from the evidence*)

Deduction: Using the relationships above, we deduce:

1. $A \xrightarrow{\text{causes}} C$ and $C \xrightarrow{\text{causes}} D \implies A \xrightarrow{\text{causes}} D$ (by transitivity)
2. $B = D \implies A \xrightarrow{\text{causes}} B$

Thus, $A \xrightarrow{\text{causes}} B$ is confirmed through transitivity (via evidence).

Conclusion: The evidence and the claim are logically consistent, demonstrating proper alignment. Figure 7.1 illustrates the above detailed scenario.

Logical Misalignment

In the claim $A \rightarrow B$, and in the evidence $C \rightarrow D$. The relation in the evidence (e.g. prevent) and the one in the claim (e.g. cause) are opposite. If we find a similarity matching (C is similar to A and D is similar to B), we can conclude a direct contradiction to the claim: the same event cannot both cause (or enable/intend) and prevent the same outcome, making the evidence more likely to *refute* the claim.

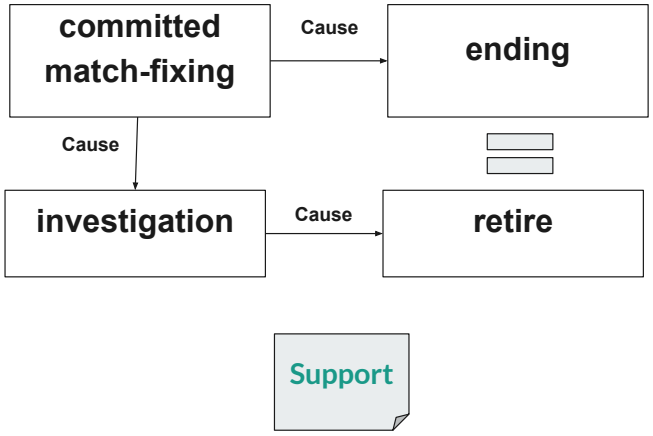

Figure 7.1: An example of a Logical Alignment.

Example. In the following claim-evidence pair:
Claim: Exercising daily causes significant muscle fatigue over time.
Evidence: Research shows that daily low-intensity exercise activates recovery mechanisms in the body, preventing the onset of chronic muscle fatigue and improving overall stamina instead. **Notation:** Let the events from the claim and evidence be represented as follows:
Notation:

A : *Exercising daily*

B : *Muscle fatigue*

C : *Activates recovery mechanisms*

D : *Prevents muscle fatigue*

Relationships: From the claim and evidence, we have the following:

1. $A \xrightarrow{\text{causes}} B$ (*Exercising causes muscle fatigue, from the claim*)

2. $A \xrightarrow{\text{causes}} C$ (*direct-cause between Exercising form the claim and activates recovery mechanisms, from the evidence*)

3. $C \xrightarrow{\text{prevents}} D$ (*prevention from the evidence*)

4. $B = D$

Deduction: Using the above relationships:

1. $A \xrightarrow{\text{causes}} C$
2. $C \xrightarrow{\text{prevents}} D \implies C \xrightarrow{\text{causes}} \neg D$
   $\implies A \xrightarrow{\text{causes}} \neg D \implies A \xrightarrow{\text{prevents}} D$
3. $A \xrightarrow{\text{causes}} B$ and simultaneously $A \xrightarrow{\text{prevents}} B$ (Contradiction).

Contradiction: A single event (A: *Exercising daily*) cannot simultaneously *cause* and *prevent* the same outcome (B: *Muscle fatigue*), leading to a logical inconsistency.

Conclusion: The claim and evidence are logically inconsistent, as the evidence contradicts the claim's assertion. This inconsistency is due to the conflicting causal and prevention relationships. Figure 7.2 illustrates the case.

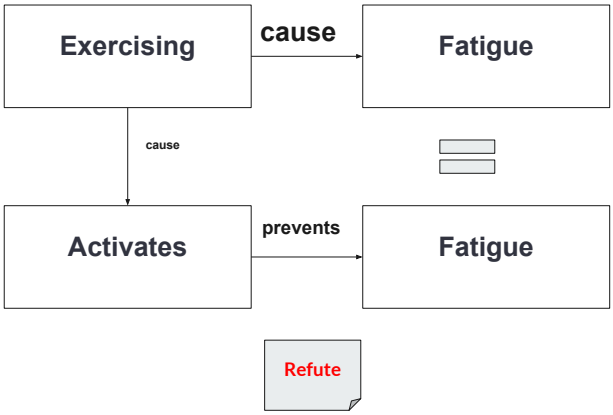

Figure 7.2: An example of a Logical Misalignment.

Causal loops

In this case, we check for a "closed causal loop" among four events A, B, C, and D by looking at the relationships (cause, enable, intend, or prevent) between each pair. We first take a claim ($A \rightarrow B$) and an evidence ($C \rightarrow D$) and infers how A might relate to C and how D might relate to B. If all four relationships form a consistent cycle (such as a chain of causes, enables, or intends), we have a closed causal loop, which implies an high probability that the evidence is **supporting** the claim. Since "prevent" is by definition considered the cause of not happening of another event, two consecutive "prevent" relations effectively become a "cause" because of the effect of a double negation. For example, if A prevents B and B prevents C, it is as though A causes C.

Example. In the following claim-evidence pair:
Claim: Poor infrastructure causes economic decline.
Evidence: transportation inefficiencies leads to supply chain disruptions and reduced economic activity

Notation:

A : *Poor infrastructure*

B : *Economic decline*

C : *Transportation inefficiencies*

D : *Supply chain disruptions*

Relationships: From the claim and evidence, we have the following:

1. $A \xrightarrow{\text{causes}} B$ (*Claim: Poor infrastructure causes economic decline*)
2. $A \xrightarrow{\text{causes}} C$ (*Evidence: Poor infrastructure causes transportation inefficiencies*)
3. $C \xrightarrow{\text{causes}} D$ (*Evidence: Transportation inefficiencies cause supply chain disruptions and reduced economic activity*)
4. $D \xrightarrow{\text{causes}} B$ (*Evidence: Supply chain disruptions and reduced economic activity cause economic decline*)

Deduction: Using the relationships above, we deduce:

1. $A \xrightarrow{\text{causes}} C$ and $C \xrightarrow{\text{causes}} D \implies A \xrightarrow{\text{causes}} D$
2. $A \xrightarrow{\text{causes}} D$ and $D \xrightarrow{\text{causes}} B \implies A \xrightarrow{\text{causes}} B$.

Conclusion: The claim $A \xrightarrow{\text{causes}} B$ is supported by the evidence through a causal loop.

Cherry-picking Scenarios

In this step, we look ***only*** at the evidence linked to a specific claim, ignoring the claim itself. We check for internal inconsistencies or selective usage of evidence, practice commonly addressed as "cherry-picking". Concretely, we group all evidence entries under the same claim, then compares each pair of evidence elements. Each piece of evidence is represented as a $\langle \text{sub}, \text{rel}, \text{obj} \rangle$ triple, where "sub" and "obj" are events or entities, and "rel" is the relationship

between them. The code measures how similar these events/entities are (e.g., sub_1 vs. sub_2, obj_1 vs. obj_2).

A claim is flagged for cherry-picking if certain patterns in the evidence emerge. For instance, it checks whether two pieces of evidence use the **same relationship** ($rel_1 = rel_2$) but involve subjects or objects that are dissimilar or opposites. If any of these mismatches is found, we deem the set of evidence potentially cherry-picked, because the evidence is either inconsistently presented or selectively used to reinforce the same relation in conflicting ways. Consequently, we assume here that the verdict is more probable to be cherry-picking.
Example. Here are two pieces of evidence for a given claim:
Evidence 1: *Frequent testing of the entire population would help identify so-called hidden hidden carriers individuals infected with SARS-CoV-2, the virus that causes Covid-19, but who have no symptoms of it. They seem to play an important role in the spread of Covid-19. Identifying these silent spreaders could help public health workers be more effective at contract tracing by identifying others who have been exposed and may require quarantine.*
Evidence 2: *Testing the entire population would undoubtedly identify a large number of such individuals, unnecessarily sidelining them from work and society.*
While the event *Testing* from the first evidence is equivalent to *Testing* from the second evidence, the second components of the relation differ significantly. Not only are they dissimilar, but they also have opposite polarities the first is positive (tracing), while the second is negative (sidelining). This discrepancy may raise concerns about a potential cherry-picking scenario. This is illustrated in Figure 7.3.

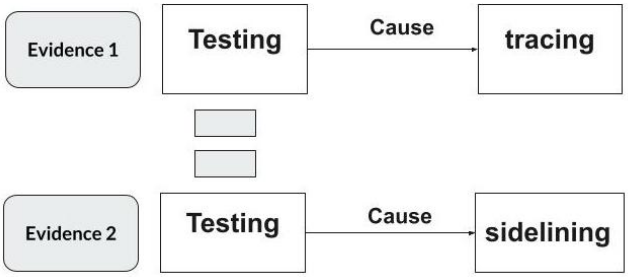

Figure 7.3: An example of a Cherry-Picking Scenarios.

7.3 Methodology - Reasoner Blocks

In this section, we present the methodology employed to predict the verdict of selected causally relevant claims. Our pipeline begins with event relation extraction conducted separately within the claim and evidence (Section 7.3.1). We also describe the approach used to extract refined causal relationships across claims and evidence (Section 7.3.2). Additionally, we detail our methods for distinguishing between similar, dissimilar, and opposite events using similarity

and dissimilarity measures to accurately establish connections between events (Section 7.3.3). These modules are then combined in our reasoning approach (Section 7.3.4).

7.3.1 Causality Extraction within the Claim/Evidence

Causal relation extraction is initially performed within a single context, either the claim or the evidence, without considering cross-context relations. To achieve this, we use our ERE models in inference mode. Since RoBERTa consistently outperformed other models for both RD and RC, we prioritize its predictions for these subtasks. On the other hand, we use REBEL for the EE task.

Example Input:
*The government's swift action to impose a lockdown **prevents** the rapid spread of COVID-19 among the population.*

Output:

- **RD = 1**,
- **RC = prevent**,
- `<triplet>` impose a lockdown `<subj>` spread of COVID-19 `<obj>` prevent

If the REBEL model produces a relation classification **RC** that differs from that of RoBERTa, we retain the prediction from RoBERTa, as it is the best-performing model according to our evaluation.

7.3.2 Causality extraction across Claim and Evidence

In this section, we describe the process of refining causality extraction between events across different contexts, specifically between those mentioned in the claim and the evidence. For the example, it would be valuable to identify the prevention relationship between the event **limit all nonessential interactions** in the evidence and the event **death** in the claim, as illustrated in Figure 7.4.

