



# Large Language Model-based Framework for Open Information Extraction, Triplet Matching, and Text Comparison

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Master Program Data Science**

eingereicht von

**Tamas Csakvari, BSc**

Matrikelnummer 11817699

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass Gábor Recski, PhD

Mitwirkung: Adam Kovacs, BSc MSc

Wien, 1. Juli 2025

*Tamas Csakvari*

Tamas Csakvari

Gábor Recski





# Large Language Model-based Framework for Open Information Extraction, Triplet Matching, and Text Comparison

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Master Program Data Science**

by

**Tamas Csakvari, BSc**

Registration Number 11817699

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass Gábor Recski, PhD

Assistance: Adam Kovacs, BSc MSc

Vienna, July 1, 2025

*Tamas Csakvari*

Tamas Csakvari

Gábor Recski



# Erklärung zur Verfassung der Arbeit

Tamas Csakvari, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 1. Juli 2025

*Tamas Csakvari*

---

Tamas Csakvari



# Danksagung

Ich möchte meinem Betreuer, Gábor Recski, meinen Dank für die Betreuung dieses Dissertationsprojekts und die Bereitstellung des institutionellen Rahmens ausdrücken, der diese Forschung ermöglicht hat. Seine akademische Expertise und Anleitung durch die formalen Anforderungen des Dissertationsprozesses waren wertvoll, um diese Arbeit zum Abschluss zu bringen.

Ich möchte auch meinem Co-Betreuer, Adam Kovacs, für seinen fachlichen Input und seine administrative Unterstützung während der verschiedenen Phasen dieses Projekts danken. Seine Beteiligung hat dazu beigetragen, diese Dissertation zu gestalten und ihre Qualität durch konstruktives Feedback und durchdachte Diskussionen zu verbessern.

Ich bin der Fakultät und den Mitarbeitern der Technischen Universität Wien dankbar, die die Ressourcen und die Umgebung bereitgestellt haben, die für den Abschluss dieser Arbeit erforderlich waren.



# Acknowledgements

I would like to express my appreciation to my supervisor, Gábor Recski, for overseeing this thesis project and providing the institutional framework that made this research possible. His academic expertise and guidance through the formal requirements of the thesis process were valuable in bringing this work to completion.

I also extend my gratitude to my co-supervisor, Adam Kovacs, for his technical input and administrative support throughout various stages of this project. His involvement helped shape this thesis and elevate its quality through constructive feedback and thoughtful discussions.

I'm grateful to the faculty and staff at Vienna University of Technology who provided the resources and environment needed to complete this work.



# Kurzfassung

Das Wachstum unstrukturierter digitaler Texte erfordert effektive Methoden zur Wissensextraktion. Während die traditionelle Information Extraction durch starre Schemata begrenzt ist, bietet Open Information Extraction (OIE) die notwendige Flexibilität. Große Sprachmodelle (Large Language Models, LLMs) sind vielversprechend für OIE, aber ihre Anwendung auf OIE und das semantische Matching von Triplets ist noch wenig erforscht.

Diese Arbeit stellt ein neuartiges, modulares LLM-basiertes Framework vor und evaluiert es. Das Framework wurde für OIE, das anschließende semantische Matching von Triplets und den Textvergleich entwickelt und auf einem deutschen juristischen Ausbildungsdatensatz mit studentischen Antworten validiert. Das Framework verwendet LLMs, um zunächst (Subjekt, Relation, Objekt)-Triplets aus den deutschen juristischen Texten zu extrahieren. Diese extrahierten Kandidaten-Triplets werden dann mittels eines weiteren LLM-gesteuerten Matching-Prozesses semantisch mit vordefinierten Ziel-Triplets (die wichtige juristische Inhalte repräsentieren) verglichen. Die Leistungsfähigkeit des Systems wurde anhand eines Datensatzes von studentischen Antworten zu einem spezifischen Rechtsfall rigoros evaluiert, indem die automatisierten Ergebnisse mit einer von Menschen kommentierten Ground Truth verglichen wurden. Mehrere hochmoderne LLMs (einschließlich der GPT-4-Serie, Llama, DeepSeek) wurden gebenchmarkt, ebenso wie alternative Methoden wie die End-to-End-LLM-Evaluierung, regelbasierte OIE und stringbasiertes Triplet-Matching.

Die Ergebnisse zeigen die beachtliche Leistungsfähigkeit des Frameworks, wobei die leistungsstärkste Konfiguration (GPT-4.1-mini sowohl für OIE als auch für das Matching) eine Genauigkeit von 80,0% und einen Matthews Correlation Coefficient (MCC) von 0,589 erreichte. Dieser modulare Ansatz aus LLM-OIE plus LLM-Matching übertraf im Allgemeinen holistische End-to-End-LLM-Methoden und einfachere regelbasierte oder stringbasierte Matching-Techniken, was den Wert strukturierter intermediärer Repräsentationen unterstreicht.

Diese Forschung bestätigt den Nutzen von LLMs für nuancierte OIE und semantische Vergleiche in spezialisierten, nicht-englischen Domänen. Das entwickelte quelloffene, modulare Framework dient als praktisches Werkzeug und trägt zum Verständnis der Fähigkeiten und Grenzen von LLMs bei der strukturierten Wissensextraktion bei und bietet

eine Grundlage für fortschrittliche automatisierte Bewertungs- und Informationsabrufsysteme.

# Abstract

The growth of unstructured digital text demands effective knowledge extraction methods. While traditional Information Extraction is limited by rigid schemas, Open Information Extraction (OIE) provides needed flexibility. Large Language Models (LLMs) show promise for OIE but their application to both OIE and semantic triplet matching remains underexplored.

This thesis introduces and evaluates a novel, modular LLM-based framework designed for OIE, subsequent semantic triplet matching, and text comparison, with validation performed on a German legal education dataset of student responses. The framework employs LLMs to first extract (subject, relation, object) triplets from the German legal texts. These extracted candidate triplets are then semantically compared against predefined target triplets (representing key legal contents) using an LLM-based triplet matching process. The system's performance was quantitatively and qualitatively evaluated on the dataset of student answers to a specific legal case, comparing LLM-based triplet matching outputs against human-annotated ground truth. Several state-of-the-art LLMs (including GPT-4 series, Llama, DeepSeek) were benchmarked, alongside alternative methods such as end-to-end LLM evaluation, rule-based OIE, and string-based triplet matching for comparison.

Results demonstrate the framework's considerable proficiency, with the top-performing configuration (GPT-4.1-mini for both OIE and triplet matching) achieving 80.0% accuracy and a Matthews Correlation Coefficient (MCC) of 0.589. This modular LLM-OIE plus LLM-matching approach generally outperformed holistic end-to-end LLM methods and simpler rule-based or string-matching techniques, highlighting the value of structured intermediate representations.

This research validates the utility of LLMs for OIE and semantic comparison in a specialized, non-English domain. The developed open-source, modular framework serves as a practical tool and contributes to understanding LLM capabilities and limitations in structured knowledge extraction, offering a foundation for advanced automated assessment and information retrieval systems.



# Contents

<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Contents</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Research Questions . . . . .	3
<b>2 Background</b>	<b>5</b>
2.1 Overview . . . . .	5
2.2 Open Information Extraction . . . . .	5
2.3 Large Language Models . . . . .	13
2.4 Large Language Models for Open Information Extraction and Triplet Matching . . . . .	16
<b>3 Use case</b>	<b>19</b>
3.1 Overview . . . . .	19
3.2 Structure of the Legal Task and Assessment Dataset . . . . .	19
3.3 System Requirements and Challenges . . . . .	23
3.4 Outlook . . . . .	24
<b>4 Methodology</b>	<b>27</b>
4.1 Overview . . . . .	27
4.2 Developed Framework . . . . .	27
4.3 LLM Integration Component . . . . .	32
4.4 Prompt Design . . . . .	32
4.5 Open Information Extraction Component . . . . .	34
4.6 Triplet Matching Component . . . . .	35
4.7 Dataset . . . . .	36
4.8 Tested LLMs . . . . .	36
4.9 Other Methods . . . . .	37
	xv

4.10	Implementation Details . . . . .	43
4.11	Experimental Setup . . . . .	43
4.12	Evaluation Methodology . . . . .	43
<b>5</b>	<b>Results</b>	<b>47</b>
5.1	Qualitative Analysis of Extracted Triplets . . . . .	47
5.2	Quantitative Analysis of Triplet Matching Performance . . . . .	54
5.3	Qualitative Analysis of Triplet Matching Performance . . . . .	56
<b>6</b>	<b>Discussion</b>	<b>63</b>
6.1	Addressing the Research Questions . . . . .	63
6.2	Comparison with Existing Work . . . . .	65
6.3	Implications of the Findings . . . . .	66
6.4	Limitations and Challenges . . . . .	67
6.5	Future Research Directions . . . . .	68
<b>7</b>	<b>Conclusion</b>	<b>71</b>
7.1	Summary of Key Findings . . . . .	71
7.2	Contributions of the Thesis . . . . .	72
	<b>Overview of Generative AI Tools Used</b>	<b>75</b>
	<b>List of Tables</b>	<b>77</b>
	<b>List of Figures</b>	<b>77</b>
	<b>Listings</b>	<b>78</b>
	<b>Bibliography</b>	<b>79</b>

# Introduction

## 1.1 Motivation

The exponential increase in unstructured digital text necessitates automated methods for extracting structured knowledge. Traditional Information Extraction (IE) systems often depend on predefined schemas or ontologies, limiting their adaptability to diverse domains, languages, or evolving knowledge landscapes (Sarawagi et al., 2008). This rigidity hinders their application where comprehensive schemas are unavailable or impractical to construct, such as in specialized fields or when dealing with novel information (Adnan and Akbar, 2019).

Open Information Extraction (OIE) offers a promising alternative by aiming to extract relational tuples, typically (subject, relation, object) triplets, directly from unstructured text without requiring a predefined schema (Yates et al., 2007). This schema-agnostic approach enhances flexibility and scalability, making OIE valuable for diverse applications including knowledge graph construction (Muhammad et al., 2020), question answering (Song et al., 2023), automated fact-checking (Song et al., 2023), and legal or educational content evaluation.

Recent advancements in Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation (Naveed et al., 2023). Their ability to process context and capture nuanced semantic relationships suggests significant potential for improving OIE systems. However, the application of state-of-the-art LLMs to OIE remains understudied. While some studies have explored dynamic prompt engineering for models like LLaMA-2 and GPT-3.5-Turbo (Ling et al., 2023; Qi et al., 2023), these efforts only begin to uncover the broader capabilities, limitations, and generalizability of LLMs in OIE tasks.

At the same time, the growing prevalence of AI-generated content underscores the urgent need for systems that can verify factual consistency between texts (Hamed et al., 2024).

A robust framework that combines LLM-driven OIE with sophisticated triplet matching would provide a structured mechanism to compare information content across texts.

This thesis is motivated by the need to build more reliable, adaptable, and scalable approaches for knowledge extraction and comparison. Leveraging the strengths of LLMs, this work aims to move beyond the limitations of traditional OIE systems by developing a language-agnostic framework, openly available at <https://github.com/TamasCsakvari/oie-llm-framework>, that supports structured knowledge extraction and validation across domains and tasks.

### 1.2 Problem Statement

Despite the promise of LLMs and the flexibility of OIE, effectively integrating these technologies into reliable, scalable pipelines remains a significant and unexplored challenge - particularly for multilingual and domain-specific applications. There is currently a lack of generalizable, modular frameworks that can leverage LLMs for both structured extraction and semantic comparison tasks across diverse inputs.

Two core problems persist:

1. **Reliable Structured Extraction:** While LLMs excel at interpreting text, extracting accurate and complete (subject, relation, object) triplets from unstructured text, without relying on predefined schemas, remains a difficult task. It is essential to ensure that the extracted triplets faithfully represent the core semantic content of the source while minimizing common issues like hallucinations or inconsistencies Zhang et al. (2023).
2. **Robust Semantic Equivalence Matching:** Determining whether two triplets express the same meaning, despite differences in wording, syntax, or structure, is a challenge. Existing triplet matching methods often fall short in capturing true semantic equivalence, especially when surface-level variations obscure deeper similarity.

This thesis addresses these challenges through three interconnected objectives:

1. Develop an LLM-based framework for OIE that eliminates the need for predefined schemas while ensuring consistent and accurate triplet extraction.
2. Create a robust LLM-based triplet matching system capable of identifying semantic equivalence across syntactic and lexical variations.
3. Implement a text comparison methodology that uses these two components to evaluate the overlap of key content between two documents.

Key technical challenges include mitigating LLM hallucinations during structured output generation and ensuring robustness in multilingual and domain-specific contexts. This work develops a flexible framework, the source code for which is accessible at <https://github.com/TamasCsakvari/oie-llm-framework>, in which the underlying LLMs are modular and exchangeable. This enables the use of state-of-the-art models as they evolve, allowing the system to adapt to different datasets and performance needs without requiring major architectural changes.

To evaluate the proposed framework, we use a German legal dataset in which the LLM-based methods are used to extract triplets and match them against a set of predefined triplets, representing key legal content. We then compare the model-generated matches with human-annotated ground truth matches to evaluate the performance of the implemented methods.

By addressing these problems, this thesis contributes to the development of more reliable and extensible methods for automated knowledge extraction and validation, particularly in domains where factual correctness, and semantic alignment are critical. An overview of the proposed LLM-based pipeline is shown in Figure 1.1.

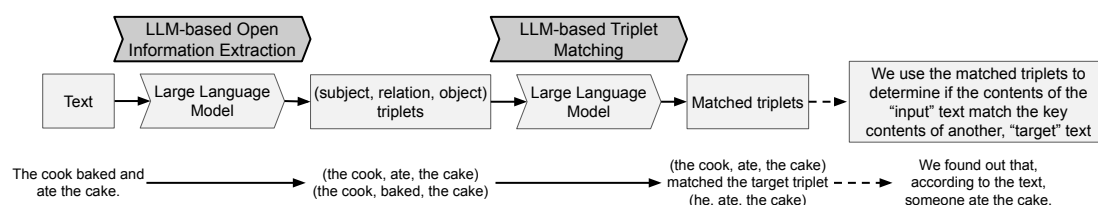


Figure 1.1: A high-level overview of the proposed LLM-based pipeline built in the proposed framework. The process begins with unstructured text, which is transformed into structured (subject, relation, object) triplets using an LLM. These extracted triplets are then semantically compared against target triplets to evaluate the text’s content.

## 1.3 Research Questions

This thesis aims to contribute to the stated problem by answering the following research questions (RQs).

- **RQ1:** To what extent does LLM-based Open Information Extraction serve as an effective foundation for semantic text comparison in German legal domains?
- **RQ2:** How do LLM-derived triplets compare to dependency graph rule-based triplet extraction when used for semantic text comparison in legal assessment?
- **RQ3:** How reliable is LLM-based triplet matching in identifying semantic equivalence between differently worded but conceptually similar information?

## 1. INTRODUCTION

---

- **RQ4:** What evaluation metrics best capture the performance of LLM-based OIE systems, particularly when human-annotated ground truth is available?

# Background

## 2.1 Overview

In this chapter, we explore the two major research areas that form the foundation of our work, followed by their emerging intersection. **First**, we provide a comprehensive examination of OIE, tracing its evolution from early rule-based systems through neural approaches, giving an overview of prominent applications, as well as looking at current triplet matching methods. **Second**, we discuss LLMs, outlining their technical foundations, capabilities, and limitations and potential in natural language processing (NLP). **Finally**, we look at the new but promising integration of these technologies, an area still not widely covered in the literature but central to our thesis. LLMs offer new possibilities for OIE through their contextual understanding and few-shot capabilities, while introducing important considerations around factual grounding and semantic triplet matching. This structure establishes the essential background before introducing our novel approach that builds upon this combination.

## 2.2 Open Information Extraction

### 2.2.1 Introduction to Open Information Extraction

The field of Information Extraction has historically focused on the task of identifying and extracting specific, pre-defined types of information from unstructured text (Sarawagi et al., 2008). This conventional approach typically relies on the creation of fixed schemas that dictate the types of entities and relationships to be extracted. These schemas are often implemented through manually crafted rules or supervised machine learning models trained on annotated data that conforms to the pre-defined structure. While effective for targeted information needs within specific domains, traditional IE exhibits limitations in its ability to scale and adapt to the vast and diverse information landscape of open-

domain text. The inherent rigidity of predefined schemas makes it challenging to discover and extract novel or unanticipated types of information, thus restricting its applicability in scenarios where the information needs are not known in advance or where the text sources are highly varied (Niklaus et al., 2018).

In response to these limitations, the paradigm of Open Information Extraction emerged as a significant shift in the field (Yates et al., 2007). Initially, OIE aimed to extract relational information from text in an unsupervised or minimally supervised manner, thereby eliminating the dependency on predefined schemas. However, the development of large-scale annotated datasets, such as LSOIE, has led to a notable shift towards supervised and neural-based approaches in modern OIE systems. This evolution enables the discovery of a wide spectrum of relationships expressed in diverse text sources, offering a more flexible and scalable solution for extracting knowledge from the ever-growing volume of online information. The core principle of OIE lies in its ability to automatically identify and extract any relational information present in the text, without the need to specify in advance what types of relationships are of interest.

OIE can be formally defined as the task of automatically extracting relational information from unstructured text into a structured format. The fundamental unit of extracted information is the triplet consisting of a subject, relation phrase, and object. For instance, from the sentence "Albert Einstein developed the theory of relativity." an OIE system would extract the triplet (Albert Einstein, developed, the theory of relativity). This structure provides a simple yet powerful way to represent knowledge without relying on predefined ontologies or relation schemas.

The primary goals of OIE systems include maximizing the coverage of extracted information by identifying a wide variety of relationships, minimizing the need for manual intervention in terms of annotation or schema design, and achieving high levels of accuracy and coherence in the extracted knowledge. However, this very goal of schema independence introduces inherent challenges related to the consistency and interpretability of the extracted relations (Yates et al., 2007). Additionally, accurately identifying the boundaries of the subject, relation, and object within sentences with complex grammatical structures remains a significant challenge (Niklaus et al., 2018).

Achieving both completeness and accuracy in OIE remains a core challenge due to the inherent complexity and ambiguity of natural language. Relation ambiguity arises because the same underlying relationship can be expressed using a multitude of different phrases and syntactic constructions. Furthermore, correctly interpreting relations often requires understanding the broader context in which they appear. OIE systems must also deal with complex sentence structures, including passive voice, negation, and various forms of modification, which can obscure the underlying relationships (Niklaus et al., 2018).

Evaluating the performance of OIE systems is crucial for measuring progress and comparing different approaches. The standard evaluation metrics of precision, recall, and F1-score are commonly used, where precision measures the proportion of correctly extracted triplets, and recall measures the proportion of true relations in the text that

were successfully extracted. However, a significant challenge in OIE evaluation is that most methods do not rely on an "exact" match of the entire triplet. Instead, they often use more lenient matching strategies, such as comparing only the predicate or the predicate and one of its arguments. Creating gold standard datasets for OIE evaluation is also a particularly challenging task due to the open-ended nature of the domain and the potential for a vast number of valid extractions. Different evaluation approaches exist, including intrinsic evaluation, which directly assesses the quality of the extracted triplets, and extrinsic evaluation, which measures the impact of OIE on the performance of downstream applications. The combination of a lack of universally agreed-upon gold standard datasets and varied evaluation protocols, including different matching strategies, complicates the direct comparison of different OIE systems (Bhardwaj et al., 2019).

## 2.2.2 Evolution of Open Information Extraction Approaches

### Rule-Based Open Information Extraction Systems

Early research in OIE focused on rule-based systems that utilized manually defined linguistic patterns and heuristics to extract relational information. Among the pioneering systems was TextRunner (Yates et al., 2007), which employed a redundancy-based approach, extracting relations that appeared frequently across a large corpus and assigning confidence scores based on statistical measures. This groundbreaking work demonstrated the first practical application of automatically extracting information without predefined schemas.

Building on this foundation, subsequent systems introduced more sophisticated pattern-based extraction techniques. ReVerb (Fader et al., 2011) emphasized linguistic constraints on relation phrases, requiring them to be connected verb phrases next to each other and using syntactic patterns to identify potential subjects and objects. These pattern-based techniques typically involve the use of lexico-syntactic patterns, such as noun-verb-noun or noun-preposition-noun, to identify potential relations and their arguments within a sentence. While offering the advantage of interpretability through human-readable extraction rules, pattern-based methods faced challenges in designing comprehensive rules that could account for the diverse linguistic expressions of the same underlying relation.

Further advancing the field, OLLIE (Schmitz et al., 2012) built upon ReVerb by including an iterative learning process, starting with a small set of seed relations and iteratively discovering new patterns and extracting additional relations from text. Around the same time, dependency parsing approaches emerged as another prominent methodology in rule-based OIE systems (Gamallo et al., 2012). By analyzing the grammatical relationships between words in a sentence, dependency parsing provides a structured representation that can be leveraged to identify subjects, relations, and objects based on their dependency relations. This approach enabled more accurate extraction in complex sentence structures compared to surface-level pattern matching, though its effectiveness remained dependent on the accuracy of the underlying dependency parser.

Despite their early successes, rule-based OIE systems exhibited several limitations (Niklaus et al., 2018). They often lacked robustness to linguistic variations, struggling to recognize the same relation expressed in different ways. Scalability was also a significant issue, as the manual effort required to create and maintain rules for a broad range of relations became impractical for open-domain extraction. Furthermore, rules designed for a specific domain often did not generalize well to other domains with different vocabulary and linguistic conventions. Finally, rule-based systems could be brittle and prone to errors when encountering ambiguous or ungrammatical input that did not conform to the predefined patterns. These inherent limitations ultimately motivated the research community to explore more data-driven approaches based on machine learning.

### Neural Network-Based Open Information Extraction Systems

The integration of neural networks and deep learning has revolutionized OIE, particularly through the adoption of sequence labeling frameworks. Unlike traditional pattern-based methods, these approaches treat OIE as a token-level classification task where models learn to tag words with their respective roles (subject, relation, object) in potential triplets. Early implementations leveraged established sequence modeling techniques like Conditional Random Fields (CRFs) (Stanovsky et al., 2018), but recent advancements have been driven by neural architectures capable of capturing complex contextual patterns without manual feature engineering.

Modern supervised approaches predominantly employ two complementary paradigms: recurrent networks and attention-based transformers. The recurrent architecture family, exemplified by works like IMoJIE (Kolluru et al., 2020), utilizes BiLSTMs to process text bidirectionally, capturing long-range dependencies while maintaining sequential integrity. In contrast, transformer-based models (Zhou et al., 2022) leverage self-attention mechanisms to model global contextual relationships simultaneously across all positions in the sentence. Both architectures automate the discovery of extraction patterns through data-driven learning, enabling nuanced triplet extraction that surpasses rule-based systems in handling syntactic variability.

While these neural approaches demonstrate superior performance, they introduce new challenges. Their data-hungry nature requires large annotated corpora for training, creating a significant dependency on data availability. The creation of benchmark datasets such as CaRB and LSOIE was a key enabler for the success of these supervised models, providing the necessary scale for effective training (Bhardwaj et al., 2019; Solawetz and Larson, 2021). However, this reliance on annotated data creates scalability bottlenecks, and domain adaptation remains problematic—models optimized for specific text genres (e.g., news articles) often suffer performance degradation when applied to technical domains or informal discourse (Cui et al., 2018).

To address the data dependency of supervised learning, researchers have explored the use of self-supervised learning techniques for OIE (Hu et al., 2020b). These methods aim to train OIE models on vast amounts of unlabeled text by defining pretext tasks that

allow the model to learn useful representations of language without explicit annotations. Examples of such tasks include predicting masked words or reconstructing corrupted input. By leveraging the abundance of unlabeled data, self-supervised learning holds the potential to make OIE more scalable and applicable to low-resource scenarios.

Neural network-based OIE systems offer several advantages over their rule-based counterparts. They demonstrate improved robustness to linguistic variations, automatically learn extraction patterns from data, exhibit better scalability and adaptability to new domains, and can handle ambiguity and noise in the input more effectively. These advancements highlight the transformative impact deep learning had on the field of OIE. They enabled the development of more flexible, scalable, and robust OIE systems (Zhou et al., 2022).

### Large Language Model-Based Open Information Extraction Methods

The emergence of LLMs has begun to influence approaches to OIE, representing the latest evolution in OIE methodology. Unlike traditional neural approaches that require specialized architectures and training on OIE-specific datasets, LLM-based methods leverage the broad knowledge and linguistic capabilities encoded in pre-trained models. These approaches typically utilize zero-shot or few-shot prompting techniques to guide LLMs in extracting structured triplets from text without additional training.

Recent work by Ling et al. (2023) has demonstrated that large language models can perform competitive OIE when provided with appropriate prompting strategies. Similarly, Qi et al. (2023) evaluated the performance of models like GPT-3.5-Turbo and LLaMA-2 on OIE tasks, finding that they can match or exceed specialized systems in certain contexts, particularly through dynamic prompt engineering techniques. However, the application of LLMs to OIE remains experimental, with significant challenges including output inconsistency, hallucination of spurious relations, and computational inefficiency compared to dedicated extraction systems.

The detailed potential and limitations of LLM-based approaches to OIE will be explored more comprehensively in Section 2.4, which examines the intersection of LLMs and OIE.