To accomplish the task described, we tested two different strategies, relying on Common Sense knowledge bases and Large Language models respectively. A key observation that motivated this approach is that event pairs across different contexts (e.g., between claim and evidence sentences) are often only weakly connected through the surrounding textual content. Instead, their underlying causal relationship is more easily inferred through commonsense reasoning. For instance, identifying that the evidence event *limit all nonessential interactions* prevents

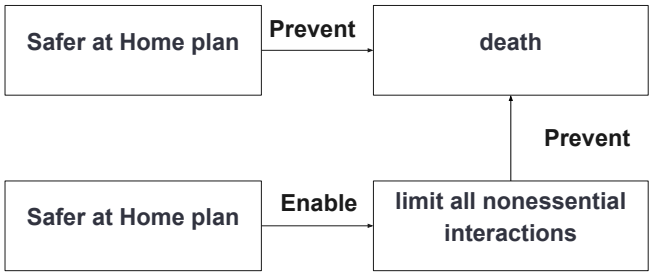

Figure 7.4: An example of Refined Causality Extraction between Events across the Claim and the Evidence

the claim event *death* requires understanding commonsense principles about public health interventions not surface-level cues from the text itself (see Figure 7.4).

Common Sense-based Causality Extraction

To support refined causality detection between events in claims and evidences, we leveraged structured commonsense knowledge as a foundation for training. Specifically, we used our curated commonsense dataset which includes both knowledge from ATOMIC and LLM-generated examples as a proxy training set to model implicit event relations grounded in everyday reasoning.

The goal was to train a classifier that takes as input a pair of events represented as short textual spans and predicts the most appropriate causal relation between them. These event pairs simulate the cross-context structure found in fact-checking: one event originates from a claim, and the other from supporting or refuting evidence. During training, the model learns to associate such event pairs with one of five target relations: *cause*, *enable*, *intend*, *prevent*, or *no_relation*.

Table 7.1 presents the size of the final commonsense dataset, split across training, development, and test sets.

Dataset	no_relation	intend	cause	enable	prevent	Total
Train	103,979	87,558	49,200	39,235	31,963	311,935
Dev	34,955	29,509	16,523	13,130	10,747	104,864
Test	34,354	29,521	16,519	13,120	10,744	104,258
Total dataset						**521,057**

Table 7.1: Overview of the employed Common Sense knowledge Base

Table 7.2 outlines the results obtained from the test set of commonsense shown in table 7.1. We can see that the system is accurately detecting the refined causal relations between the events of commonsense.

Class	Precision	Recall	F1-Score
cause	0.8248	0.8424	0.8335
intend	0.8523	0.8924	0.8719
prevent	0.9849	0.9929	0.9889
enable	0.9755	0.9776	0.9765
no_relation	0.8669	0.8208	0.8432

Table 7.2: Results of Causality Extraction between Claim Events and Evidence Events using Commonsense

LLMs causality extraction

LLMs are widely recognized for their strong capabilities in understanding human language and performing common-sense reasoning. Leveraging this strength, we explored prompting techniques to extract refined causal relationships between events in claims and evidence.

For this purpose, we utilized the LLM Phi-3-Medium-4K-Instruct, which demonstrates state-of-the-art performance in common-sense reasoning, while at the same time It requires fewer parameters and less computational effort [1]. The following prompt was employed in a few-shot manner to guide the model's output.

Event Relation Extraction Prompt

System: You are an expert in commonsense reasoning and refined causal relation extraction. Your task is to identify the most appropriate relation between two events. The possible relations are: `cause`, `intend`, `prevent`, `enable`, or `no-relation`. Only answer with one of these relation names no explanations.

Definitions:

- `cause`: Connects an event with its effect.
- `intend`: Connects an event with the effect it is intended to cause, regardless of outcome.
- `enable`: Connects a condition with the event it helps to bring about as an enabling factor.
- `prevent`: Connects an entity or event with another event that it causes *not* to happen.
- `no-relation`: No meaningful causal or intentional relation exists.

Examples:

- Q: What will be the relation between *earthquake* and *death*?
  A: `cause`
- Q: What will be the relation between *vaccination* and *infection*?
  A: `prevent`
- Q: What will be the relation between *signing a contract* and *starting the job*?
  A: `enable`
- Q: What will be the relation between *training hard* and *winning the race*?
  A: `intend`

Final Query:
Q: What will be the relation between {*event1*} and {*event2*}?
A:

To evaluate the performance of the LLM in extracting refined causality between events across claims and evidence, and since no ground-truth is available for this sake, we have performed qualitative assessment through manual evaluation. We sampled 40 examples from the dataset where the LLMs performed causality extraction. These examples, a mix of claims and evidence, were manually assessed for accuracy. Out of the 40 examples, 33 were correctly processed,

resulting in an accuracy of 82.5%.

7.3.3 Similarity, Dissimilarity, and Opposites

To determine if two events are the same, different, or opposites, we rely on sentence similarity, dissimilarity, and Polarity (Pol).

We evaluate two configurations: (1) Events only and (2) Events with context(event spans concatenated with the claim or evidence from which they originate). For cases where the events are an exact match (same surface form), we rely on the "Events only" configuration. However, when the events do not exactly match, we switch to the "Events with context" configuration, which incorporates additional context from the Claim/Evidence text. This enhancement allows for a more accurate reflection of event similarity.

To better explain this approach, we will make use of the 3 following examples.

Example 1

Claim: Dr. Qadir went on hunger strike when in prison and stopped when he could get his demands; then he was released from custody on January 25, 2006, as a result of **efforts by special envoy of the Austrian foreign ministry**, Gudrun Harrer, a journalist.
Evidence: He was released from custody on January 25, 2006, as a result of **efforts** by special envoy of the Austrian foreign ministry, Gudrun Harrer.

Example 2

Claim: Sumo wrestler Toyozakura Toshiaki committed match-fixing, **ending** his career in 2011 that started in 1989.
Evidence: He was forced to **retire** in April 2011 after an investigation by the Japan Sumo Association found him guilty of match-fixing.

Example 3

Claim: The **drought** caused severe crop failures in the region.
Evidence: Because of the prolonged dry **conditions**, agricultural yields in the area were dramatically lower than usual.

In these examples, we have an alignment between events in the claim and events in the evidence. For each case, the "Events with context" configuration produced the higher similarity score, as shown in Table 7.3. Based on empirically results on the entirety of use cases, we

set the threshold for event similarity to **0.54**, as events with similarity above this value are considered similar.

Input	Ex1	Ex2	Ex3
Events only	0.3235	0.2448	0.2589
Events + context	**0.7632**	**0.6709**	**0.6533**

Table 7.3: Cosine similarity for the three examples above.

Sometimes, events represent concepts that are simply *dissimilar* (indicated by a similarity score falling below a certain threshold). However, in other cases, they represent exact *opposites*. According to [24], **antonyms can be as similar or even more similar than synonyms, aside from their Pol**. This insight suggests that we can detect opposites by identifying pairs of events with high similarity but contrasting polarities.

To investigate this, we sampled five pairs of claims and corresponding evidence in which the statements contradict each other, and the events involved are opposites. We computed similarity and polarity under three configurations: 1. Evaluating events in isolation, 2. Evaluating whole claim text vs whole evidence text, and 3. Evaluating the full triple text (sub, rel, obj). We can discuss the following example, represented also in Figure 7.5:
Claim: *A study released on November 15, 2023, found that companies with strict dress codes experience a decline in employee morale.*
Evidence: *The Workplace Institute surveyed 150 firms with relaxed dress codes in October 2023 and found that employees reported a 25% higher job satisfaction rate than those at 100 companies enforcing formal attire. Notably, 72% of respondents from casual dress policy advocates felt more creative at work.*

We then determined which configuration best captured the correct polarity (e.g., one event positive, the other negative) alongside high similarity. The configuration that produced the correct opposing polarities with strong similarity scores was ultimately chosen to identify and confirm opposite relationships.

Similarity is computed as the cosine similarity between the embeddings of two input events obtained from SentenceBERT:

$$\text{Similarity} = \cos(\text{Embedding}_{\text{Event}_1}, \text{Embedding}_{\text{Event}_2})$$

For polarity detection, we utilize the DistilBERT base uncased model fine-tuned by Hugging-

face[1] for sentiment analysis, and reaching an accuracy of 91.3%. The model takes the input text and outputs a polarity classification: N (Negative) or P (Positive).

Table 7.4 demonstrates that evaluating the complete triple (*sub*, *rel*, *obj*) yields the hightest percentage of correct polarities, and the second highest similarity scores making it the most effective configuration among those considered.

Based on the empirically determined similarity threshold and the opposites check, which suggests that opposites are semantically similar but exhibit different polarities, we define the following rules. Given two text inputs T_1 and T_2:

- **Is-Similar**$(T_1, T_2) \implies$ Similarity$(T_1, T_2) >$ threshold **and** Pol$(T_1) =$ Pol(T_2) (PP or NN).
- **Is-Dissimilar**$(T_1, T_2) \implies$ Similarity$(T_1, T_2) <$ threshold.
- **Opposites**$(T_1, T_2) \implies$ Similarity$(T_1, T_2) >$ threshold **and** Pol$(T_1) \neq$ Pol(T_2) (PN or NP).

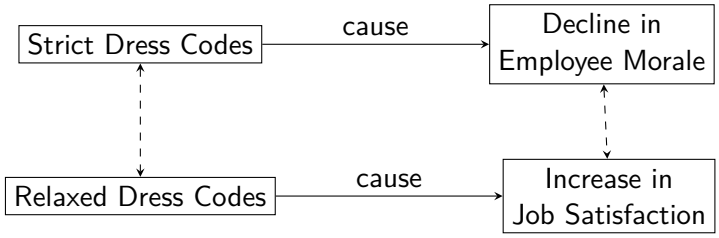

Figure 7.5: Two pairs of "opposite" triples illustrating cause relationships.

Comparison	Correct Polarity	Average Similarity
Claim vs Evidence	20%	**0.76**
Claim Event vs Evidence Event	50%	0.50
Triple Claim vs Triple Evidence	**80%**	0.61

Table 7.4: Comparison of correct polarity and average similarity across different levels of analysis.

7.3.4 Reasoning Approach

The reasoner analyzes the claim and evidence by checking the relationships between their events. It performs the following key tasks:

[1] https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english

1. **Causal Loop Check** (Figure 7.6) verifies if the events form a closed causal cycle, indicating support for the claim.

2. **Similarity and Relationship Check** (Figure 7.7) compares the relationships and similarities between events to determine alignment or contradiction.

3. **Cherry-picking Detection** (Figure 7.8) identifies inconsistencies or selective usage of evidence that may bias the verdict.

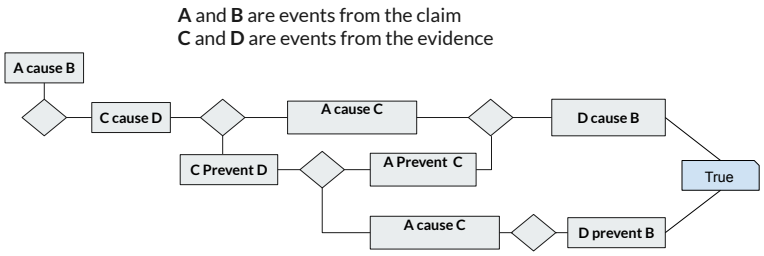

Figure 7.6: Implementation structure of the Causal Loop Check.