### 2.2.3 Multilingual Open Information Extraction

#### Challenges in Multilingual Open Information Extraction

Extending OIE to languages other than English presents a unique set of challenges (Ro et al., 2020). Language-specific syntactic structures pose a significant hurdle, as grammatical rules and word order vary considerably across languages (Niklaus et al., 2018). Techniques developed primarily for English, which often relies on a relatively fixed subject-verb-object (SVO) order, may not be directly applicable to languages with free or flexible word order (e.g., German, Turkish) or morphologically rich systems (e.g., Finnish, Arabic). This necessitates the development of OIE methods that are either language-

agnostic or specifically tailored to handle syntactic and morphological particularities (Saha et al., 2017).

Another major challenge in multilingual OIE is the uneven availability of resources across languages (Ro et al., 2020). Many non-English languages, particularly low-resource ones, lack large annotated datasets, high-quality linguistic tools (e.g., dependency parsers), and robust pre-trained language models comparable to those available for English. This scarcity impedes both the development and evaluation of OIE systems for these languages (Conneau et al., 2019).

Transfer learning approaches offer a promising solution to the resource gap (Hu et al., 2020a). These methods leverage knowledge from high-resource languages (e.g., English) to improve performance on low-resource languages through cross-lingual word embeddings, multilingual pre-trained models (e.g., mBERT, XLM-R), and parameter-efficient fine-tuning. However, their effectiveness depends on factors like linguistic similarity between source and target languages and the availability of even minimal parallel data (Conneau et al., 2019).

### German-Specific Open Information Extraction Research

To evaluate the LLM-based OIE framework developed in this thesis, German language text is used as the primary test case, specifically from the German legal domain. German-specific OIE research remains underexplored compared to English. German morphological complexity (e.g., case markings like "der Mann" vs. "dem Mann", compound nouns such as "Bahnhofsuhr" from "Bahnhof" + "Uhr") and flexible word order (e.g., verb-final clauses in subordinate sentences like "Ich weiß, dass er morgen kommt") pose distinct challenges for OIE systems designed for English (Akbik and Löser, 2012).

Ro et al. (2020) investigated multilingual neural OIE using BERT-based architectures, demonstrating applicability to German but noting performance gaps due to structural divergences from English. Earlier, Akbik and Löser (2012) developed KrakeN, a language-independent OIE system applicable to German, combining dependency parsing with rules for argument extraction. Their work highlights the necessity of syntactic adaptations for German, such as handling separable verbs (e.g., "ruft...an" in "Er ruft seinen Freund an") and case-driven argument identification. Subsequent efforts like Bassa et al. (2018) further refined rule-based methods specifically for German, addressing challenges unique to the language. Despite progress, German OIE still lacks standardized benchmarks and diverse datasets comparable to English (e.g., CaRB; OIE2016 by Bhardwaj et al. (2019); Stanovsky and Dagan (2016)).

#### 2.2.4 Applications of Open Information Extraction

A primary application for OIE is the automated construction and enrichment of knowledge bases. Extracted tuples can directly populate large-scale knowledge graphs, with entities as nodes and relations as edges (Muhammad et al., 2020). The process also supports ontology learning by discovering new concepts and relationships from text, which

can then be used to populate and refine existing knowledge resources (Zhang et al., 2019).

OIE’s structured output is also crucial for information retrieval and verification tasks. In question answering, the extracted tuples allow for direct semantic matching between a query and factual statements (Song et al., 2023; Khot et al., 2017). They also serve as verifiable factual anchors for retrieval-augmented generation systems, improving the accuracy and grounding of answers (Lewis et al., 2020). This principle of structured matching is central to automated fact-checking, where tuples from a claim are compared against those from source documents to assess its validity (Song et al., 2023). Similarly, by representing content as normalized tuples, OIE enables advanced text comparison for tasks like paraphrase detection and version analysis (Thenmozhi and Kumar, 2018; Zhang et al., 2019). This also extends to extractive summarization, where OIE identifies a document’s key factual statements to create a concise summary.

### 2.2.5 Triplet Matching and Semantic Equivalence

Determining when two extracted triplets convey the same or similar meaning is a fundamental challenge in OIE, particularly given the relation ambiguity and complex sentence structures outlined earlier (Niklaus et al., 2018). Establishing semantic equivalence is complex because it must account not only for superficial lexical similarity but also for deeper semantic relationships such as synonymy, paraphrase, and contextual nuances. While exact lexical matching may indicate a surface-level resemblance, confirming true semantic equivalence requires robust theoretical frameworks and well-designed evaluation mechanisms.

Historically, triplet matching has been primarily employed to evaluate OIE systems, as seen in benchmarks like OIE2016 (Stanovsky and Dagan, 2016) and CaRB (Bhardwaj et al., 2019). In these settings, system outputs are compared against ground-truth triplets using string-based techniques, often involving exact matching alongside approximate (fuzzy) matching methods, to calculate metrics such as precision, recall, and F1-scores. However, this narrow application overlooks the broader potential of triplet matching for downstream tasks, including knowledge base construction, text comparison, and fact verification, where semantic alignment of triplets is crucial.

Various approaches have been proposed to address the matching challenge. Traditional methods rely on lexical comparisons, wherein the subject, relation, and object components are matched either exactly or using fuzzy matching techniques that incorporate normalization, stemming, or edit-distance measures. Although effective for capturing surface-level similarities, these approaches often fall short in recognizing synonymous phrases, paraphrases, or subtle variations in meaning, echoing the limitations of rule-based OIE systems (Niklaus et al., 2018).

To address these shortcomings, embedding-based similarity methods have emerged as a more sophisticated solution. These methods represent triplet components or entire triplets as vectors in a continuous semantic space using word or sentence embeddings.

Similarity between triplets can then be measured by aggregating the component similarities (e.g., via cosine similarity) or by directly comparing composite triplet embeddings, thereby capturing richer semantic nuances and contextual dependencies. This approach aligns with the advancements in neural network-based OIE systems, which leverage deep learning to model complex linguistic patterns (Zhou et al., 2022).

Recent advances in LLMs open new avenues for semantically matching triplets. By leveraging their deep contextual understanding and reasoning capabilities, LLMs hold the promise of overcoming limitations inherent in both traditional lexical methods and conventional embedding-based approaches. Surveys like Xu et al. (2024) highlight LLMs' potential in generative information extraction tasks, suggesting they could be used to determine semantic equivalence between triplets. However, fully realizing these benefits remains an active area of research, with challenges such as output inconsistency noted in LLM-based OIE methods.

In the context of OIE applications, effective triplet matching is essential. For knowledge base construction, matching triplets helps avoid duplicates and ensures consistency across the knowledge graph (Muhammad et al., 2020). In text comparison, matching triplets enables semantic alignment beyond lexical overlap, facilitating tasks like version analysis and survey generation (Zhang et al., 2019). For fact verification, matching triplets between claims and evidence is key to assessing veracity and combating misinformation (Song et al., 2023). These applications underscore the practical significance of triplet matching beyond evaluation.

## 2.3 Large Language Models

LLMs have fundamentally transformed the field of NLP, enabling unprecedented capabilities in text understanding and generation. This section outlines their evolution, technical foundations, capabilities, and their inherent limitations.

### 2.3.1 Evolution and Background

#### Historical Development

The development of LLMs represents a significant shift in the natural language processing paradigm. Early neural approaches to language modeling primarily utilized recurrent architectures such as Recurrent Neural Networks (RNNs) (Mikolov et al., 2010) and Long Short-Term Memory networks (LSTMs) (Sundermeyer et al., 2012). These models, while groundbreaking at the time, suffered from limitations in capturing long-range dependencies and scaling efficiently.

The introduction of the Transformer architecture by Vaswani et al. (2017) marked a pivotal turning point, enabling parallel processing of sequences and more effective modeling of long-range dependencies through self-attention mechanisms. This architectural innovation catalyzed rapid progress, leading to increasingly capable models. Key milestones in this evolution include BERT (Devlin et al., 2019), which pioneered bidirectional contextual representations; T5 (Raffel et al., 2020), which reformulated NLP tasks as text-to-text problems; and GPT series models culminating in GPT-3 (Brown et al., 2020), which demonstrated remarkable few-shot learning capabilities.

This progression reflects a fundamental shift from developing task-specific architectures to general-purpose language models that can be adapted to a wide range of applications with minimal task-specific training (Bommasani et al., 2021). The transition has been enabled by advances in computational resources, dataset scale, and algorithmic innovations that allow models to effectively utilize vast amounts of text data (Zhao et al., 2023).

#### Core Architectural Innovations

The Transformer architecture remains the foundation of modern LLMs, with its self-attention mechanism enabling models to dynamically emphasize relevant parts of input sequences (Vaswani et al., 2017). This mechanism allows each token in a sequence to attend to all other tokens, facilitating the capture of complex linguistic patterns and dependencies regardless of their distance in the text.

Positional encoding techniques address the inherent permutation invariance of self-attention, providing models with information about token positions in sequences. Various approaches have been developed, from the original sinusoidal encodings (Vaswani et al., 2017) to learned absolute positional embeddings, relative positional embeddings (Shaw et al., 2018), and modern rotary position embeddings (RoPE) (Su et al., 2024).

Scaling laws have emerged as another critical aspect of LLM development. While Kaplan et al. (2020) initially demonstrated relationships between model size and performance, subsequent work by Hoffmann et al. (2022) established that optimal scaling requires balancing model size, dataset size, and computational budget through chinchilla-optimal training. These findings suggest that continued improvements require coordinated scaling of multiple factors rather than simply increasing parameter counts.

The emergence of qualitatively new capabilities at scale remains an active research area. While increasing parameter count, models exhibit behaviors like in-context learning (Brown et al., 2020) and chain-of-thought reasoning (Wei et al., 2022).

### 2.3.2 Technical Foundations

#### Model Architecture and Pre-training

Modern LLMs are predominantly based on the Transformer architecture, which consists of multiple layers of self-attention and feed-forward neural networks. The architecture can be configured in different ways, leading to decoder-only models (like GPT), encoder-only models (like BERT), or encoder-decoder models (like T5). Each configuration offers distinct advantages for different types of NLP tasks, with decoder-only models being particularly suited for text generation and completion tasks relevant to OIE applications.

The pre-training objectives for transformer-based language models fall broadly into two categories: autoregressive and autoencoding approaches. Autoregressive models like the GPT series (Radford et al., 2018) are trained to predict the next token in a sequence given all previous tokens, effectively modeling the probability distribution of text. This approach naturally supports generative tasks but provides unidirectional context.

In contrast, autoencoding models like BERT (Devlin et al., 2019) employ masked language modeling objectives, where random tokens in the input are masked and the model is trained to reconstruct them based on bidirectional context. This approach yields rich contextual representations particularly suited for understanding tasks but requires additional adaptation for generation.

Text-to-text models like T5 (Raffel et al., 2020) unify these approaches by framing all NLP tasks as text generation problems, with task-specific formatting of inputs and outputs. This framework simplifies multi-task learning and transfer, as the model architecture remains consistent across applications.

Recent advances include multimodal pre-training, where models are trained on combined datasets of text and other modalities such as images (Radford et al., 2021), or code (Chen et al., 2021). These approaches expand the models' representational capabilities and application domains, enabling cross-modal reasoning and generation.

## Prompt Engineering

Prompt engineering has emerged as a critical methodology for effectively utilizing pre-trained LLMs. Zero-shot prompting involves crafting instructions that enable models to perform tasks without examples, relying on their pre-trained knowledge. Few-shot prompting extends this approach by including demonstrative examples within the prompt, providing explicit patterns for the model to follow (Brown et al., 2020).

The effectiveness of prompting strategies varies significantly based on wording, formatting, and the inclusion of specific elements like chain-of-thought reasoning instructions. However, research demonstrates that apparent reasoning capabilities may depend heavily on example selection and task structure. Several factors influence prompt effectiveness, including task formulation clarity, example selection for few-shot settings, and explicit instructions for desired reasoning processes. Despite advances in systematic prompt design, variability in model responses remains an ongoing challenge for applications requiring deterministic outputs (Sahoo et al., 2024).

### 2.3.3 Capabilities and Applications of Large Language Models in Natural Language Processing

LLMs represent a fundamental shift from specialized, task-specific models to unified systems capable of in-context learning. Their versatility stems from a powerful text generation capability, refined through techniques like instruction tuning and reinforcement learning from human feedback (RLHF) to produce fluent and controllable outputs (Ouyang et al., 2022). This allows them to address a wide range of NLP tasks with minimal specific training, including text summarization (Raffel et al., 2020), classification via zero-shot prompting, and multilingual machine translation (Xue et al., 2020).

Beyond these applications, LLMs also handle tasks requiring more complex or structured outputs. In question answering (QA), they perform both extractive and generative reasoning, often enhanced by retrieval systems to ensure factual reliability (Lewis et al., 2020; Zhao et al., 2023). Crucially for this thesis, they can perform information extraction by identifying and structuring relations from text using prompts alone (Liu et al., 2022). This capability extends to sophisticated reasoning and inference, where techniques like chain-of-thought prompting enable models to solve multi-step problems that require logical deduction (Wei et al., 2022).

### 2.3.4 Technical Limitations

Despite their capabilities, LLMs face significant technical limitations. The computational requirements for both training and inference present barriers to widespread deployment, particularly in resource-constrained environments (Bender et al., 2021). State-of-the-art models require substantial GPU/TPU resources, limiting their accessibility and increasing operational costs. Temporal knowledge cutoff represents another inherent limitation, as models cannot access information beyond their training data (Liska et al., 2022).

Perhaps most critically for extraction tasks, LLMs lack explicit reasoning traces and explainability. While their outputs may be accurate, the processes by which they arrive at these conclusions remain largely opaque (Zhao et al., 2024).

### 2.4 Large Language Models for Open Information Extraction and Triplet Matching

The application of LLMs to OIE represents a significant area of research, aiming to transform unstructured textual data into structured triplets. Furthermore, LLMs are being investigated for their utility in semantic triplet matching, the task of discerning semantic equivalence between different triplet formulations. This section outlines the inherent strengths LLMs bring to OIE, discusses the considerable challenges that accompany their deployment, and examines their role in semantic triplet matching.

LLMs possess several characteristics that are advantageous for OIE tasks. Their comprehensive linguistic knowledge, acquired from training on extensive text corpora, enables the recognition of diverse syntactic patterns that express similar semantic relationships (Liu et al., 2022; Xu et al., 2024). This allows for a nuanced understanding that can extend beyond clausal boundaries, capturing relations that span multiple sentences, a capability often challenging for traditional OIE systems (Dagdelen et al., 2024; Qi et al., 2023). Moreover, the few-shot learning capabilities inherent in many LLMs significantly reduce the dependency on large, meticulously annotated datasets that are typically required for supervised neural OIE methods (Brown et al., 2020). This facilitates quicker adaptation to new domains or specific extraction tasks. Another notable strength is the innate multilingualism of many contemporary LLMs, which provides pathways toward cross-lingual information extraction without the need for extensive language-specific engineering or separate models for each target language (Li et al., 2024).

Despite their potential, the deployment of LLMs for OIE is fraught with significant hurdles. A primary concern is the propensity for LLMs to hallucinate, generating plausible but factually incorrect triplets that are not substantiated by the source text (Huang et al., 2025). This undermines the reliability of the extracted knowledge. Output inconsistency is another persistent issue; minor variations in prompting can lead to disparate extraction results, making it challenging to achieve stable and reproducible outcomes (Ashok and Lipton, 2023). Computational efficiency and scalability also present major barriers. The inference process for LLMs is resource-intensive compared to specialized OIE architectures, potentially limiting their applicability for large-scale tasks that involve processing millions of documents (Ding et al., 2023; Lin et al., 2024). The opaque, black-box nature of LLMs further complicates reliability assessment and error analysis, as the reasoning behind specific extractions is difficult to trace, although improving interpretability remains an active area of research (Singh et al., 2024). Finally, the field currently lacks standardized evaluation protocols and benchmarks specifically designed for LLM-based OIE, which impedes robust comparisons and systematic progress (Bhardwaj et al., 2019).

In addition to extraction, using LLMs for semantic triplet matching remains largely unexplored. Unlike traditional methods that combine lexical and statistical approaches or contextual embeddings, LLMs theoretically could leverage broader contextual understanding for paraphrase recognition. If developed effectively, such capabilities might eventually aid knowledge base population and redundancy reduction. However, current OIE evaluation methodologies primarily measure syntactic agreement (Bhardwaj et al., 2019) rather than semantic equivalence, making performance claims difficult to verify.

To navigate these challenges, current research explores various strategies for integrating LLMs into OIE pipelines. These include direct end-to-end extraction, hybrid architectures, effective prompt engineering with clear constraints (Li et al., 2023; Ling et al., 2023; Wei et al., 2023), and parameter-efficient fine-tuning (PEFT) techniques. Compared to traditional OIE systems, LLM-based approaches tend to leverage a broader contextual understanding for processing complex syntax rather than relying on predefined patterns. While they typically require less task-specific training data and exhibit better performance transfer across different domains, they often suffer from higher inference costs and less deterministic output. These differing characteristics suggest potential for complementary use, leading to explorations of hybrid systems. Key future research directions include the development of robust mechanisms for factuality verification to counter hallucinations, strategies to improve computational efficiency and scalability, and the establishment of comprehensive, standardized evaluation frameworks specifically for LLM-based OIE and triplet matching (Zhao et al., 2023).



# CHAPTER 3

## Use case

### 3.1 Overview

In German-language legal education, students are required to analyze complex legal problem scenarios and to construct structured written responses grounded in statutory law. These student responses need to include precise legal analysis and correct application of legal norms to case-specific facts.

Traditionally, such student responses are evaluated manually by legal educators or examiners. While this method offers depth and expert insight, it is also fraught with challenges. Subjectivity in interpretation can lead to inconsistent grading, especially in cases where responses are semantically correct but differently worded. Moreover, manual assessment is time-consuming and labor-intensive, particularly for large student cohorts, making scalability a significant issue.

This system is based on a use case developed from materials provided by a legal textbook publisher. The system's aim is to support criterion-based assessment by evaluating how well students apply key legal criteria to case-specific facts. This chapter introduces the legal and educational setting, the dataset structure, and the practical constraints that inform the development of the automated assessment framework.

### 3.2 Structure of the Legal Task and Assessment Dataset

#### 3.2.1 Representative Case: *Fanny und das Fahrrad*

To illustrate the use of the solution schema and the automated assessment task, this thesis focuses on one representative example: the fictional legal case *Fanny und das Fahrrad*. In this scenario, a woman named Fanny purchases a bicycle from Paula, who is in possession of the bicycle but is not the legal owner. Paula, however, presents herself

as the owner and sells the bicycle to Fanny for a reasonable price. Students are asked to determine who owns the bicycle.

Although this case serves as a central example in the present work, it is part of a larger and extensible collection of legal problem cases. Each case is designed to target a specific legal doctrine and is annotated in the same structured way. This standardization is crucial for the framework's ability to generalize to a wide range of legal scenarios.

Erwin schuldet Paula Geld und hat ihr daher zur Sicherheit sein Fahrrad verpfändet und übergeben. Als Fanny bei Paula auf Besuch ist und das Fahrrad sieht, möchte sie es unbedingt haben. Paula, die sich als Eigentümerin ausgibt, verkauft Fanny das Fahrrad um angemessene 100, und übergibt es ihr. Wem gehört das Fahrrad?

Figure 3.1: The representative case *Fanny und das Fahrrad*.

#### 3.2.2 Structured Solution Schemas

Each legal problem case is associated with a solution schema developed by subject matter experts. These schemas define the expected reasoning path a student should follow to arrive at a correct legal analysis. These components are categorized into three classes:

- Legal theory: General legal rules such as statutory provisions and doctrinal requirements.
- Application to facts: Concrete application of legal principles to the facts of the case.
- Supplementary information: Structuring or explanatory elements that are encouraged but not required.

For *Fanny und das Fahrrad*, the solution schema includes a range of expectations, each associated with a specific point value. The original solution schema is shown in Figure 3.2. Among the criteria are:

- The recognition that a derivative transfer of ownership is not possible.
- The identification of a valid legal title ("Kaufvertrag") and a mode of transfer ("Übergabe").
- Classification of the object (bicycle) as a movable good ("bewegliche Sache").
- Establishing that the transaction was remunerated ("entgeltlich"), supported by facts such as the payment of a price or the formation of a purchase contract.

- An argument for Fannys good faith ("Redlichkeit"), based on the absence of contrary indications in the case facts.
- The selection of the correct alternative condition from §367 ABGB, in this case, acquisition from a person of trust ("Vertrauensmann").

Some criteria are mandatory but not scored (e.g., identifying that §367 ABGB governs the situation), while others are each worth 0.5 or 1 point depending on their complexity and relevance. Multiple correct phrasings are permitted for each criterion, and the schema accounts for common synonyms and paraphrased expressions.

### 3. USE CASE

---

1. Es ist ein gutgläubiger Eigentumserwerb zu prüfen (Kein Punkt, soll aber in der Lösung enthalten sein)
2. Es ist kein derivativer Eigentumserwerb möglich (1 P)
3. Es liegen Titel (Kaufvertrag) und Modus (Übergabe) (1 P)
4. Das Fahrrad ist eine bewegliche Sache (1/2 P)
5. Ein Entgeltliches Rechtsgeschäft liegt vor, weil
  - Ein Kaufvertrag geschlossen wurde
  - Ein Kaufpreis gezahlt wurde
  - 100 Euro gezahlt wurden
  - Einer der drei Gründe genügt! (1/2 P)
6. Redlichkeit liegt vor, weil
  - Ein angemessener Preis gezahlt wurde/ der Kaufpreis ist angemessen/ 100 EURO sind ein angemessener Preis für das Rad
  - Paula gibt sich als Eigentümerin des Rads aus, Paula behauptet Eigentümerin zu sein UND es gibt keine Angaben im Sachverhalt warum Fanny daran Zweifel sollte, dass Paula nicht die Wahrheit sagt.
  - Wenn sich aus dem Sachverhalt keine gegenteiligen Anhaltspunkte ergeben, ist stets von Redlichkeit auszugehen
  - Einer der drei Gründe genügt! (1/2 P)
7. Von den 3 Alternativvoraussetzung
  - Erwerb in der öffentlichen Versteigerung
  - Erwerb vom Unternehmer im gewöhnlichen Betrieb seines Unternehmens
  - Erwerb vom Vertrauensmann
  - liegt im konkreten Fall der Erwerb vom Vertrauensmann vor weil Paula Pfandgläubigerin von Erwin ist (1/2 P)
8. Feststellung, dass Fanny gutgläubig Eigentum erworben hat (Kein Punkt, soll aber in der Lösung enthalten sein)

Figure 3.2: The solution scheme for the case *Fanny und das Fahrrad*.

### 3.2.3 Student Responses

The dataset contains written responses to the case *Fanny und das Fahrrad*, five of these responses were annotated as described in the next section, one of those is shown as an example in Figure 3.3. These responses are composed in German and vary in linguistic style and structural organization. Most texts range between 250 and 400 words.

Responses often include domain-specific linguistic phenomena such as:

- Legal abbreviations (e.g., "gem. §", "Abs."),
- Compound nouns (e.g., "Eigentumserwerb", "Kaufvertragserfüllung"),
- Latin phrases (e.g., "bona fide"),
- Bullet-pointed or appositional sentence structures (e.g., "Entgeltlich: Kaufvertrag (100€)").

These features present challenges for information extraction and sentence segmentation, and they necessitate customized preprocessing to ensure accurate semantic interpretation.

### 3.2.4 Annotations and Ground Truth

Each student response is manually evaluated by legal experts. For each criterion defined in the schema, a binary label is applied to indicate whether the student has adequately addressed the requirement. For example, according to the expert annotation, the student response shown in Figure 3.3, fulfills all of the criteria listed in Figure 3.2, except for criterion number 2. These annotations serve as the ground truth for system training and evaluation. The dataset includes multiple student responses with detailed expert annotations, designed to support evaluation.

## 3.3 System Requirements and Challenges

An automated assessment system for this task must address several domain-specific challenges. A primary challenge is semantic matching, as students often express the same legal point using different words or syntactic constructions. For instance, the concept of an "entgeltliches Rechtsgeschäft" can be phrased as "ein Kaufvertrag wurde abgeschlossen" or "100 wurden gezahlt." A robust system must detect this equivalence to ensure fair scoring. This task is further complicated by the high linguistic density of German legal writing, which features compound structures, flexible word order, and legal jargon that challenge standard parsing and extraction models.

Furthermore, the systems scoring must be both robust and transparent. This requires matching to go beyond simple keywords to evaluate the underlying legal reasoning and

§367 ABGB schützt den gutgläubigen Erwerber. Redlich ist gem. §368 Abs. 1 nur, wer den Veräuerer aus wahrscheinlichen Gründen für den Eigentümer halten konnte. §367 führt zum originären Eigentumserwerb, er spricht von beweglichen Sachen. Der Erwerb muss ferner auf einem objektiv gültigen Titelgeschäft zwischen Veräuerer und Erwerber beruhen, und entgeltlich sein. Zusätzlich muss eine der besonderen Voraussetzungen des §367 gegeben sein:

- Erwerb in einer öffentlichen Versteigerung
- Von einem Unternehmer im gewöhnlichen Betrieb seines Unternehmens
- Von einem Vertrauensmann (wer vom Eigentümer die Gewahrsame an der Sache übertragen bekommen hat)

Gutgläubigkeit von Fanny: Laut SV gibt sich Paula als Eigentümerin aus und keine gegenteiligen Anhaltspunkte sind gegeben, weswegen Fanny Paula aus wahrscheinlichen Gründen nicht für die Eigentümerin halten dürfe. Sie ist somit gutgläubig.