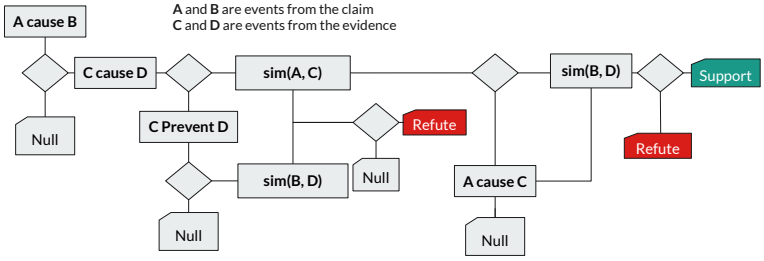

Figure 7.7: Implementation structure of a Similarity and Relationship check. (**sim** is referring to **Is-similar** shortened for visibility)

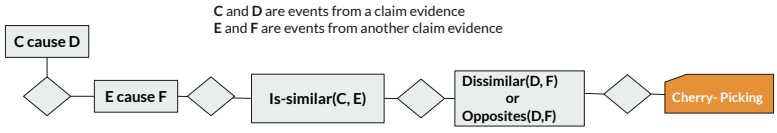

Figure 7.8: Implementation structure of a cherry picking scenario check.

7.4 Evaluation

This section outlines the evaluation datasets and the strategies used to assess our reasoning framework. We explain the filtering criteria applied to each dataset and the evaluation setups adopted for performance analysis.

7.4.1 Evaluation Datasets

To assess the effectiveness of our reasoning framework, we conducted evaluations on two widely used fact-checking datasets: AVERITEC and FEVEROUS. For AVERITEC, we retained only claims linked to informative textual answers, excluding boolean and unanswerable cases. In FEVEROUS, we filtered the data to include only claims supported fully by textual evidence.

Additionally, we constructed a manually curated subset from AVERITEC, ensuring that each instance contained at least one valid use case where the reasoner was expected to generate a verdict. For this subset, we applied minimal manual corrections to subject and object spans within claims and evidence to fix minor misalignments introduced by the EE system. Relation labels predicted by the ERE model were left unchanged. These adjustments aimed to simulate a cleaner extraction context limited to individual claims and evidence texts to ensure reliable inputs for subsequent reasoning steps, including similarity, polarity, and cross-event relation extraction. This setup enabled a more accurate assessment of the reasoner's performance under improved semantic conditions.

AVeriTeC

The AVERITEC dataset consists of 4,568 real-world claims, each paired with question–answer evidence and textual justifications used to determine verdicts. Every item is annotated with one of four verdict labels:

- **Supported:** The evidence fully supports the claim.
- **Refuted:** The evidence directly contradicts the claim.
- **Conflicting Evidence (Cherry-picking):** The evidence presents conflicting information neither fully supporting nor fully refuting the claim.
- **Not Enough Evidence**: The evidence is insufficient to make a conclusive judgment about the claim s veracity.

In our evaluation, we use the training subset of AVERITEC, as it is aligned with ground truth labels that enable quantitative assessment.

To prepare the AVERITEC dataset for evaluation, we first excluded claims answered with either a boolean or unanswerable response, retaining only those with extractive or abstractive answers. We also exclude the *Not Enough Evidence* portion of the dataset, since our reasoning system is not designed to produce this type of verdict.

FEVEROUS

The FEVEROUS dataset [5] (Fact Extraction and VERification Over Unstructured and Structured information) contains 87,026 claims annotated with evidence sourced from Wikipedia. Due to time and resource constraints, we randomly selected a subset of 4,392 claims for our experiments, assuming this sample size to be reasonably sufficient for evaluating the reasoner's performance.

Each claim is labelled as *supports*, *refutes*, or *not enough info*. Evidence may include textual sentences or table cells, along with annotator metadata (e.g., query actions, page clicks, evidence types). For our experiments, we retained only claims supported by fully textual evidence and excluded those referencing table cells, yielding a filtered subset suitable for text-only based reasoning. We Also exclude the *NOT ENOUGH INFO* part since it is not handled by the reasoner. We then filtered out claims where was no causal relation, retaining a total of 1183 claims for evaluation. These were distributed across two verifiable categories: 705 *SUPPORTS* and 478 *REFUTES*.

Reasoner Specific Subset (RSS)

Our reasoning framework is based on a set of rule-based mechanisms operating over event relations, similarity, and polarity. While this approach allows for explicit and interpretable inference, it does not guarantee coverage of all examples within the datasets. This limitation applies to both AVERITEC and FEVEROUS, as not all claim–evidence pairs contain use cases compatible with the system's reasoning rules.

To ensure a fair evaluation, we constructed a controlled subset consisting of claim–evidence pairs that contain verified reasoning use cases. While we do not assume that the reasoner will always respond correctly since the final outcome also depends on other components such as similarity scoring and polarity detection we ensure that a valid use case is present. This setup allows us to meaningfully assess the reasoner's behaviour in cases where its mechanisms should, in principle, be activated, and helps isolate genuine reasoning failures from issues caused by the absence of applicable use cases.

The creation process involved the following steps:

- We randomly sampled 765 claims from the AVERITEC dataset.
- For each claim–evidence pair, we first checked whether it contained a valid use case for reasoning (e.g., causal loops or contradictions).
- If a valid use case was found, we then inspected the event representations extracted independently from the claim and the evidence texts using our ERE models.

- We manually corrected misaligned subject and object spans, as these event structures are the foundation for all subsequent reasoning operations, including cross-sentence event relation extraction between the claim and evidence, similarity computation, and polarity detection.

Due to the significant manual effort required, we limited this process to one dataset. Without any particular preference between AVERITEC and FEVEROUS, we selected AVERITEC based on practical constraints.

The final benchmark consists of 86 validated (claim, evidence) pairs across 60 unique claims.

Table 7.5 shows the stats for each dataset after filtering.

Filtering Step	**AVERITEC**	**FEVEROUS**
Total unique claims	2998	1736
Total answers / total evidences	8479	3836
Answer Type Distribution (AVeriTeC)		
Extractive	4571	–
Abstractive	2225	–
Boolean	1297	–
Unanswerable	386	–
Claims excluding Boolean and Unanswerable types	2783	1736
Claims with no relation	850	44
Claims after excluding claim with not enough evidence	1759	1183
Label Distribution		
Refuted / REFUTES	1066	478
Supported / SUPPORTS	581	705
Conflicting Evidence / Cherrypicking	112	–

Table 7.5: Filtering steps and label distributions for AVERITEC and FEVEROUS datasets before running the reasoning pipeline.

7.4.2 Evaluation stategy

In our evaluation, we define two distinct configurations for computing performance metrics:

- **Configuration 1 – Tolerant:** This configuration is applied in the fully automatic setting for both AVERITEC and FEVEROUS. It adopts a lenient evaluation strategy, focusing only on the system's performance when it chooses to respond.

  - **Recall** is defined as the proportion of cases where the reasoner successfully produces a verdict (i.e., does not abstain), relative to the total number of evaluated cases.
  - **Precision** measures the proportion of correct verdicts among those that were actually produced (i.e., abstentions are excluded).
  - **Abstentions** (*None* outputs) are excluded from metric computation. This is because they may arise from either genuinely irrelevant inputs or from missed reasoning opportunities due to limitations in similarity matching or claim–evidence alignment. Penalizing such cases equally would not accurately reflect the system's reasoning capability.

- **Configuration 2 – Strict:** This configuration uses a more rigorous evaluation policy. It is primarily applied to the manually verified subset of AVERITEC, where each example has been checked to ensure that it contains a reasoning-relevant claim–evidence pair. We manually evaluated 765 claim, in which we could select 60 claims that contains at least one of the selected use cases that we have discussed earlier that mean the system is ideally expected to react and answer correctly, of course is till depend on the similarity calculation, polarity, and relations across claim and relation events.

  - The system is expected to always produce a verdict. Every abstention (*None*) is treated as a **false negative (FN)**, thus reducing recall.
  - **Recall** is computed as the number of correct verdicts over the total number of evaluated cases, including abstentions.

  For the automatic versions of AVERITEC and FEVEROUS, we apply this configuration across all examples, including those where the model abstains from producing a verdict. Every abstention is treated as a false negative, **regardless of whether the inputs represent a valid use case or not**. While this may appear stringent, it allows us to assess the model's robustness under challenging and ambiguous scenarios, acknowledging that a complete manual evaluation of all use cases is infeasible.

  The metrics are computed as follows:

  - **True Positives (TP) and True Negatives (TN):** Cases where the predicted verdict matches the gold label.
  - **False Positives (FP):** Cases where the system produces an incorrect verdict.
  - **False Negatives (FN):** Cases where the system abstains from providing a verdict.

This formulation allows us to evaluate the effectiveness of the reasoner both in terms of its ability to provide answers (recall) and the accuracy of those answers (precision).

7.5 Discussion

Table 7.6 summarizes the performance of our reasoning framework across different configurations, datasets, and knowledge sources. We evaluate both the cases where the reasoner was dependent on the LLM or the model trained ion commonsense for inferring the relation between events across the claim and evidence triples on the AVERITEC and FEVEROUS

Test Set	Knowledge Source	Precision (P)	Recall (R)	F1-Score
RSS †	LLMs	0.55	0.45	0.5
	Common Sense	0.51	0.45	0.48
AVERITEC (Strict)	LLMs	0.48	0.19	0.27
	Common Sense	0.54	0.2	0.29
AVERITEC (Tolerant)	LLMs	0.47	0.35	0.4
	Common Sense	0.52	0.37	0.43
FEVEROUS(Strict)	LLMs	0.5	0.44	0.47
	Common Sense	0.51	0.44	0.47
FEVEROUS(Tolerant)	LLMs	0.52	0.62	0.56
	Common Sense	0.52	0.62	0.56

Table 7.6: Precision, recall, and F1-Score for each knowledge source across the different evaluation datasets. † RSS refers to the Reasoner-Specific Subset, composed exclusively of validated use cases; tolerant evaluation was unnecessary as all examples are guaranteed to trigger reasoning.

On the reasoner specific subset of, the system achieves an F1-score of 0.50 with LLMs and 0.48 with common-sense knowledge bases ERE. The over-performance of LLMs based reasoning is very slight. In 50% of the cases the model either does not provide an answer or the answer is wrong. This can be due to wrong relation between events across the claim, evidence, wrong polarity, or similarity calculations.

Across the non manually selected bases of both AVERITEC and FEVEROUS, we observe significant variance between strict and tolerant settings:

- In the **strict configuration**, recall drops significantly on AVERITEC for both sources. This is expected, as abstentions (*None*) are penalized as false negatives anyway.
- In contrast, the **tolerant configuration** presents a more optimistic perspective. When the system chooses to respond, its predictions are generally correct. This is evident in the improved F1-scores.

On FEVEROUS, the performance is more balanced across both reasoners regardless of the model used for extracting cross claim-evidence relations. Under the strict setting, both achieve

an F1-score of 0.47. In the tolerant setting, this increases to 0.56 for both, with recall peaking at 0.62. This highlights that FEVEROUS may contain more concrete, fact-based pairs that are easier to match and reason over.