Titel: Kaufvertrag zwischen Fanny und Paula

Entgeltlich: Kaufvertrag (100€)

Bewegliche Sache: Fahrrad

Vertrauensmann: Paula hat von Erwin, dem bisherigen Eigentümer die Gewahrsame aufgrund der Verpfändung erhalten und gilt deswegen als Vertrauensmann des Erwin gegenüber Fanny.

Fanny erwirbt aufgrund dessen originär Eigentum am Fahrrad. (§367 ABGB)

Figure 3.3: An example student answer for the case *Fanny und das Fahrrad*.

its connection to case facts. The system must also accommodate multiple valid justifications for a single criterion and distinguish between its partial and complete fulfillment. Finally, while this thesis focuses on one representative case, a key requirement for the framework is generalization. Each new case introduces variations in vocabulary and fact patterns. The system must therefore be robust to this variation, recognizing recurring legal structures to ensure its effectiveness across diverse contexts.

## 3.4 Outlook

This chapter has detailed the use case for automated legal assessment, outlining both its pedagogical value and its technical complexity. The structured dataset, expert annotations, and reusable solution schemas provide the foundation for building a scalable feedback system. However, linguistic and semantic variability present significant obstacles that the proposed framework must address.

The ultimate goal is to develop a general framework to handle a wide range of legal

cases. The next chapter will introduce the methodology used in our framework to extract structured knowledge from student responses and align it with the predefined legal criteria.



# Methodology

## 4.1 Overview

This chapter details the design, implementation, and evaluation strategy of the proposed framework for LLM-based OIE, triplet matching, and text comparison. First, we give an overview of the developed framework, and its components. Then the dataset, the tested LLMs, and other methods are detailed, followed by an explanation of the implementation and experimental settings. Finally, we describe the evaluation methodology.

## 4.2 Developed Framework

The framework is implemented as a modular Python system designed for flexibility and extensibility. Its central goal is to employ the natural language understanding capabilities of LLMs to extract structured information, specifically subject-relation-object triplets, from unstructured text. These extracted triplets subsequently facilitate semantic text comparison, allowing the assessment of textual content against predefined criteria represented as target triplets. The system design relies heavily on LLMs as the core engine for both information extraction and semantic matching tasks.

### 4.2.1 Terminology

For clarity, the following terms are defined based on their usage within the framework:

- **Input text:** The raw text document (e.g., a student's answer) supplied to the system.
- **Extracted triplets / Candidate triplets:** The set of (subject, relation, object) triplets identified and generated from the input text by the OIE component. These are then used as candidate triplets for comparison in the matching phase.

- **Target triplets:** Triplets manually defined by the authors based on expert annotations of the legal case solution criteria in the evaluated German legal dataset. Each triplet represents the specific information sought by a given criterion, serving as the ground truth target for the matching step.
- **Matching:** The process of comparing a target triplet against the candidate triplets from an input text to ascertain if the target information is present, done by checking if there is at least one matching candidate triplet for a given target triplet.
- **Matched triplets:** For a given target triplet, this refers to the target itself paired with the subset of candidate triplets identified as semantically equivalent.
- **Reference matches:** The human-annotated matches indicating whether each input text fulfills each criterion. It is the ground truth matching used to evaluate the matchings resulting from automatic methods.

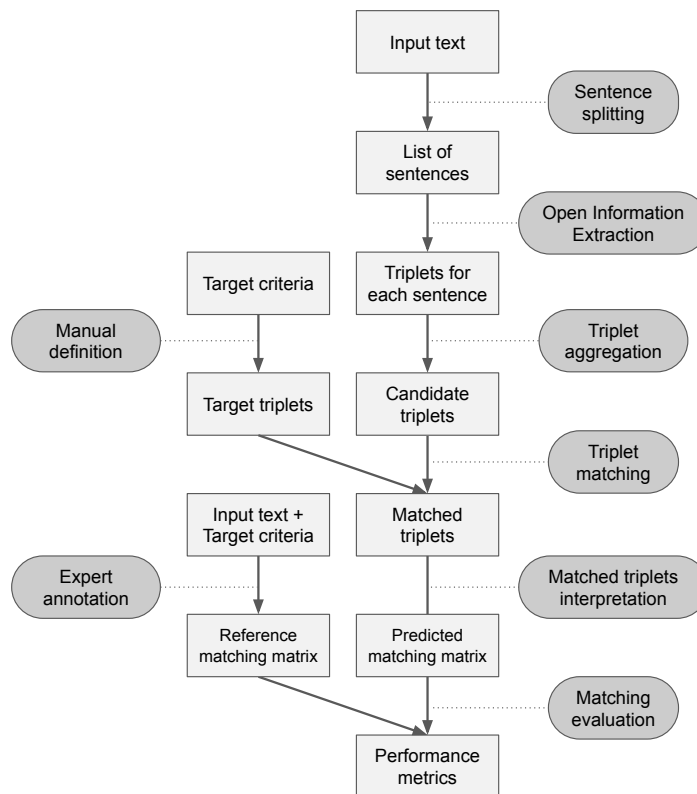


Figure 4.1: A conceptual overview of the primary LLM-based pipeline. The flowchart illustrates the key processing stages from raw text input to matched triplet generation, including sentence segmentation, LLM-based Open Information Extraction (OIE), and triplet matching.

### 4.2.2 System Components and Interaction Flow

The framework comprises several interacting software modules, including components for Open Information Extraction, triplet matching, and matching evaluation, and implementations of other methods (detailed in Section 4.9) for comparison.

The core interaction flow for the main LLM-based method proceeds as follows:

1. **Initialization:** Load input texts (e.g., student answers) and predefined target triplets (representing criteria) from data files. Initialize the selected LLM interface.
2. **Sentence Segmentation:** For each input text, split the raw string into a list of individual sentences using a rule-based sentence boundary detection approach.
3. **Open Information Extraction:**
  - Iterate through each sentence obtained from the previous step.
  - For each sentence, format an OIE-specific prompt containing the sentence text and instructions for triplet extraction.
  - Send the prompt to the initialized LLM via the LLM integration component.
  - Receive the LLM's textual response, which is expected to contain zero or more triplets in a specified string format.
  - Parse the LLM's response string using regular expressions to extract the subject, relation, and object for each identified triplet, creating `Triplet` objects.
4. **Candidate Triplet Aggregation:** Collect all `Triplet` objects extracted from all sentences within a single input text into one comprehensive list of candidate triplets for that text.
5. **Triplet Matching:**
  - Iterate through each predefined target triplet.
  - For the current target triplet, format a matching-specific prompt containing this single target triplet and the entire list of candidate triplets aggregated from the input text.
  - Send the matching prompt to the LLM.
  - Receive the LLM's textual response, expected to contain the subset of candidate triplets that semantically match the target triplet, or an indicator of no matches (e.i., "NO MATCHES").
  - Parse the LLM's response to identify the matching candidate `Triplet` objects. Store the result as a `TripletMatch` object, linking the target triplet to its identified candidate matches (which may be an empty list).

6. **Result Compilation (Matched Triplets Interpretation):** Collect all `TripletMatch` objects for each input text. To prepare for evaluation, convert these results into a boolean matrix, hereafter referred to as the predicted matching matrix. In this matrix, rows represent the input texts and columns represent the target criteria. A cell value of `True` indicates that the framework found a match for a given criterion in a given text, while `False` indicates no match was found.
7. **Matching Evaluation:** Load the human-annotated reference matching matrix, which serves as the ground truth. Compare the predicted matching matrix (generated in the previous step) cell-by-cell against the reference matching matrix. This comparison yields the counts of True Positives, True Negatives, False Positives, and False Negatives, which are then used to calculate the final performance metrics.
8. **Output Generation:** Save detailed outputs, including extracted triplets, matching decisions with supporting candidates, the final boolean matrix, and calculated performance metrics, to output files for analysis.

Figure 4.1 provides a conceptual overview of these stages, while Figure 4.2 illustrates the entire data flow with a concrete example, tracing a sample sentence from input to final evaluation.

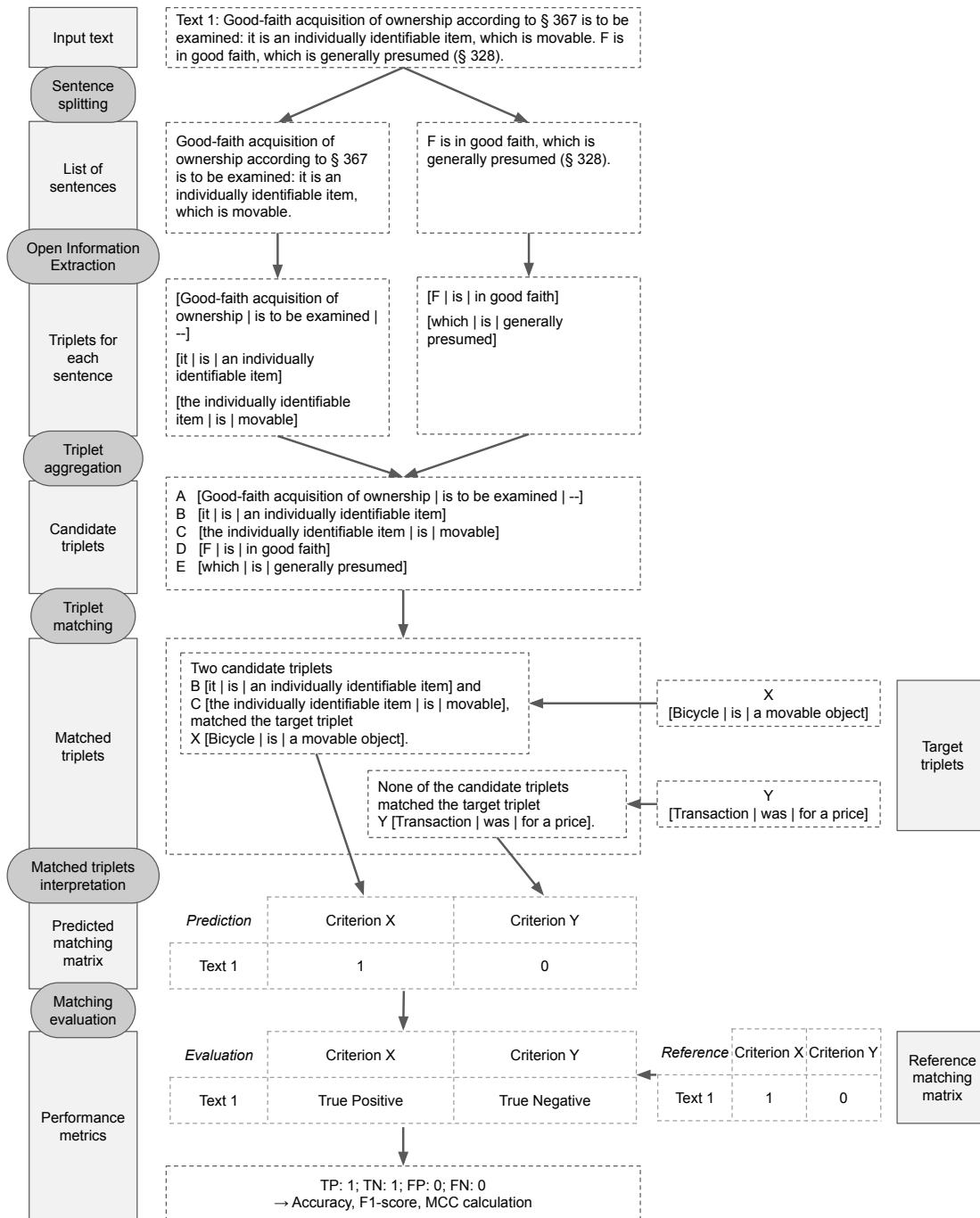


Figure 4.2: A detailed illustration of the pipeline's data flow using a concrete example. The flowchart traces a sample input text ("Good-faith acquisition...") through each processing stage. It shows the transformation from a sentence into candidate triplets, the comparison of these against target triplets, and the resulting interpretation into a cell of the predicted matching matrix, which is then evaluated against the reference matrix.

### 4.3 LLM Integration Component

A dedicated component manages interactions with different LLMs, providing a unified interface that abstracts the underlying mechanism, whether it be API calls or local model inference. This component dynamically selects the appropriate provider (e.i., OpenAI, Together AI, or a local vLLM instance) based on configuration, API key availability, and model compatibility checks defined within the class logic.

For local model deployment, the framework utilizes the vLLM library. It attempts to start a vLLM OpenAI-compatible server as a background process and communicates with it via a standard OpenAI client interface directed at the local server endpoint.

For cloud-based models accessed via APIs, the framework employs the official Python client libraries provided by OpenAI and Together AI. The integration component initializes the correct client, retrieves credentials securely from environment variables, formats API requests using chat completion endpoints, manages the interaction, and handles basic error reporting. Standard LLM parameters like maximum token limits and temperature are configured and passed with each request.

### 4.4 Prompt Design

The performance of our LLM-based framework hinges on well-designed prompts for two key tasks: OIE and triplet matching. These prompts guide the LLM to extract and match structured information from German legal texts effectively. This section details the structure, and purpose of both prompts.

#### 4.4.1 Open Information Extraction Prompt

The OIE prompt (Listing 4.1) directs the LLM to extract subject-relation-object triplets from individual sentences, outputting them in the format [subject | relation | object]. It defines the LLM's role (e.i., "natural language processing expert"), clearly outlines the task (extract subject-relation-object triplets), specifies the precise output format (e.g., '[subject | relation | object]'), and includes constraints (e.g., "output only the triplets"). To further guide the LLM and demonstrate triplet extraction, few-shot examples are included within the prompt template.

```
You are a natural language processing expert. Extract all subject-relation-
object triplets from the following sentence. Use this format: [subject |
relation | object]
Output only the triplets, no other text.

EXAMPLES:
Sentence:
Ozeanwasser in der Nähe der Oberfläche löst Kohlendioxid aus der Atmosphäre auf.

Triplets:
```

```
[Ozeanwasser in der Nähe der Oberfläche | löst auf | Kohlendioxid aus der
Atmosphäre]

Sentence:
UN-Generalsekretär Ban Ki-Moon sagte: Ich fordere, dass alle politischen,
militärischen und Milizenführer die Feindseligkeiten einstellen und die Gewalt
gegen Zivilisten beenden.
Triplets:
[UN-Generalsekretär Ban Ki-Moon | fordere | dass alle politischen,
militärischen und Milizenführer die Feindseligkeiten einstellen und die Gewalt
gegen Zivilisten beenden]
[alle politischen, militärischen und Milizenführer | stellen ein |
Feindseligkeiten]
[UN-Generalsekretär Ban Ki-Moon | sagte | Ich fordere, dass alle politischen,
militärischen und Milizenführer die Feindseligkeiten einstellen und die Gewalt
gegen Zivilisten beenden]

Sentence:
Mit der Entwicklung leistungsstärkerer Mikroskope wurden Viren entdeckt, und
sogar Atome wurden schließlich sichtbar.
Triplets:
[Viren | entdeckt | Mit der Entwicklung leistungsstärkerer Mikroskope]
[Atome | wurden | Mit der Entwicklung leistungsstärkerer Mikroskope sichtbar]

Now, please process the following sentence:
Sentence:
{{sentence}}
Triplets:
```

Listing 4.1: Few-shot LLM prompt used for triplet extraction

#### 4.4.2 Triplet Matching Prompt

The Triplet Matching prompt (Listing 4.2) tasks the LLM with comparing a predefined target triplet to a list of candidate triplets, identifying those that are semantically equivalent. The output is either a list of matching triplets or "NO MATCHES" if none align. Like the OIE prompt, it assigns the "natural language processing expert" role and relies on few-shot examples to demonstrate matching logic.

```
You are a natural language processing expert.
You will receive one target triplet and one or more candidate triplets in the
following format.
[subject | relation | object]

Output all of the candidate triplets that match the target triplet closely
enough. It is also possible that none of them or many of them match.
Do not make any changes to the candidate triplet you chose. Do not output any
other text than the candidate triplets that match the target triplet or if none
match, the text "NO MATCHES".
```

The candidate triplets come from a text, our goal is to check whether or not they contain key information from the reference text, which is represented by the target triplet.

EXAMPLES:

Target triplet:

[Vorliegen | von | Angebot und Annahme]

Candidate triplets:

[Es liegt vor | ein | Angebot und eine Annahme]

[Angebot | ist nicht | Annahme]

[Es gibt | ein | Angebot]

[Die Parteien | haben | einen Kaufvertrag geschlossen]

[Vorliegen | einer | Willenserklärung]

[Angebot | und | Annahme | sind vorhanden]

Candidate triplets that match the [Vorliegen | von | Angebot und Annahme]

target triplet:

[Es liegt vor | ein | Angebot und eine Annahme]

[Es gibt | ein | Angebot]

[Angebot | und | Annahme | sind vorhanden]

Target triplet:

[Verjährung | tritt ein nach | drei Jahren]

Candidate triplets:

[Verjährung | beginnt ab | Vertragsbruch]

[Frist | beträgt | fünf Jahre]

[Klage | muss erhoben werden | innerhalb von zwei Jahren]

[Verjährung | kann unterbrochen werden | durch Anerkennung]

Candidate triplets that match the [Verjährung | tritt ein nach | drei Jahren]

target triplet:

NO MATCHES

Now, please process the following sentence:

Target triplet:

{{target\_triplet}}

Candidate triplets:

{{candidate\_triplets}}

Candidate triplets that match the {{target\_triplet}} target triplet:

Listing 4.2: Few-shot LLM prompt used for triplet matching

## 4.5 Open Information Extraction Component

This component performs the OIE task, primarily using an LLM-driven approach, although a rule-based alternative is available for comparison.

The core LLM-based methodology involves prompting a selected language model to identify and structure information. Input text is first divided into sentences using a

simple rule-based segmentation routine. For each sentence, the OIE component formats a detailed prompt containing the sentence and specific instructions. This prompt is sent to the LLM interface, which returns a textual response expected to contain the extracted triplets. This approach leverages the LLM's inherent language processing capabilities without requiring task-specific model training.

The LLM generates a textual response containing triplets. To transform this unstructured text into a usable format for further processing, the system employs a parsing mechanism designed to extract the subject, relation, and object components from each triplet.

The parsing process begins by analyzing the LLM's output to identify triplet structures, extracting subject, relation, and object based on their positions relative to the delimiters. Designed for robustness, it accommodates formatting variations like extra spaces while systematically capturing all triplets, even in multi-triplet outputs (e.g. [Subject1 | Verb1 | Object1] [Subject2 | Verb2 | Object2]). This step focuses purely on structural extraction, omitting semantic validation or interpretation, as the system trusts the LLM to generate meaningful triplets. Minimal normalization (e.i. trimming whitespace) ensures clean data without altering the LLM's original content.

Once extracted, these components are used to create structured `Triplet` objects, which encapsulate the subject, relation, and object as distinct elements. This conversion ensures that the unstructured text response becomes a collection of organized data entities that the system can easily manipulate and analyze in subsequent steps, such as triplet matching.

This separation of generation and parsing provides modularity, allowing the framework to utilize different LLMs or prompt designs, provided the output conforms to the expected triplet format. By converting the textual response into structured objects, the system prepares the extracted information for the subsequent step, triplet matching.

## 4.6 Triplet Matching Component

This component assesses the semantic equivalence between a target triplet and candidate triplets extracted from the text, predominantly using an LLM. A simpler string-matching alternative is also implemented.

The LLM-based matching algorithm proceeds iteratively for each target triplet. In each iteration, the component constructs a dedicated prompt that pairs a single target triplet with the full set of candidate triplets (as described in 4.4). This one-target-per-prompt design results in multiple LLM calls for each target triplet being evaluated. This approach was chosen to maximize matching accuracy by allowing the LLM to focus on a single, well-defined comparison. An alternative, "batch" approach of using a single prompt to match all target triplets against all candidates simultaneously was considered. While such a method would improve speed and reduce computational cost, it was deferred due to the risk of lower accuracy caused by more complex queries.

The classification of a match versus a non-match for a given criterion is determined based on the LLM’s response. The system parses the LLM’s output (again using the regular expression approach described for OIE parsing) to retrieve the list of identified matching candidate triplets. If this list is non-empty, the criterion is classified as met (True); if the list is empty (or the LLM returns the "NO MATCHES" string), it signifies no match (False). This binary decision is derived directly from the presence or absence of matched candidates returned by the LLM, effectively relying on the LLM’s interpretation of the matching prompt as the decision threshold.

## 4.7 Dataset

This research employs the German legal case dataset described in Chapter 3, specifically focusing on the case *Fanny und das Fahrrad*. The dataset consists of 5 student responses to this case, each manually expert-annotated with binary labels indicating whether specific legal criteria were adequately addressed. These annotations serve as our ground truth matching for evaluation. These 5 student texts serve as the input texts for all experimental methods, with each text processed through our extraction and matching pipeline. The predefined solution criteria for the case were manually transformed into target triplets representing the key concepts students must address in their answer, with one target triplet defined for each of the 8 criteria. The target triplets are shown in Listing 4.3.

```
[gutgläubiger Eigentumserwerb | ist | zu prüfen]
[derivativer Eigentumserwerb | ist nicht | möglich]
[Vorliegen | von | Kaufvertrag und Übergabe]
[Fahrrad | ist | eine bewegliche Sache]
[Kaufvertrag | ist | entgeltlich]
[Fanny | ist | redlich oder gutgläubig]
[Erwerb vom Vertrauensmann | liegt vor | weil Paula Pfandgläubigerin von Erwin ist]
[Fanny | hat | Eigentum gutgläubig erworben]
```

Listing 4.3: Manually defined target triplets, one target triplet for each of the criteria shown in Figure 3.2

## 4.8 Tested LLMs

In our experiments, we employ a diverse set of state-of-the-art LLMs to assess their efficacy in triplet extraction and matching tasks. The selection is based on their performance capabilities, and availability. Below, we list the chosen models with their characteristics.

**GPT-4o** Released in March 2024 by OpenAI, GPT-4o is a large-scale multimodal language model built on a transformer architecture. It is designed for advanced natural

language understanding and generation, trained on a vast corpus of internet texts (Hurst et al., 2024).

**GPT-4o-mini-2024-07-18** Released in July 2024 by OpenAI, GPT-4o-mini-2024-07-18 is a streamlined, fine-tuned variant of GPT-4o. This transformer-based model features a reduced parameter size for enhanced efficiency while retaining strong language processing capabilities (Hurst et al., 2024).

**GPT-4.1-mini-2025-04-14** Released in April 2025 by OpenAI, GPT-4.1-mini-2025-04-14 is an advanced iteration within the GPT-4 series. It incorporates an optimized transformer architecture and improved training data for enhanced performance.

**Meta Llama-3.3-70B-Instruct-Turbo** Released in June 2024 by Meta, Meta Llama-3.3-70B-Instruct-Turbo is a 70-billion-parameter model based on an open-source transformer design. It is engineered for flexibility and customization in language processing tasks (Grattafiori et al., 2024).

**Meta Llama-4-Maverick-17B-128E-Instruct-FP8** Released in January 2025 by Meta, Meta Llama-4-Maverick-17B-128E-Instruct-FP8 is a 17-billion-parameter transformer model optimized with FP8 precision for computational efficiency. It is designed to balance high performance with reduced resource demands (Meta AI, 2025).

**DeepSeek-V3** Released in September 2024 by DeepSeek AI, DeepSeek-V3 is the third iteration of their language model series, built on a transformer architecture for superior natural language understanding and generation (Liu et al., 2024).

## 4.9 Other Methods

To provide comparative context for the primary LLM-based pipeline, several alternative methods are implemented and evaluated. These methods either replace specific components of the main pipeline or bypass certain stages entirely, allowing for a detailed analysis of the contributions of each component. Specifically:

- The **End-to-end LLM Method** directly assesses whether the input text satisfies a given criterion using the LLM, bypassing the intermediate steps of triplet extraction and matching.
- The **Rule-based Open Information Extraction Method** replaces the LLM-based triplet extraction with a deterministic, rule-based system while retaining the LLM-based matching component.

- The **String Matching-based Triplet Matching** method substitutes the LLM-based triplet matching with a string similarity approach, keeping the LLM-based triplet extraction intact.

Each of these methods is described in detail below, explaining their implementation and design principles. Figure 4.3 provides a high-level comparison, visualizing how each alternative method modifies the primary pipeline.