7.6 Conclusion

In this chapter, we integrated causal reasoning into automated fact-checking. By leveraging semantically refined event relationships and reasoning rules, our system addresses the limitations of existing approaches that often lack causal interpretability. The dual evaluation of LLM relations-based and common sense bases causal reasoning demonstrates the system's potentials in serving explainable fact-checking.

While the reasoner achieves around 50% f1-score, its strength lies not in outperforming existing models in raw metrics, but in offering structured, interpretable justifications for verdicts when valid causal use cases are present. Rather than functioning as a standalone predictor, the system is best positioned as a complementary layer that can enhance existing fact-checking models. By surfacing explicit causal links, logical inconsistencies, and polarity mismatches, it provides interpretable insights that can either support or question the decisions made by black-box veracity classifiers.

Chapter 8

Conclusions

8.1 Summary of the Research

In my thesis, the work focuses on the automatic extraction of refined causal relationships between events from text, with applications in narrative generation and explainable fact-checking. We introduced the *Facts and Events Relationship Ontology (FARO)*, a novel data model specifically developed to represent event relationships within knowledge graphs. FARO provides a semantically-rich structure that allows users to traverse event flows effectively, either retrospectively to identify causes and conditions, prospectively to explore consequences, or through various other semantic connections. By explicitly encoding these relationships, FARO enables the construction of interconnected event timelines.

As a foundational contribution, we first conducted an extensive comparative analysis of existing partially overlapping event relationship models. This systematic literature review identified gaps and redundancies in the existing approaches, ultimately informing the precise set of relationships captured by FARO. Building upon this ontology, we developed an enriched event relation dataset that extends beyond existing benchmarks such as TimeBank and EventCausality. Our dataset uniquely integrates causal and contingent relationships, including refined relation types such as *Cause*, *Intend*, *Prevent*, *Enable*, and *Not Cause*. To address common challenges such as class imbalance and limited relation coverage, we combined semi-automatic annotation, rigorous manual validation, and synthetic data generation leveraging LLM. Furthermore, incorporating commonsense knowledge from ATOMIC has resulted in a robust and comprehensive knowledge graph dataset, facilitating deeper research in narrative generation, causal event extraction, and automated fact-checking.

Subsequently, we conducted an in-depth investigation of the event relation extraction task, with a particular focus on fine-grained causal relations. We introduced the first model capable of extracting these refined causal links from text, achieving an average F1-score of 0.763 across relation detection, relation classification, and event extraction.

Several key findings emerged from our analysis: end-to-end models generally outperform sequential approaches, especially in relation detection and classification; the integration of commonsense knowledge consistently boosts extraction performance; RoBERTa exhibits superior effectiveness among tested PLMs; and despite progress, event extraction remains inherently challenging. Additionally, while LLM demonstrated promising potential in few-shot scenarios, they have yet to match the performance of fine-tuned PLM like RoBERTa and BERT.

We also demonstrated practical applicability by developing a publicly accessible pipeline enabling users to perform event relation extraction interactively.

We have explored the possible downstream application of our model such as complex narrative generation and explainable automated fact-checking.

In the narrative context, we enhanced the WebNLG dataset by integrating the FARO model, enriching semantic expressiveness of generated texts. Preliminary qualitative analysis indicated that training on refined event relations notably improves completeness of generated narratives, although fluency remained largely unaffected.

In the context of explainable fact-checking, we introduced a novel system that integrates semantically refined event relations with explicit causal reasoning rules, addressing the interpretability gap in existing approaches. Our reasoning framework achieved an F1-score of 50% on a manually verified subset containing confirmed reasoning use cases, and demonstrated comparably strong performance on broader datasets where such use cases were not explicitly annotated. These results provide a solid foundation for leveraging fine-grained causal extraction in fact-checking tasks and establish a competitive baseline for future improvements in explainable reasoning systems.

8.2 Future Work

Looking ahead, this thesis envisions future work across three complementary themes: **Knowledge Engineering with LLMs**, **Event Relation Knowledge Graph Construction**, and **Downstream Applications**. Each area will be developed as follows.

Knowledge Engineering with LLMs Building on the observed potential of LLMs for ontology-driven tasks, future work will aim to deepen our understanding of the factors that influence LLM performance in this context:

- *Ontology Characteristics*: Investigate how different ontology structures and vocabularies affect the quality and accuracy of LLM-generated outputs, especially in specialized domains.

- *Competency Question (CQ) Design*: Study the impact of CQ formulation strategies and the influence of property features—such as names, descriptions, and logic—on LLM responses.
- *Ontology Reuse and Complexity*: Evaluate how well LLMs handle ontologies that incorporate or extend existing data models.
- *Broader Ontology Coverage*: Extend our analysis to include ontologies from datasets like CQ2SPARQLOWL [85] and SILKNOW [102], to generalize findings.
- *Human vs. LLM Performance*: Compare CQs generated by LLMs with those produced by domain experts unfamiliar with the data model.
- *Refined Evaluation Protocols*: Improve CQ quality control by filtering redundant or irrelevant queries and refining performance metrics.

Event Relation Knowledge Graph Construction In the domain of knowledge graph construction and refinement, several technical directions will be pursued:

- *Coreference Resolution*: Enhance internal consistency by resolving event coreferences and aligning event mentions across contexts.
- *Entity Linking*: Connect extracted events and relations to external knowledge bases such as EventKG [34] and Wikidata [123].
- *Graph Validation Techniques*: Apply link prediction and deletion techniques to assess and improve reliability and completeness.

Downstream Applications For the two applications we have discussed—narrative generation and explainable fact-checking—future work will focus on extending and refining the utility of the proposed ERE system within these contexts.

- **Narrative Generation**:
  - Improve event clustering by identifying sub-events and relations at the document level.
  - Use NLP-based data augmentation to enhance dataset quality and coverage.
- **Explainable Fact-Checking**:

- Integrate external knowledge bases (e.g., Wikidata, domain-specific ontologies) into the reasoning layer.
- Expand the evaluation dataset using high-quality claim-evidence pairs, including data from *CheckWhy* [109] and potential crowdsourced annotations.

8.3 Future Research Directions

Building upon the foundation established in this thesis, several promising avenues for future research emerge. These directions aim to extend the semantic depth, contextual reach, and real-world applicability of event-based KGs.

Commonsense and Implicit Event Linking for Knowledge Graph Updating

In real-world applications such as journalism or public health surveillance, it is crucial for KGs to dynamically adapt to new information. We propose developing a continual learning pipeline that leverages both commonsense priors and document context to match incoming news events with existing nodes in the KG, even when surface features differ. This would include linking co-referential or semantically overlapping events, even across heterogeneous sources. Techniques such as event coreference resolution [139] and contrastive learning for event matching [38] can be leveraged to enhance this pipeline. Another potential can be improving our refined causlaity extraction model to be able to do cross paragraph or even cross documnets relations. This aligns with recent efforts in discourse-level event extraction and narrative coherence modelling [22, 133]. A hybrid approach combining fine-tuned PLMs as knowledge-augmented transformers (e.g., using our commonsense dataset as base) with the natural language understanding power of LLMs could help infer implicit causality across distant events.

Incorporating event "status" (e.g., *planned, happened, did not happen*) would support both temporal and semantic updates, enabling the continuous evolution and refinement of the knowledge graph [120]. Such modelling could further help identify recurring patterns that often govern transitions between event states—revealing, for example, common precursors that increase the likelihood of a planned event materializing or being cancelled.

Multimodal Event Representation and Enrichment

Text alone may not capture the full context of events—news often includes images or videos that provide additional cues. Future work could investigate multimodal representation learning techniques [65, 137] to encode visual and textual signals jointly. This would be particularly useful for detecting visually anchored causal narratives (e.g., protest imagery linked with

reported causes or consequences) and for building richer KG representations.

User-Driven KG Refinement and Browser-Based Lightweight Models

We propose the development of a browser-based KG exploration interface that embeds a lightweight version of our causal relation extraction model (e.g., using distilled RoBERTa or quantized transformer variants). This would allow users to highlight causal chains in real-time while navigating news articles. Integrated user feedback mechanisms could annotate new event links or flag errors, feeding back into an active learning loop for KG improvement. This aligns with research in interactive machine learning and human-in-the-loop KG construction [7].

Detection of Logical Fallacies and Causal Loops of Browsed News Articles

A particularly impactful application of our system was the detection of logical inconsistencies or misleading causal narratives in fact-checking tasks. But this was restricted to only researcher use on a small dataset. We propose using the updated KG to automatically identify causal loops, contradictions, or spurious correlations (e.g., post hoc fallacies) also as a browser extension, can be particularly useful on fast-paced platforms like Twitter or Reddit. The challenge here will be minimizing the complexity of the reasoner to ensure real-time analysis, lightweight processing, and seamless integration without disrupting user experience.

By extending the current contributions into these directions, we envision a next-generation causal knowledge platform that is multimodal, continuously evolving, and socially grounded. This system could serve not only academic purposes but also support journalistic integrity, policy modelling, and public education.

Publications List

The research carried out during this reporting period has led to the publication of the following scientific papers:

Conference Proceedings

1. Rebboud, Youssra; Lisena, Pasquale; Troncy, Raphaël. **Beyond causality: Representing event relations in knowledge graphs**. In *EKAW 2022, 23rd International Conference on Knowledge Engineering and Knowledge Management*, 26-29 September 2022, Bolzano, Italy.

2. Rebboud, Youssra; Lisena, Pasquale; Troncy, Raphaël. **Prompt-based data augmentation for semantically-precise event relation classification**. In *SEMMES 2023, Semantic Methods for Events and Stories*, May 23-28, 2023, Heraklion, Greece.

3. de Kok, Mike; Rebboud, Youssra; Lisena, Pasquale; Troncy Raphaël; Tiddi, Ilaria. **From nodes to narratives: A knowledge graph-based storytelling approach**. In *TEXT2STORY 2024, 7th International Workshop on Narrative Extraction from Texts (Text2Story)*, colocated with ECIR 2024, 24 March 2024, Glasgow, UK.

4. Rebboud, Youssra; Tailhardat, Lionel; Lisena, Pasquale; Troncy, Raphaël. **Can LLMs generate competency questions?**. In *ESWC 2024, Extended Semantic Web Conference, Special Track on Large Language Models for Knowledge Engineering*, 26-30 May 2024, Hersonissos, Greece.

5. Rebboud, Youssra; Lisena, Pasquale; Tailhardat, Lionel; Troncy, Raphaël. **Benchmarking LLM-based ontology conceptualization: A proposal**. In *ISWC 2024, 23rd International Semantic Web Conference*, 11-15 November 2024, Baltimore, USA.

6. Flores, Gustavo Miguel; Rebboud, Youssra; Pasquale Lisena; Troncy, Raphaël. **Streamlining Event Relation Extraction: A Pipeline Leveraging Pretrained and Large Language Models for Inference**. In *24th International Conference on Knowledge Engineering and*

Knowledge Management, Poster & Demo Track, Amsterdam, Netherlands, November 26-28, 2024.

To be submitted

1. Rebboud, Youssra; Lisena, Pasquale; Troncy, Raphaël. **Integrating Causal Reasoning into Automated Fact-Checking**.

2. Rebboud, Youssra; Lisena, Pasquale; Troncy, Raphaël. **Leveraging Common Sense Knowledge and LLMs for Joint Event Extraction and Relation Classification**. .

Résumé en français

8.1 Introduction

Cette thèse traite du défi scientifique que constitue l'extraction précise des flux événementiels à partir de données textuelles, cruciale pour la prise de décision, l'analyse historique et la modélisation prédictive.

Pour commencer, nous présentons FARO (Facts and Event Relations Ontology), une ontologie innovante permettant de structurer 25 relations distinctes entre événements et faits, facilitant ainsi des représentations sémantiques enrichies et une meilleure compréhension des interactions événementielles.

Dans un premier temps, nous abordons les modèles de données existants, en analysant les ontologies et base de données actuels dédiés aux relations entre les événements. Cette thèse critique révèle les limites des modèles de données actuels, notamment leur focalisation sur des relations entre évènements générales telles que la causalité simple ou les relations temporelles, laissant de côté des interactions plus raffinées.

Ensuite, nous détaillons la conception de l'ontologie FARO. Nous expliquons le choix des classes principales telles que « Évènement » et « Condition », et leur regroupement sous la classe abstraite « Relata ». FARO intègre des relations plus sophistiquées, notamment la causalité directe, la facilitation, la prévention, l'intention, et des relations comparatives comme l'opposition et la similarité. Ces relations enrichies permettent une meilleure modélisation des flux événementiels complexes et une exploration plus fine des liens entre événements.

Pour faciliter l'extraction robuste des relations événementielles raffinées, nous avons constitué une base de données annoté de manière novatrice, comprenant plus de 500 000 phrases. Ce corpus, enrichi par l'utilisation de grands modèles de langage (LLMs), d'intelligence artificielle générative et de connaissances de bon sense issues du graphe de connaissances ATOMIC, couvre des relations telles que la causalité directe, la facilitation, la prévention et l'intention. Nous détaillons le processus rigoureux d'annotation manuelle et semi-automatique, ainsi que les méthodes d'augmentation des données, notamment via des techniques de prompt

engineering et l'intégration de connaissances de bon sens.

Nous développons ensuite un modèle avancé d'extraction capable d'identifier avec précision ces relations causales fines. Notre approche comprend l'évaluation comparative de plusieurs stratégies, notamment l'extraction causale à grain fin en tant que pipeline complet ou en sous-tâches séparées, et l'intégration de connaissances de bon sens dans les modèles d'extraction.

Notre travail inclut également le développement d'une plateforme interactive basée sur Streamlit, permettant aux utilisateurs d'expérimenter différents modèles et configurations pour l'extraction des relations événementielles à partir de textes en temps réel. Cette plateforme offre une visualisation intuitive des résultats d'extraction et une évaluation qualitative facilitée pour les chercheurs et utilisateurs généraux.

Nous validons enfin l'efficacité pratique de notre approche par deux applications concrètes :

- Génération narrative améliorée en utilisant des graphes de connaissances structurés enrichis par des relations causales fines, nous générons des récits plus cohérents et sémantiquement riches. Cette approche est évaluée qualitativement et quantitativement, montrant une amélioration significative en comparaison des modèles existants basés uniquement sur des relations événementielles générales.
- Vérification automatique des faits: nous présentons un système innovant qui exploite les relations causales fines pour améliorer la prédiction du verdict et fournir des explications compréhensibles par l'homme sur les contradictions éventuelles entre revendications et preuves. Notre système, utilisant un raisonnement causal explicite, démontre une capacité accrue à détecter les contradictions logiques, améliorant ainsi la transparence et la fiabilité des systèmes de fact-checking.

En conclusion, cette thèse propose une méthodologie complète allant de la modélisation ontologique à l'extraction et à l'application des relations événementielles raffinées, ouvrant des perspectives prometteuses pour des recherches futures dans l'analyse sémantique des événements, la génération narrative et la vérification automatisée des faits.

8.2 Modèle des Données

Ce chapitre propose un modèle sémantique pour la représentation des relations entre événements, un aspect crucial mais souvent sous-exploité dans les graphes de connaissances. Alors que les relations temporelles et causales ont été largement étudiées, d'autres types de relations comme la corrélation, l'intention, la concession ou encore la comparaison restent peu modélisées dans les ressources existantes.

Nous présentons **FARO** (Facts and Events Relationship Ontology), une ontologie conçue pour représenter jusqu'à 25 types de relations distinctes entre événements et conditions, réparties en quatre catégories principales : temporelles, méréologiques, contingentes et comparatives. Contrairement à d'autres modèles, FARO intègre une hiérarchie des relations et des contraintes logiques (e.g. disjonction, transitivité), facilitant l'inférence sémantique.

Le chapitre s'ouvre sur un état de l'art détaillé des modèles existants (ontologies et jeux de données), mettant en évidence leurs limites. Il en ressort que les approches existantes ne permettent pas de capturer l'ensemble des relations inter-événementielles de manière homogène. En comparaison, FARO propose une couverture plus complète et un design modulaire facilitant son intégration dans des environnements sémantiques.

Nous discutons ensuite des choix de modélisation de FARO, qui repose sur deux classes principales (`Event` et `Condition`) regroupées sous `Relata`, et permet de représenter l'état d'un événement (réalisé, non réalisé, potentiel ou planifié). Des exemples tirés de textes illustrent la richesse des relations couvertes, y compris celles au-delà de la causalité simple (e.g. `enables`, `prevents`, `not cause`, `correlates-with`, etc.).

Enfin, nous montrons comment FARO peut contribuer à la construction de graphes d'événements mieux connectés et sémantiquement enrichis, facilitant la navigation dans les flux d'événements et l'inférence automatique. FARO est disponible publiquement en OWL[1].

8.3 Base des Données

e chapitre présente la construction d'un jeu de données de relations entre événements destiné à améliorer l'extraction automatique de relations causales fines. Il se concentre sur cinq types de relations issues de l'ontologie FARO : *Cause*, *Intend*, *Prevent*, *Enable*, et *Not Cause*.

8.3.1 Sources et annotation

Deux ressources existantes ont été réutilisées : TimeBank et EventCausality, toutes deux au format TimeML. Une nouvelle balise `RLINK` a été introduite pour généraliser les liens entre événements, avec quatre attributs principaux : identifiant de lien, type de relation, identifiant de l'événement source, et identifiant de l'événement cible.

Un processus semi-automatique a été employé pour générer des paires candidates, basé sur des mots-signaux spécifiques à chaque type de relation. Une annotation manuelle a ensuite été réalisée par deux annotateurs, avec un accord inter-annotateur (Cohen's kappa) de 0.7112. La première phase a permis d'obtenir le nombre suivant de relations annotées correctement :

[1] https://purl.org/faro/

Relation	Cause	Intend	Prevent	Enable	Not-Cause
Nombre de relations	283	44	13	18	3

Pour pallier l'importante imbalance, des articles AFP (Agence France-Presse) ont été exploités. L'annotation manuelle de ces nouveaux articles a permis d'augmenter à 81 relations de type *Prevent* et 100 de type *Enable*, avec les nouvelles statistiques suivantes :

Relation	Cause	Intend	Prevent	Enable	Not-Cause
Nombre final	283	44	81	100	3

8.3.2 Augmentation par LLM

Pour élargir davantage le corpus, des modèles de langage (GPT-3.5 `text-davinci-003`) ont été utilisés pour générer 600 phrases par type de relation sous-représentée. Les prompts incluaient : (i) la définition selon FARO, (ii) des exemples extraits du jeu de données, et (iii) une requête spécifique selon le type de relation.

Un exemple de prompt : *Give me very long political example sentences following these examples and give me each sentence in one line.* Les phrases générées ont été validées manuellement, avec un taux de précision de 90.77% pour les phrases, 75.15% pour le premier événement, et 66.82% pour le second :

Relation	Phrases correctes	ET1 correct	ET2 correct
Intend	93.82%	75.13%	73.47%
Prevent	97%	81.83%	77%
Enable	81.5%	68.5%	50%

Le jeu de données final contient 1228 phrases synthétiques supplémentaires, menant à un total de 1891 phrases après dé-duplication.

8.3.3 Test set et évaluation

Le jeu de test initial a été nettoyé via Similarité Cosine avec SentenceBERT (seuil de 90%). Pour le compléter, 216 phrases avec des relations fines ont été extraites manuellement depuis le jeu AVeriTeC.

Statistiques finales du jeu augmenté

Relation	Original	Augmenté
Cause	268	268
Intend	42	459
Prevent	81	500
Enable	100	450
No-Relation	172	172
Total	663	1849

8.3.4 Ajout de Connaissances de Bon Sens

Des triplets provenant d'ATOMIC (ex. *xIntent*, *xWant*, *xEffect*) ont été intégrés et alignés avec les relations FARO, en particulier pour *Cause* et *Intend*. Pour combler les lacunes sur *Enable* et *Prevent*, de nouvelles phrases ont été générées par LLMs comme Zephyr et Truthful-DPO.

Des exemples négatifs ont été introduits via permutation des sujets et objets des relations existantes. Au total, les données combinées (réelles et synthétiques) couvrent plus de 526,000 phrases :

Source	Total Phrases	Relations incluses
CNC (nouvelles)	3,316	Cause, No-Relation
ATOMIC (CS)	315,173	Cause, Intend, No-Rel
LLMs (CS synth.)	205,884	Prevent, Enable, No-Rel

8.3.5 Graphe de Connaissances

Un graphe de connaissances a été construit pour représenter les relations entre événements, suivant les principes FAIR. Les phrases sont modélisées avec NIF, les relations avec l'ontologie FARO, et la provenance via le vocabulaire PROV-O (provenance humaine, AFP, ATOMIC ou génération LLM).

Chaque phrase est reliée à une source (dataset, publication, ou agent logiciel) à travers une activité de provenance contenant le titre, les auteurs et l'année.

8.3.6 Conclusion

Ce chapitre propose un pipeline complet pour construire un jeu de données annoté en relations d'événements, enrichi par LLMs et connaissances de bon sens. Il introduit un format unifié de représentation des relations (`RLINK`) et une modélisation RDF permettant l'utilisation en raisonnement causale, vérification automatique de faits et génération narrative. Ce travail offre une base solide pour l'évaluation et l'amélioration des systèmes d'extraction de relations événementielles.

8.4 Extraction des Relations entre Événements à partir de Textes

Ce chapitre explore l'extraction fine de relations causales entre événements, telles que *cause*, *enable*, *prevent* et *intend*, en mettant en œuvre des modèles d'état de l'art. Le modèle proposé améliore de 25% la performance par rapport au précédent meilleur système [92], atteignant un score F1 moyen de **0.763**.

8.4.1 État de l'art :

Des approches supervisées, non-supervisées, par supervision distante ou semi-supervisées ont été utilisées pour extraire les relations d'événements (causalité, temporalité, coréférence). Les LLMs comme GPT-3.5 (via few-shot learning) surpassent REBEL [46] sur CoNLL04.