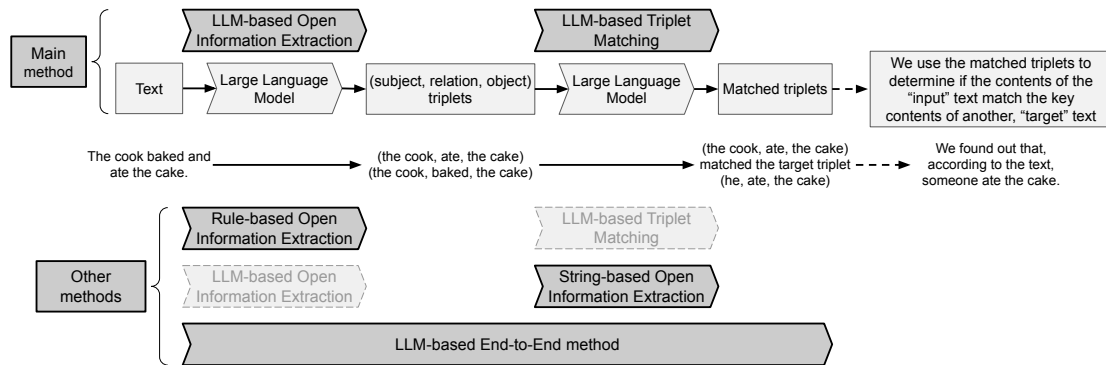


Figure 4.3: Comprehensive overview of the entire experimental setup. This diagram contrasts the primary LLM-based pipeline with the alternative methods evaluated in this study. It visualizes how the components for triplet extraction and matching are either replaced or bypassed in the End-to-end LLM, Rule-based OIE, and String Matching methods, providing a complete picture of the comparative analysis.

#### 4.9.1 End-to-end LLM Method

The End-to-end LLM Method directly evaluates whether a given text satisfies a specified criterion by leveraging the language understanding capabilities of a LLM. This approach eliminates the need for intermediate steps such as OIE and triplet matching, instead relying on the LLM to holistically assess the relationship between the text and the criterion.

##### Implementation Details:

- Prompt Design: A prompt is used to instruct the LLM, shown in Listing 4.4. The prompt includes:
  - A task description: instructing the LLM to decide if the given text fulfills the given criterion.
  - Few-shot examples: These are carefully selected pairs of text and criterion, each followed by the correct boolean output (‘True’ or ‘False’). For instance, one example presents a text about a plaintiff claiming damages under §823 BGB due to negligent driving causing an accident, with a criterion checking

if the text discusses tort-based damage claims under §823 BGB, resulting in ‘True’. Another example involves a tenant demanding rent reduction due to a leaky roof under §536 BGB, with a criterion about warranty claims in sales contracts under §433 BGB, resulting in ‘False’. These examples guide the LLM in understanding the expected reasoning and output format.

- The actual input: The text and criterion to be evaluated, formatted consistently with the examples to ensure the LLM processes them similarly.
- LLM Inference: The constructed prompt, containing the task description, examples, and the input text-criterion pair, is submitted to the LLM. The LLM processes this input and generates a response, expected to be a single word: ‘True’ if the text satisfies the criterion, or ‘False’ if it does not. This direct output leverages the LLM’s ability to interpret natural language relationships without breaking the task into smaller components.
- Response Parsing: The response parsing method interprets the LLM’s output to derive a boolean judgment through a structured three-step logic. First, it checks if the response contains the term "true" (case-insensitive) without any mention of "false," in which case it returns True. Conversely, if the response includes "false" but omits "true," it returns False. To ensure robustness, ambiguous cases trigger an error, flagging the output as invalid to handle unexpected LLM behavior reliably. This approach balances clarity with rigorous validation.

This method assesses the LLM’s capability to comprehend and evaluate text-criterion relationships directly, without relying on structured intermediate representations like triplets. It offers computational efficiency by avoiding multiple processing stages, reducing the number of LLM calls required compared to the primary pipeline. However, it may sacrifice interpretability, as it does not provide explicit intermediate outputs (such as extracted triplets) that explain the reasoning behind the judgment, making it more of a black-box approach compared to the triplet-based methods.

#### 4.9.2 Rule-based Open Information Extraction Method

To provide a deterministic benchmark for our primary LLM-based approach, we developed and implemented a custom rule-based OIE system. This was necessary due to the lack of readily available, open-source rule-based OIE tools specifically for German. The system is a simple baseline, designed to handle common grammatical structures for comparison, and is not intended to compete with state-of-the-art OIE models.

Our implementation uses the spaCy library and its German model (`de_core_news_sm`) for syntactic analysis (Honnibal et al., 2020). It works by applying a predefined set of linguistic rules to the dependency parse tree of each sentence to extract triplets. The extracted triplets are then fed into the standard LLM-based Triplet Matching component for comparison against target triplets, enabling an evaluation of the rule-based OIE’s

effectiveness relative to the LLM-based approach while maintaining consistency in the matching mechanism.

### Implementation Details:

- **Dependency Parsing:** Each sentence from the input text is processed using spaCy's dependency parser, which analyzes the sentence based on German grammar rules. This parser assigns syntactic roles to each token (word or punctuation), such as 'sb' for subject or 'oa' for accusative object, producing a dependency parse tree that represents the sentence's grammatical structure. This tree serves as the foundation for applying extraction rules.
- **Rule Set Design:** A comprehensive set of rules is defined to identify linguistic patterns corresponding to triplet structures:
  - **Main Clauses:** Rules target the root verb of a sentence, extracting associated subjects (identified by dependency labels like 'sb' or 'ep') and objects (e.g., 'oa', 'da', 'og') to form basic subject-verb-object (SVO) triplets.
  - **Coordinated Clauses:** Rules handle conjunctions (e.g., "und") to extract multiple triplets from sentences with coordinated elements, such as subjects or verbs, ensuring all parallel relationships are captured.
  - **Subordinate Clauses:** Patterns are applied to dependent clauses, like relative clauses, to extract embedded relationships, adjusting for their syntactic dependencies to the main clause.
  - **Predicative Relationships:** Rules capture copula constructions (e.g., "ist") and appositions, forming descriptive triplets such as "Der Hund ist groß" yielding [Der Hund | ist | groß].
  - **Prepositional Phrases:** For noun-linked prepositions, generic predicates like *has [preposition]* are used. For example, "Das Buch mit dem roten Cover" becomes [Das Buch | hat mit | dem roten Cover]). Verb-linked prepositions merge the verb and preposition into a relational predicate (e.g., "liegt auf").
  - **Passive Constructions:** Rules detect passive voice through auxiliary verbs (e.g., "wurde") and form triplets using the past participle, preserving the syntactic structure. Agents introduced by prepositions (e.g., "von") are linked to the patient. For example, "Das Haus wurde von dem Bauarbeiter gebaut" yields [Bauarbeiter | gebaut | Das Haus].
- **Adjustment Rules:** Additional rules refine the extracted triplets:
  - **Negations:** Negation particles (e.g., "nicht") are detected and prepended to the predicate, modifying the triplet. For example, "Der Hund beißt nicht den Mann" becomes [Der Hund | nicht beißt | den Mann]

- **Reflexive Pronouns:** Reflexive constructions are handled by ensuring pronouns refer back to the subject, adjusting triplets like "Er wäscht sich" to [Er | wäscht | Er].
- **Triplet Formation:** For each sentence, the rules are applied to every token in the dependency tree, collecting potential triplets. These are then filtered to remove duplicates (e.g., identical subject-predicate-object combinations) and malformed entries, ensuring only valid triplets are retained.
- **Integration with Matching Component:** The resulting triplets are designated as candidate triplets and passed to the LLM-based Triplet Matching component, identical to that used in the primary pipeline. This consistency allows for a direct comparison of OIE methods, isolating the effect of triplet extraction techniques on overall performance.

This simple method offers a transparent and interpretable alternative to LLM-based OIE, as the extraction logic is explicitly defined through linguistic rules. However, as a simple baseline that has not been extensively optimized, its coverage is inherently limited by its fixed set of rules. It struggles with complex sentence structures that do not match its rules, where an LLM's broader contextual understanding provides an advantage.

### 4.9.3 String Matching-based Triplet Matching

The String Matching-based Triplet Matching method replaces the LLM-based matching component of the primary pipeline with a deterministic string similarity approach. While retaining the LLM-based OIE for generating candidate triplets, this method assesses the similarity between target and candidate triplets using a surface-level comparison technique. By relying on string matching, this approach evaluates the necessity of the LLM's semantic understanding in the matching phase, providing a computationally simpler baseline for comparison.

#### Implementation Details:

- **Component-wise Similarity and Greedy Matching:** For each target triplet, its subject, predicate, and object components (converted to lowercase) are compared against those of every candidate triplet. Similarity scores, on a scale of 0 to 1, for all nine possible component-to-component pairings (e.g., target's subject vs. candidate's predicate) are calculated using the `fuzzywuzzy.fuzz.ratio` function. The method then employs a greedy algorithm to select the three best unique component pairings that maximize the sum of their individual similarity scores, ensuring each component from the target and candidate triplets is used at most once in these pairings. The similarity scores of these three selected pairings are then averaged.
- **Threshold-based Matching:** A candidate triplet is deemed a match if this final normalized score surpasses a predefined threshold of 0.5 (on the scale of 0.0 to 1.0).

The comparison is repeated for each target triplet against all candidate triplets extracted from the text. The result is a collection of matching candidate triplets for each target, which replaces the LLM-based matching output in subsequent evaluation steps. This process systematically builds a set of matches without requiring semantic interpretation beyond string similarity.

This method is computationally efficient and straightforward to implement, as it relies on direct string comparison rather than the resource-intensive inferences of an LLM. However, it overlooks semantic similarities, such as synonyms (e.g., "beißt" vs. "knabbert") or paraphrased expressions, that an LLM could recognize through contextual understanding. This approach serves as a baseline to quantify the added value of the LLM's semantic capabilities in the triplet matching stage.

```
Decide if the given text fulfills the given criterion.

EXAMPLES:
Text:
Der Kläger macht Schadenersatz nach § 823 BGB geltend, da der Beklagte durch
fahrlässiges Fahren einen Unfall verursacht habe. Die Kausalität zwischen dem
Fehlverhalten und dem Gesundheitsschaden ist gegeben. Eine Rechtfertigung liegt
nicht vor.
Criterion:
Der Text thematisiert Schadenersatzansprüche aus unerlaubter Handlung gemäß §
823 BGB.
Output:
True

Text:
Der Mieter verlangt Mietminderung wegen eines undichten Daches gemäß § 536 BGB.
Der Vermieter habe die Pflicht zur Instandhaltung verletzt. Ein rechtzeitiges
Handeln des Vermieters blieb aus.
Criterion:
Der Text behandelt Gewährleistungsansprüche beim Kaufvertrag nach § 433 BGB.
Output:
False

Now, please process the following text and criterion:
Text:
{{text}}
Criterion:
{{criterion}}
Output:
```

Listing 4.4: Few-shot LLM prompt used for end-to-end text-criterion assesment

## 4.10 Implementation Details

The framework is realized as a structured Python package. The codebase is organized into modules responsible for distinct functionalities like configuration management, data structure definitions, LLM interfacing, OIE logic, triplet matching logic, evaluation procedures, implementation of other methods, and dataset-specific loading utilities. A central script orchestrates the execution of experiments and analysis.

Key software abstractions include classes representing the LLM interface, the OIE extractor, the triplet matcher, and the core data entities (e.g. `Triplet`). Evaluation logic is encapsulated in dedicated functions and classes for managing result dataframes and calculating metrics. The alternative methods are implemented as separate classes.

The modular structure promotes extensibility, making it simple to integrate new LLMs, OIE techniques, or matching strategies. System behavior is highly customizable through the configuration file, allowing adjustments to model choice, generation parameters, and prompt content. Dependencies on external libraries are managed through standard Python package management.

The developed framework, detailed in this section, is implemented in Python and its source code is publicly accessible at <https://github.com/TamasCsakvari/oie-llm-framework>.

## 4.11 Experimental Setup

Experiments are configured primarily through a dedicated configuration module and executed by the main orchestration script. Key configurable parameters include the choice of LLM, generation settings like maximum tokens and temperature, the specific prompt templates used for OIE and matching, API credentials, paths for data input and result output, and settings for local model execution if used. These configurations allow for systematic variation of parameters to explore their impact on performance during experimental runs.

## 4.12 Evaluation Methodology

The framework's effectiveness, particularly its ability to correctly match text content against predefined criteria via triplet matching, is evaluated quantitatively using a German legal case dataset. The system's automated matching decisions are compared against a human-annotated reference dataset.

### 4.12.1 Performance Metrics

The comparison between the framework's automated matching decisions (represented as a boolean matrix) and the human-annotated ground truth is quantified using standard binary classification metrics. First, the counts for the four confusion matrix categories

are determined by comparing the system's prediction matrix ( $Y_{pred}$ ) with the reference matrix ( $Y_{true}$ ) element-wise for all  $N$  comparisons (texts  $\times$  criteria):

The following metrics are calculated from four key counts: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

- **Accuracy**
- **Precision**
- **Recall**
- **Specificity**
- **F1-Score:** The harmonic mean of Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

- **Balanced Accuracy:** The average of Recall and Specificity.

$$\text{Balanced Accuracy} = \frac{\text{Recall} + \text{Specificity}}{2}$$

- **Matthews Correlation Coefficient (MCC):** A correlation coefficient between the observed and predicted binary classifications; ranges from -1 to +1, where +1 indicates perfect prediction, 0 random prediction, and -1 total disagreement.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

These metrics provide a comprehensive view of the system's performance across different dimensions.