8.4.2 Méthodologie proposée :

Trois sous-tâches sont identifiées :

(1) **Détection de relation (RD)** : classification binaire (*relation causale ou non*),

(2) **Classification de la relation (RC)** : prédiction parmi les 5 classes,

(3) **Extraction des événements (EE)** : détection des spans via étiquetage BIO.

Trois stratégies sont évaluées :

- **Modularisée** : chaque sous-tâche est traitée indépendamment (RoBERTa, BERT),
- **Fin-à-fin (end-to-end)** : architecture multi-tâches avec têtes dédiées (RoBERTa, REBEL),
- **Prompting LLM** : évaluation de GPT-4 et Zephyr via zero/few-shot.

8.4.3 Résultats expérimentaux (F1-moyen sur données combinées) :

- **Modèle RoBERTa (séparé)** : 0.763 (meilleur résultat global),
- **REBEL (end-to-end)** : 0.68,
- **GPT-4 (4-shot)** : 0.39,
- **Zephyr (4-shot)** : 0.23.

Impact du raisonnement de sens commun : L'intégration de connaissances de sens commun (via données augmentées GPT-3.5) améliore :

- **REBEL** : +4% (RD), +10% (RC), +11% (EE),
- **RoBERTa** : légère amélioration en RC, stabilité en RD.

Comparaison des stratégies :

- **66.6%** des cas : le **fin-à-fin** est supérieur à l'approche modulaire.
- **Extraction d'événements** : mieux gérée par les modèles séparés.
- **LLMs** : sous-performent face aux PLMs. GPT-4 dépasse Zephyr mais reste en retrait (F1 max = 0.52).

8.4.4 Interface démo :

Une application Streamlit est développée. Elle permet à l'utilisateur de :

- choisir les modèles pour chaque sous-tâche (RD, RC, EE),
- visualiser les relations extraites avec mise en évidence colorée,
- tester l'extraction sur des phrases personnalisées.

8.4.5 Conclusion :

- Les PLMs surpassent les LLMs pour les tâches structurées.
- L'architecture modulaire avec RoBERTa donne les meilleurs résultats.
- L'intégration du raisonnement de sens commun est bénéfique pour la généralisation.

8.5 Applications des graphes de connaissances basés sur les événements

Ce chapitre explore deux principales applications des graphes de connaissances basés sur les événements qui ont été développées et évaluées : la génération de récits basée sur les graphes de connaissances et la vérification de faits sensible à la causalité. Ces deux applications s'appuient sur les relations d'événements définies dans l'ontologie FARO et utilisent des techniques d'extraction de relations événementielles pour améliorer le raisonnement, la génération de texte et la prise de décision.

8.5.1 Génération de récits à partir de graphes d'événements

Les récits sont des outils fondamentaux pour transmettre le savoir et influencer la perception humaine. Les modèles actuels de génération de texte tels que BERT et GPT-3, bien que très fluides linguistiquement, sont limités en termes de couverture des connaissances et de cohérence sémantique. Pour remédier à ces limites, notre approche intègre des graphes de connaissances basés sur les événements et utilise des relations causales affinées pour produire des récits plus informatifs et cohérents sur le plan sémantique.

La méthode étend le jeu de données WebNLG en y incorporant le jeu de données FARO, introduisant des relations sémantiques fines telles que *cause*, *prevent*, *enable*, et *intend*. Ces relations enrichissent la structure et le contenu des récits générés.

Le processus comprend :

- L'extraction des événements et des relations fines à l'aide de REBEL, ainsi que le regroupement des mentions coréférentes via la résolution de coréférence événementielle.
- Le résumé du graphe de connaissances à l'aide d'une requête SPARQL guidée par des heuristiques basées sur la fréquence pour identifier les nœuds pertinents (4W : qui, quoi, quand, où).
- L'injection de ces nœuds sélectionnés dans le modèle JointGT pour la génération de texte. Le modèle est affiné sur un jeu de données combiné WebNLG + FARO.

Le modèle a été évalué quantitativement (BLEU, METEOR, ROUGE) et qualitativement par des annotations humaines. Bien que les scores de métriques soient légèrement inférieurs à cause des différences de structure et d'échelle, les résultats qualitatifs ont montré une amélioration de l'adéquation et de la cohérence des récits générés.

Les travaux futurs viseront à améliorer la qualité de l'extraction d'événements et à sélectionner des événements indirectement liés pour générer des sous-graphes plus riches.

8.5.2 Raisonnement causal pour la vérification de faits

Dans le domaine de la vérification de faits, nous proposons une approche fondée sur le raisonnement causal explicable, visant à détecter et justifier les incohérences entre affirmations et preuves à l'aide de relations événementielles affinées. Contrairement aux systèmes actuels souvent opaques, notre méthode s'appuie sur une architecture interprétable combinant extraction de relations, calculs sémantiques et règles logiques.

Le moteur de raisonnement mis en place démontre de bonnes capacités d'inférence, en particulier lorsque les cas d'usage correspondent aux hypothèses causales du système. Dans ces configurations, le raisonneur s'avère efficace, et il reste performant même dans des contextes plus génériques où la couverture n'est pas garantie. Cette robustesse est constatée aussi bien sur AVERITEC que sur FEVEROUS, ce dernier montrant une compatibilité naturelle avec les cas d'usage du système.

Ces résultats valident l'intérêt de l'intégration du raisonnement causal fin dans des systèmes de fact-checking explicables, et ouvrent la voie à des développements futurs visant à enrichir l'extraction automatique et à étendre l'évaluation à des jeux de données variés, notamment ceux orientés vers la causalité comme CHECKWHY.

8.5.3 Conclusion

Ces deux applications démontrent la puissance et la polyvalence des graphes de connaissances événementiels pour améliorer la compréhension du langage, le raisonnement et l'explicabilité. Qu'il s'agisse de génération de récits ou de vérification de faits causale, l'intégration de sémantiques événementielles fines et de représentations structurées permet de concevoir de nouveaux systèmes d'IA interprétables, plus proches de la logique humaine.

8.6 Conclusion générale

Cette thèse a exploré l'extraction automatique de relations causales affinées entre événements à partir de textes, en se concentrant sur deux principales applications : la génération de récits et la vérification explicable de faits. Ce travail propose à la fois des ressources fondamentales et des systèmes concrets qui approfondissent notre compréhension de la causalité événementielle et enrichissent la sémantique des applications en aval.

Pour pallier l'absence de formalisation dans les modèles de relations événementielles existants, nous avons introduit l'ontologie FARO (Facts and Events Relationship Ontology), conçue pour modéliser des relations causales et contingentes fines au sein de graphes de connaissances. FARO permet un raisonnement rétrospectif et prospectif via des relations sémantiques précises

telles que *Cause*, *Intend*, *Enable*, *Prevent*, et *Not Cause*. Ce modèle favorise une navigation plus significative entre événements, au bénéfice de la représentation des connaissances et de l'interprétabilité dans les systèmes de décision.

Sur cette base, nous avons construit un jeu de données raffiné de relations événementielles, au-delà des benchmarks traditionnels comme TimeBank ou EventCausality. Ce jeu intègre de l'augmentation de données via LLMs, une validation manuelle, ainsi que des connaissances issues d'ATOMIC, pour une représentation plus riche et diversifiée des connexions causales entre événements. En traitant les problèmes de rareté relationnelle et de déséquilibre des classes, nous posons les bases de futurs systèmes d'extraction causale plus performants.

Nous avons ensuite proposé un modèle basé sur RoBERTa pour l'extraction de ces relations. Des expérimentations approfondies ont montré que notre approche surpasse significativement les bases existantes, avec une amélioration d'environ 25% du F1-score. L'évaluation a révélé plusieurs points clés : les modèles de bout en bout surpassent les approches en pipeline, les connaissances de sens commun améliorent la précision, et les PLMs comme RoBERTa restent les plus efficaces. Cependant, la désambiguïsation des relations et la coréférence événementielle restent des défis ouverts.

Pour favoriser l'accès et l'expérimentation, nous avons développé une pipeline interactive et publique pour l'extraction des relations événementielles. Nous avons démontré la valeur pratique de ce système à travers deux cas d'usage concrets :

Génération de récits : en intégrant le jeu FARO au benchmark WebNLG, nous avons permis la génération de récits sémantiquement enrichis. Les expériences montrent que les relations événementielles raffinées améliorent l'informativité et la structure du texte généré, bien que les gains en fluidité restent modestes. Les limites liées à la couverture des sous-événements soulignent le besoin de techniques plus avancées de résumé de graphe et de regroupement au niveau du document.

Vérification explicable de faits : Nous avons proposé un cadre de raisonnement explicable combinant des relations causales extraites et des règles logiques explicites pour inférer la véracité des affirmations. Le raisonneur montre des performances prometteuses, notamment en termes d'explicabilité. Lorsqu'il est confronté à des cas d'usage bien définis, il produit des résultats solides, validant ainsi la pertinence de son architecture orientée interprétabilité. Même dans des contextes plus génériques ou moins structurés, où l'alignement avec les schémas de raisonnement attendus n'est pas garanti, le système reste robuste, avec de bonnes performances observées sur les jeux de données AVERITEC et FEVEROUS. La généralisation observée sur FEVEROUS suggère une compatibilité naturelle avec les faits concrets, renforçant son potentiel pour des applications de vérification transparentes et adaptables.

Dans l'ensemble, cette thèse propose de nouveaux modèles, jeux de données, outils et cadres

de raisonnement qui font progresser l'état de l'art en extraction de relations événementielles et dans ses applications. En combinant sémantique formelle, apprentissage machine et cas d'usage concrets, ce travail ouvre la voie à de futures recherches sur la compréhension causale à partir des textes.

Bibliography

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. Technical report, Microsoft, 2024.

[2] Manel Achichi, Pasquale Lisena, Konstantin Todorov, Raphaël Troncy, and Jean Delahousse. DOREMUS: A Graph of Linked Musical Works. In *17th International Semantic Web Conference (ISWC)*, Monterey, CA, USA, 10 2018.

[3] Bradley P. Allen and Paul T. Groth. Evaluating Class Membership Relations in Knowledge Graphs using Large Language Models. In *21st Extended Semantic Web Conference*

(ESWC), Special Track on Large Language Models for Knowledge Engineering, 2024.

[4] Bradley P. Allen, Lise Stork, and Paul Groth. Knowledge Engineering Using Large Language Models. *Transactions on Graph Data and Knowledge*, 2023.

[5] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task. In Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne, and Andreas Vlachos, editors, *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic, November 2021. Association for Computational Linguistics.

[6] Emna Amdouni, Arkopaul Sarkar, Clement Jonquet, and Mohamed Hedi Karray. IndustryPortal: a Common Repository for FAIR Ontologies in Industry 4.0. In *22nd International Semantic Web Conference (ISWC), Poster and Demo Tracj*, Athens, Greece, 2023.

[7] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. In *AI Magazine*, volume 35, pages 105–120, 2014.

[8] Mary-Jane Antia and C. Maria Keet. Automating the Generation of Competency Questions for Ontologies with AgOCQs. In Fernando Ortiz-Rodriguez, Boris Villazón-Terrazas, Sanju Tiwari, and Carlos Bobed, editors, *Knowledge Graphs and Semantic Web*, pages 213–227, Cham, 2023. Springer Nature Switzerland.

[9] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes. *VLDB Endowment*, 17(2):92––105, 2023.

[10] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating fact checking explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, July 2020. Association for Computational Linguistics.

[11] Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. Revisiting Joint Modeling of Cross-document Entity and Event Coreference Resolution. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy, 2019. ACL.

[12] Karim Benhocine, Adel Hansali, Leila Zemmouchi-Ghomari, and Abdessamed Reda Ghomari. Towards an automatic SPARQL query generation from ontology competency questions. *International Journal of Computers and Applications*, 44(10):971–980, 2022.

[13] Inès Blin. Building Narrative Structures from Knowledge Graphs. In *The Semantic Web: ESWC 2022 Satellite Events*, pages 234–251, Germany, 2022. Springer.

[14] Inès Blin, Ilaria Tiddi, Remi van Trijp, and Annette ten Teije. Identifying graph traversal strategies to build narrative graphs. *Under review*, 2023.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[16] Tommaso Caselli and Piek Vossen. The Event StoryLine Corpus: A New Benchmark for Causal and Temporal Relation Extraction. In *Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada, 08 2017. Association for Computational Linguistics.

[17] J Harry Caufield, Harshad Hegde, Vincent Emonet, Nomi L Harris, Marcin P Joachimiak, Nicolas Matentzoglu, HyeongSik Kim, Sierra Moxon, Justin T Reese, Melissa A Haendel, Peter N Robinson, and Christopher J Mungall. Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning. *Bioinformatics*, page 104, 02 2024.

[18] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), 2024.

[19] Harrison Chase. LangChain, October 2022.

[20] Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online, 2020. ACL.

[21] Zhiyun Chen, Qing Zhang, Jie Liu, Yufei Wang, Haocheng Lv, LanXuan Wang, Jianyong Duan, Mingying Xv, and Hao Wang. Counterfactual Multimodal Fact-Checking Method Based on Causal Intervention. In Zhouchen Lin, Ming-Ming Cheng, Ran He, Kurban

Ubul, Wushouer Silamu, Hongbin Zha, Jie Zhou, and Cheng-Lin Liu, editors, *Pattern Recognition and Computer Vision*, pages 582–595, Singapore, 2025. Springer Nature Singapore.

[22] Prafulla Kumar Choubey and Ruihong Huang. Discourse level event temporal ordering with the aid of event coreference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3757–3769, 2021.

[23] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *ArXiv*, abs/1803.05457, 2018.

[24] Sebastian J. Crutch, Paul Williams, Gerard R. Ridgway, and Laura Borgenicht. The role of polarity in antonym and synonym conceptual knowledge: Evidence from stroke aphasia and multidimensional ratings of abstract words. *Neuropsychologia*, 50(11):2636–2644, 2012.

[25] Jacopo de Berardinis, Valentina Anita Carriero, Nitisha Jain, Nicolas Lazzari, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. The Polifonia Ontology Network: Building a Semantic Backbone for Musical Heritage. In *The Semantic Web – ISWC*, 2023.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[27] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In *2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1286–1305. ACL, 2021.

[28] Martin Doerr. The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata. *AI magazine*, 24(3):75–75, 2003.

[29] Hongchao Fan and Liqiu Meng. Analysis of events in 3D building models. In Lin Liu, Xia Li, Kai Liu, Xinchang Zhang, and Aijun Chen, editors, *Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Geo-Simulation and Virtual GIS Environments*, volume 7143, pages 1047 – 1058. International Society for Optics and Photonics, SPIE, 2008.

[30] Johannes Frey, Lars-Peter Meyer, Natanael Arndt, Felix Brei, and Kirill Bulert. Benchmarking the Abilities of Large Language Models for RDF Knowledge Graph Creation and

Comprehension: How Well Do LLMs Speak Turtle? In *Workshop on Deep Learning for Knowledge Graphs (DL4KG)*, 2023.

[31] Antony Galton. States, processes and events, and the ontology of causal relations. *Frontiers in Artificial Intelligence and Applications*, 239:279–292, Jan 2012.

[32] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG Challenge: Generating Text from RDF Data. In *10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain, 2017. ACL.

[33] Hamed Babaei Giglou, Jennifer D'Souza, Felix Engel, and Sören Auer. LLMs4OM: Matching Ontologies with Large Language Models. In *21st Extended Semantic Web Conference (ESWC), Special Track on Large Language Models for Knowledge Engineering*, 2024.

[34] Simon Gottschalk and Elena Demidova. EventKG – the hub of event knowledge on the web – and biographical timeline generation. *Semantic Web*, 10:1039–1070, 2019. 6.

[35] Melanie C Green, Timothy C Brock, and Geoff F Kaufman. Understanding media enjoyment: The role of transportation into narrative worlds. *Communication theory*, 14(4):311–327, 2004.

[36] Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. What is Event Knowledge Graph: A Survey. *CoRR*, abs/2112.15280, 2021.

[37] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 2022.

[38] Yujing Han, Nanyun Peng, and Heng Ji. Fine-grained event role alignment using contrastive learning with event graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3938–3950, 2021.

[39] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1803–1812, New York, NY, USA, 2017. Association for Computing Machinery.

[40] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP Using Linked Data. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof

Janowicz, editors, *The Semantic Web – ISWC 2013*, pages 98–113, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[41] Sven Hertling and Heiko Paulheim. OLaLa: Ontology Matching with Large Language Models. In *12th Knowledge Capture Conference (KCAP)*. Association for Computing Machinery, 2023.

[42] Marvin Hofer, Johannes Frey, and Erhard Rahm. Towards self-configuring Knowledge Graph Construction Pipelines using LLMs – A Case Study with RML. In *5th International Workshop on Knowledge Graph Construction*, 2024.

[43] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[44] Yu Hong, Tongtao Zhang, Tim O'Gorman, Sharone Horowit-Hendler, Heng Ji, and Martha Palmer. Building a Cross-document Event-Event Relation Corpus. In *10th Linguistic Annotation Workshop 2016 (LAW-X 2016)*, pages 1–6, Berlin, Germany, Aug 2016. Association for Computational Linguistics.

[45] Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. Large language model-based event relation extraction with rationales. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *31st International Conference on Computational Linguistics*, pages 7484–7496, Abu Dhabi, UAE, 2025. Association for Computational Linguistics.

[46] Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381. ACL, 2021.

[47] IFLA Study Group on the Functional Requirements for Bibliographic Records. Functional Requirements for Bibliographic Records: Final Report. Technical report, International Federation of Library Associations and Institutions (IFLA), 2009.

[48] James Malone, Andy Brown, Allyson L Lister, Jon Ison, Duncan Hull, Helen Parkinson, and Robert Stevens. The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *Journal of biomedical semantics*, 2014.

[49] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[50] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[51] Vikas Jindal, Seema Bawa, and Shalini Batra. A review of ranking approaches for semantic search on Web. *Information Processing & Management*, 50(2):416–425, 2014.

[52] Ken Kaneiwa, Michiaki Iwazume, and Ken Fukuda. An Upper Ontology for Event Classifications and Relations. In *AI 2007: Advances in Artificial Intelligence*, pages 394–403, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[53] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. JointGT: Graph-Text Joint Representation Learning for Text Generation from Knowledge Graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 2526–2538. Association for Computational Linguistics, 2021.

[54] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. JointGT: Graph-Text Joint Representation Learning for Text Generation from Knowledge Graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538. ACL, 2021.

[55] C. Maria Keet. The African wildlife ontology tutorial ontologies. *Journal of Biomedical Semantics*, 11(1), 2020.

[56] Selim Kılıç. Kappa testi. *Journal of Mood Disorders*, 5(3), 2015.

[57] Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling, 2023.

[58] Holger Knublauch and Dimitris Kontokostas. Shapes Constraint Language (SHACL). Technical report, W3C, 2017.

[59] Ioannis Kompatsiaris. Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support. https://demcare.eu/, 2012.

[60] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. In *NAACL*, 2019.

[61] Neema Kotonya and Francesca Toni. Explainable automated fact-checking: A survey. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th Inter-*

national Conference on Computational Linguistics, pages 5430–5443, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[62] Daniel Krech, Gunnar AAstrand Grimnes, Graham Higgins, Jörn Hees, Iwan Aucamp, Niklas Lindström, Natanael Arndt, Ashley Sommer, Edmond Chuc, Ivan Herman, Alex Nelson, Jamie McCusker, Tom Gillespie, Thomas Kluyver, Florian Ludwig, Pierre-Antoine Champin, Mark Watts, Urs Holzer, Ed Summers, Whit Morriss, Donny Winston, Drew Perttula, Filip Kovacevic, Remi Chateauneu, Harold Solbrig, Benjamin Cogrel, and Veyndan Stuart. RDFLib, August 2023.

[63] Adila Alfa Krisnadhi and Pascal Hitzler. A Core Pattern for Events. In *7th Workshop on Ontology and Semantic Web Patterns (WOP@ISWC)*, Kobe, Japan, 2016. IOS Press.

[64] Manolis Kyriakakis, Ion Androutsopoulos, Artur Saudabayev, and Joan Ginés i Ametllé. Transfer learning for causal sentence detection. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors, *18th BioNLP Workshop and Shared Task*, pages 292–297, Florence, Italy, August 2019. Association for Computational Linguistics.

[65] Jie Lei, Licheng Yu Li, Mohit Bansal, and Tamara L Berg. Less is more: Clipbert for video-and-language learning via sparse sampling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021.

[66] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics, 2020.

[67] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. Pretrained Language Model for Text Generation: A Survey. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), Survey Track*, pages 4492–4499. IJCAI Organization, 2021.

[68] Zhongyang Li, Xiao Ding, Ting Liu, J. Edward Hu, and Benjamin Van Durme. Guided Generation of Cause and Effect. In *29^{th} International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[69] Pasquale Lisena, Albert Meroño-Peñuela, Tobias Kuhn, and Raphaël Troncy. Easy Web API Development with SPARQL Transformer. In *Proceedings of the 18^{th} International Semantic Web Conference (ISWC)*, Auckland, New Zealand, 2019. Semantic Web Science Association (SWSA).

[70] Pasquale Lisena, Daniel Schwabe, Marieke van Erp, Raphaël Troncy, William Tullett, Inger Leemans, Lizzie Marx, and Sofia Colette Ehrich. Capturing the Semantics of Smell: The Odeuropa Data Model for Olfactory Heritage Information. In *The Semantic Web*, pages 387–405. Springer International Publishing, 2022.

[71] Junfei Liu, Shaotong Sun, and Fatemeh Nargesian. Causal Dataset Discovery with Large Language Models. In *Workshop on Human-In-the-Loop Data Analytics*, pages 1—-8. Association for Computing Machinery, 2024.

[72] Kang Liu, Yubo Chen, Jian Liu, Xinyu Zuo, and Jun Zhao. Extracting Events and Their Relations from Texts: A Survey on Recent Research Progress and Challenges. *AI Open*, 1:22–39, 2020.

[73] Li Liu, Mengge He, Guanghui Xu, Mingkui Tan, and Qi Wu. How to Train Your Agent to Read and Write. In *AAAI Conference on Artificial Intelligence*, pages 13397–13405, 2021.

[74] Zhiyi Luo, Yuchen Sha, Kenny Q. Zhu, Seung-won Hwang, and Zhongyuan Wang. Commonsense Causal Reasoning between Short Texts. In *Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, page 421–430. AAAI Press, 2016.

[75] Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. Coherence boosting: When your pretrained language model is not paying enough attention. In *60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[76] Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. Annotating Causality in the TempEval-3 Corpus. In *EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden, 4 2014. Association for Computational Linguistics.

[77] Paramita Mirza and Sara Tonelli. CATENA: Causal and temporal relation extraction from natural language texts. In *26^{th} International Conference on Computational Linguistics*, pages 64–75. ACL, 2016.

[78] Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. Overview of tac kbp 2015 event nugget track. *Theory and Applications of Categories*, 2015.

[79] Boris Motik. Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web (ISWC)*, pages 550–565. Springer Berlin Heidelberg, 2010.

[80] Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. Joint Reasoning for Temporal and Causal Relations. In *56^{th} Annual Meeting of the Association for Computational Linguistics*, volume 1, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[81] OpenAI. GPT-4 Technical Report, 2023.

[82] Panče Panov, Larisa N. Soldatova, and Sašo Džeroski. Generic ontology of datatypes. *Information Sciences*, 329:900–920, 2016.

[83] Mateusz Pawlik and Nikolaus Augsten. RTED: a robust algorithm for the tree edit distance. *VLDB Endowment*, 5(4):334—-345, 2011.

[84] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[85] Jedrzej Potoniec, Dawid Wiśniewski, Agnieszka Ławrynowicz, and C. Maria Keet. Dataset of ontology competency questions to SPARQL-OWL queries translations. *Data in Brief*, 2020.

[86] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111, 2022.

[87] James Pustejovsky, José M. Castaño, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. TimeML: Robust Specification of Event and Temporal Expressions in Text. In Mark T. Maybury, editor, *New Directions in Question Answering*, pages 28–34. AAAI Press, 2003.

[88] Joe Raad and Christophe Cruz. A Survey on Ontology Evaluation Methods. In *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Lisbonne, Portugal, 2015.

[89] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1.8, 2019.

[90] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21(1), 2020.

[91] Lance Ramshaw and Mitch Marcus. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*, 1995.

[92] Youssra Rebboud, Pasquale Lisena, and Raphaël Troncy. Prompt-based data augmentation for semantically-precise event relation classification. In CEUR, editor, *SEMMES 2023, Semantic Methods for Events and Stories, May 23-28, 2023, Heraklion, Greece*, Heraklion, 2023.

[93] Youssra Rebboud, Lionel Tailhardat, Pasquale Lisena, and Raphaël Troncy. Can LLMs generate competency questions? In *21st Extended Semantic Web Conference (ESWC), Special Track on Large Language Models for Knowledge Engineering*, 2024.

[94] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics.

[95] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z. Pan, Kees van Deemter, and Robert Stevens. Towards Competency Question-Driven Ontology Authoring. In *11th European Semantic Web Conference (ESWC)*, 2014.

[96] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating Pretrained Language Models for Graph-to-Text Generation. In *3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online, 2021. ACL.

[97] Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphael Troncy, and Xavier Tannier. Searching News Articles Using an Event Knowledge Graph Leveraged by Wikidata. In *2019 World Wide Web Conference*, WWW, page 1232–1239, New York, NY, USA, 2019. ACL.

[98] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021.

[99] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *33rd AAAI Conference on Artificial Intelligence*. AAAI Press, 2019.

[100] Jonathan Schaffer. The Metaphysics of Causation, 2016.

[101] Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. F–a Model of Events Based on the Foundational Ontology Dolce+DnS Ultralight. In *5th International Conference on Knowledge Capture (K-CAP)*, page 137–144, New York, NY, USA, 2009. Association for Computing Machinery.

[102] Thomas Schleider, Raphaël Troncy, Mar Gaitan, Ester Alba, and et al. The SILKNOW Knowledge Graph. *Semantic Web Journal, Special Issue on Cultural Heritage and Semantic Web, March 2021, IOS Press*, 2021.

[103] Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. AVeriTeC: A Dataset for Real-world Claim Verification with Evidence from the Web. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 65128–65167, New Orleans, Louisiana, USA, 2023. Curran Associates, Inc.

[104] Timo Pierre Schrader, Simon Razniewski, Lukas Lange, and Annemarie Friedrich. BoschAI @ Causal News Corpus 2023: Robust Cause-Effect Span Extraction using Multi-Layer Sequence Tagging and Data Augmentation. In Ali Hürriyetoğlu, Hristo Tanev, Vanni Zavarella, Reyyan Yeniterzi, Erdem Yörük, and Milena Slavcheva, editors, *Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text*, pages 38–43, Varna, Bulgaria, 2023. INCOMA Ltd., Shoumen, Bulgaria.

[105] Juan Sequeda, Dean Allemang, and Bryon Jacob. A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases, 2023.

[106] Ryan Shaw, Raphaël Troncy, and Lynda Hardman. LODE: Linking Open Descriptions of Events. In Asunción Gómez-Pérez, Yong Yu, and Ying Ding, editors, *The Semantic Web*, pages 153–167, Berlin, Heidelberg, 2009. Springer.

[107] Rob Shearer, Boris Motik, and Ian Horrocks. HermiT: A Highly-Efficient OWL Reasoner. In *OWL: Experiences and Directions*, 2008.

[108] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 395–405, New York, NY, USA, 2019. Association for Computing Machinery.

[109] Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. CHECKWHY: Causal Fact Verification via Argument Structure. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15636–15659, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[110] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. Linked Open Vocabulary Ranking and Terms Discovery. In *12th International Conference on Semantic Systems*, pages 1—-8. Association for Computing Machinery, 2016.

[111] Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. NORIA-O: an Ontology for Anomaly Detection and Incident Management in ICT Systems. In *Semantic Web – 21st International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26 - 30, 2024, Proceedings*, 2024.

[112] Fiona Anting Tan, Jay Desai, and Srinivasan H. Sengamedu. Enhancing fact verification with causal knowledge graphs and transformer-based retrieval for deductive reasoning. In Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos, editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 151–169, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[113] Fiona Anting Tan, Hansi Hettiarachchi, Ali Hürriyetoğlu, Tommaso Caselli, Onur Uca, Farhana Ferdousi Liza, and Nelleke Oostdijk. Event Causality Identification with Causal News Corpus - Shared Task 3, CASE 2022. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, pages 195–208, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[114] Fiona Anting Tan, Ali Hürriyetoğlu, Tommaso Caselli, Nelleke Oostdijk, Tadashi Nomoto, Hansi Hettiarachchi, Iqra Ameer, Onur Uca, Farhana Ferdousi Liza, and Tiancheng Hu. The Causal News Corpus: Annotating Causal Relations in Event Sentences from News. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2298–2310, Marseille, France, June 2022. European Language Resources Association.

[115] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023.

[116] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs. In *34th International Conference on Machine*

Learning (ICML), volume 70, pages 3462—3471. JMLR.org, 2017.

[117] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct Distillation of LM Alignment, 2023.

[118] Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In 7^{th} *International Workshop on Semantic Evaluation (SemEval)*, pages 1–9, Atlanta, USA, 2013. Association for Computational Linguistics.

[119] Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. Design and use of the Simple Event Model (SEM). *Journal of Web Semantics*, 9(2):128–136, 2011.

[120] Shikhar Vashishtha and Partha Talukdar. Temporal event knowledge graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4730–4741, 2021.

[121] Marc Verhagen, Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok B. Jang, Anna Rumshisky, John Phillips, and James Pustejovsky. Automating Temporal Annotation with TARSQI. In *ACL Interactive Poster and Demonstration Sessions*, pages 81–84, Ann Arbor, Michigan, 2005. ACL.

[122] Marc Verhagen, Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok Bae Jang, Anna Rumshisky, Jon Phillips, and James Pustejovsky. Automating temporal annotation with TARSQI. In *ACL interactive poster and demonstration sessions*, pages 81–84, 2005.

[123] Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014.

[124] Somin Wadhwa, Silvio Amir, and Byron Wallace. Revisiting Relation Extraction in the era of Large Language Models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada, July 2023. Association for Computational Linguistics.

[125] Jiaqi Wang, Yuying Chang, Zhong Li, Ning An, Qi Ma, Lei Hei, Haibo Luo, Yifei Lu, and Feiliang Ren. TechGPT-2.0: A large language model project to solve the task of knowledge graph construction, 2024.

[126] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

[127] Dawid Wisniewski, Jedrzej Potoniec, and Agnieszka Lawrynowicz. SeeQuery: An Automatic Method for Recommending Translations of Ontology Competency Questions into SPARQL-OWL. In *30th ACM International Conference on Information & Knowledge Management (CIKM)*, page 2119–2128. Association for Computing Machinery, 2021.

[128] Dawid Wisniewski, Jedrzej Potoniec, Agnieszka Lawrynowicz, and C. Maria Keet. Competency Questions and SPARQL-OWL Queries Dataset and Analysis. *Journal of Web Semantics*, 59:100534, 2019.

[129] Dawid Wiśniewski, Jędrzej Potoniec, and Agnieszka Ławrynowicz. ReqTagger: A Rule-Based Tagger for Automatic Glossary of Terms Extraction from Ontology Requirements. *Foundations of Computing and Decision Sciences*, 47(1):65–86, 2022.

[130] Phillip Wolff. Representing Causation. *Journal of experimental psychology. General*, 136:82–111, 03 2007.

[131] Jie Yang, Soyeon Caren Han, and Josiah Poon. A survey on extraction of causal relations from natural language text. *Knowledge and Information Systems*, 64:1161 – 1186, 2021.

[132] Xinran Yang and Ilaria Tiddi. Creative Storytelling with Language Models and Knowledge Graphs. In *International Conference on Information and Knowledge Management*, 2020.

[133] Xinyu Yao, Zheng Zhang, Muhao Chen, Yizhou Wang, and Heng Ji. Weakly supervised temporal relation extraction with reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 39–49, 2020.

[134] Bei Yu, Yingya Li, and Jun Wang. Detecting Causal Language Use in Science Findings. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4656–4666, 2019.

[135] Bei Yu, Jun Wang, Lu Guo, and Yingya Li. Measuring Correlation-to-Causation Exaggeration in Press Releases. In *28th International Conference on Computational Linguistics (COLING)*, pages 4860–4872, 2020.

[136] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[137] Rowan Zellers, Ari Holtzman, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022.

[138] ChengXiang Zhai. Large language models and future of information retrieval: Opportunities and challenges. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 481–490. Association for Computing Machinery, 2024.

[139] Sheng Zhang, Heng Ji, Shaodan Pan, Jonathan May, and Xiaoman Pan. Annotating and modeling fine-grained event evolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5612–5623, 2020.

[140] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China.