#### 4.12.2 Method Evaluation

The evaluation includes several comparative analyses. The primary LLM pipeline is executed using 5 different LLMs, allowing for direct comparison of different models' effectiveness based on the calculated metrics. Furthermore, the performance of the main pipeline is compared against the implemented "other methods": the End-to-end LLM approach, the Rule-based OIE combined with LLM Matching, and the LLM OIE combined with String Matching. All results are aggregated into a summary table for clear comparison.

In addition to quantitative metrics, qualitative analysis is facilitated by logging detailed intermediate outputs. The extracted triplets for each sentence and the specific candidate triplets identified as matches for each criterion are saved to text files. Manual inspection

of these outputs allows for identifying patterns in extraction quality, understanding specific matching errors, and assessing the system's robustness in handling diverse linguistic expressions of the target criteria.

The methods subjected to quantitative evaluation are the following. For the "other method" configurations that require an LLM, we used GPT-4.1-mini, as it was the top-performing model in the main pipeline evaluation.

- Main method: LLM-based OIE combined with LLM-based matching instantiated with the following models, described in Section 4.8:
  - GPT-4o
  - GPT-4o-mini
  - GPT-4.1-mini
  - Llama-3.3-70B-Instruct-Turbo
  - Llama-4-Maverick-17B-128E-Instruct-FP8
  - DeepSeek-V3
- Other method: End-to-end LLM Matching using GPT-4.1-mini
- Other method: LLM-based (GPT-4.1-mini) OIE combined with String Matching.
- Other method: Rule-based OIE combined with LLM-based (GPT-4.1-mini) Matching.

Listed in the same order, for brevity, the methods will be referred to as:

1. GPT-4o (main method)
2. GPT-4o-mini (main method)
3. GPT-4.1-mini (main method)
4. Llama-3.3 (main method)
5. Llama-4 (main method)
6. DeepSeek-V3 (main method)
7. End-to-end LLM (other method)
8. String matching (other method)
9. Rule-based OIE (other method)



# Results

## 5.1 Qualitative Analysis of Extracted Triplets

This section evaluates the quality of triplets extracted by four methods: three LLM-based approaches (GPT-4o, GPT-4.1-mini, and DeepSeek-V3) and the rule-based system. The analysis focuses on syntactic accuracy, semantic coherence, and contextual relevance, drawing upon representative sentence examples and a categorization of common extraction errors.

### 5.1.1 Comparative Analysis of Triplet Extraction Methods on Representative Sentences

The quality of triplet extraction is not an isolated concern; it directly dictates the success of subsequent triplet matching. Since matching inherits all strengths and weaknesses from the initial extraction, any errors like omissions, fragmentation, or misinterpretation inevitably propagate, potentially undermining the entire semantic comparison. The following representative sentences were selected to illustrate how variations in extraction characteristics (such as granularity, contextual awareness, and syntactic robustness) impact the extracted triplets, and thus the input provided for matching. This qualitative comparison is therefore essential for understanding the practical implications of different extraction approaches on the reliability of semantic comparison, particularly in a high-stakes domain like legal text, and for interpreting the quantitative results of the matching process (as discussed in Section 5.2).

The subsequent examples showcase the OIE performance on three distinct sentences from the German legal texts. For each original sentence and its English translation, the triplets generated by GPT-4.1-mini, GPT-4o, DeepSeek-V3, and the rule-based system are provided, along with a concise characterization of each method's output. These

examples were chosen to highlight instances where the methods yielded notably different extraction results.

### Text 1, Sentence 1

**Original Sentence:** "SV kein Schild an dem Fahrrad angebracht war, welches auf die Pfandeigenschaft des Fahrrads hinwies."

**Translation:** *According to the case description (SV), no sign was attached to the bicycle indicating its pledge status.*

- **GPT-4.1-mini:**  
[an dem Fahrrad | war angebracht | kein Schild],  
[welches | wies hin | auf die Pfandeigenschaft des Fahrrads],  
[Schild | wies hin | auf die Pfandeigenschaft des Fahrrads]  
This model tends to extract fragmented relationships, emphasizing granularity over a more holistic capture of context.
- **GPT-4o:**  
[kein Schild an dem Fahrrad | angebracht war | welches auf die Pfandeigenschaft des Fahrrads hinwies]  
In contrast, this model merges the main and relative clauses into a single triplet, thereby preserving contextual coherence.
- **DeepSeek-V3:**  
[SV | kein Schild angebracht | an dem Fahrrad],  
[Schild | hinwies | auf die Pfandeigenschaft des Fahrrads]  
This approach explicitly splits actors (such as "SV") and causal links into separate, distinct triplets.
- **Rule-Based:**  
*No triplets extracted.*  
The rule-based method fails entirely to parse this sentence, likely due to the nested clause structure and indirect relationships.

### Text 1, Sentence 2

**Original Sentence:** "F hatte keinen Grund zu vermuten, dass P nicht die Eigentümerin des Fahrrads, das sich in ihrer Sphäre befindet, sei; zudem gab P auch an, Eigentümerin zu sein."

**Translation:** *F had no reason to suspect that P was not the owner of the bicycle, which was in her possession; moreover, P claimed to be the owner.*

- **GPT-4.1-mini:**  
[F | hatte keinen Grund zu vermuten | dass P nicht die Eigentümerin des

Fahrrads sei],  
 [das Fahrrad | befindet sich | in ihrer Sphäre],  
 [P | gab an | Eigentümerin zu sein]  
 Here, GPT-4.1-mini effectively balances the extraction of intent, spatial context, and direct statements into distinct, meaningful triplets.

- **GPT-4o:**

[F | hatte | keinen Grund zu vermuten, dass P nicht die Eigentümerin des Fahrrads, das sich in ihrer Sphäre befindet, sei],  
 [P | gab an | Eigentümerin zu sein]  
 GPT-4o demonstrates its tendency to condense nested clauses, as seen in the first triplet, while still retaining the core semantic meaning of the statement.

- **DeepSeek-V3:**

[F | hatte keinen Grund zu vermuten | dass P nicht die Eigentümerin des Fahrrads, das sich in ihrer Sphäre befindet, sei],  
 [P | gab an | Eigentümerin zu sein],  
 [Fahrrad | befindet sich in | ihrer Sphäre]  
 Similar to GPT-4.1-mini for this sentence, DeepSeek-V3 also isolates the spatial context concerning the bicycle into a standalone triplet.

- **Rule-Based:**

[F | hatte | ],  
 [das | befindet | das],  
 [die Eigentümerin des Fahrrads | befindet | die Eigentümerin des Fahrrads],  
 [P | sei | ],  
 [die Eigentümerin des Fahrrads | sei | ],  
 [P | gab an | ]  
 The rule-based system produces several nonsensical or severely incomplete triplets, indicative of its difficulties with the dependency parsing required for this complex sentence structure.

### Text 4, Sentence 3

**Original Sentence:** "Kennt Sie zB. ihr eigenes Fahrrad und könnte Sie Zweifel haben, zB weil Erwins Rad ein Fahrrad ein Herrenfahrrad ist, dann wäre die Gutgläubigkeit zu verneinen."

**Translation:** *If she knows, e.g., her own bicycle and could have doubts, e.g., because Erwins bicycle is a mens bicycle, then good faith would be denied.*

- **GPT-4.1-mini:**

[Sie | kennt | ihr eigenes Fahrrad],  
 [Sie | könnte haben | Zweifel],  
 [Erwins Rad | ist | ein Herrenfahrrad],

[die Gutgläubigkeit | wäre zu verneinen | wenn Sie Zweifel haben]  
 GPT-4.1-mini successfully captures the conditional logic present in the sentence and explicitly links it to the legal consequence in the final triplet.

- **GPT-4o:**

[Sie | kennt | ihr eigenes Fahrrad],  
 [Sie | könnte haben | Zweifel],  
 [Erwins Rad | ist | ein Herrenfahrrad],  
 [die Gutgläubigkeit | wäre | zu verneinen]

This model simplifies the conditional phrasing, particularly in the object of the consequence triplet, but still manages to preserve the core causal link.

- **DeepSeek-V3:**

[Sie | kennt | ihr eigenes Fahrrad],  
 [Sie | könnte haben | Zweifel],  
 [Erwins Rad | ist | ein Herrenfahrrad],  
 [die Gutgläubigkeit | wäre zu verneinen | (implied context: if conditions above are met)]

DeepSeek-V3 attempts to enhance clarity by annotating the implied conditional context within the object of the triplet expressing the consequence.

- **Rule-Based:**

[Sie | Kennt | ihr eigenes Fahrrad]

The rule-based method largely ignores the complex conditional clauses and the resulting causal link, managing to extract only a single, isolated fact from the sentence.

### 5.1.2 Common Errors in Triplet Extraction

A review of the extracted triplets highlighted several recurring types of errors that diminished their quality by causing information loss or semantic distortion. Instead of using highly technical labels, these errors can be grouped into four broader, more intuitive categories. Understanding these patterns is key to grasping the practical limitations of each extraction method.

**1. Loss of Information and Context** This was the most frequent issue, where extracted triplets were technically correct but semantically incomplete. This occurred in several ways: sometimes models would omit crucial qualifying details, such as the reason for a legal conclusion. In other cases, they would fragment a single, coherent idea across multiple triplets or, conversely, collapse distinct concepts (like items in a list) into a single, overloaded triplet.

- **Example (Omission):** From "Gutgläubigkeit liegt vor, weil lt. SV kein Schild... war", a model extracted: [Gutgläubigkeit | liegt vor | ], omitting the entire causal clause ("weil...").

- **Example (Omission):** From "Der derivative Erwerb von Eigentum scheidet aus...", a model extracted: [Der derivative Erwerb | scheidet aus | ], omitting the legal rationale that followed.
- **Example (Collapsing a List):** From "Die Alternativvoraussetzungen sind der Erwerb in einer öffentlichen Versteigerung, der Erwerb im Unternehmen oder...", a model extracted a single triplet, collapsing three distinct conditions into one object: [Die Alternativvoraussetzungen | sind | der Erwerb in..., der Erwerb im... oder...]

**2. Difficulty Identifying the Correct Subject or Reference** Models sometimes struggled to correctly determine who or what was performing an action or being described. This was particularly evident with pronouns (e.g., "er," "sie," "es" - he/she/it), leading to triplets with ambiguous or misattributed subjects. This error also occurred when a source qualifier was mistaken for an actor.

- **Example (Ambiguous Pronoun):** From "...weder P, noch E können es herausverlangen, er hat das Eigentumsrecht verloren.", a model extracted: [er | hat verloren | das Eigentumsrecht], where "er" is ambiguous.
- **Example (Unresolved Pronoun):** From "Kennt Sie zB. ihr eigenes Fahrrad...", a model extracted: [Sie | Kennt | ihr eigenes Fahrrad], leaving the pronoun "Sie" unresolved.
- **Example (Misidentified Actor):** From "SV kein Schild an dem Fahrrad angebracht war...", a model incorrectly assigned the source ("SV") as the actor: [SV | kein Schild angebracht | an dem Fahrrad].

**3. Inconsistent Handling of Legal and Source Qualifiers** References to legal statutes (e.g., §367 ABGB) or sources of information (e.g., "laut SV" - according to the case description) were handled inconsistently. At times, these crucial qualifiers were omitted entirely. In other cases, they were awkwardly merged into a triplet or misidentified as a core component, like the subject, distorting the statement's meaning.

- **Example (Omitted Statute):** From "§367 ABGB schützt den gutgläubigen Erwerber.", a model extracted: [ABGB | schützt | den gutgläubigen Erwerber], omitting the critical "§367" reference.
- **Example (Awkward Integration):** From "Titel ist der Kaufvertrag und wurde das Fahrrad laut SV auch übergehen", a model extracted: [der Kaufvertrag | wurde | das Fahrrad laut SV auch übergehen], misattributing the subject and awkwardly placing "laut SV" in the object.

- **Example (Separate Integration):** From "Fanny erwirbt... Eigentum am Fahrrad. (§367 ABGB)", one model created an extra, explanatory triplet to connect the statute: [§367 ABGB | regelt | Fanny erwirbt originär Eigentum am Fahrrad], a valid but distinct approach to integration.

**4. Mishandling of Negation** Errors arose when models struggled to accurately represent negative statements. This included failing to associate the negation with the correct part of the sentence or extracting negative statements as separate facts without capturing the causal link between them, which is often crucial in legal reasoning.

- **Example (Incorrect Isolation):** From "Da Paula nicht Eigentümerin... ist kann Sie diese nicht verkaufen.", a rule-based system extracted: [Sie | not kann verkaufen | diese], incorrectly isolating the negation and misidentifying the subject.
- **Example (Separated Facts):** From the same sentence, a model extracted two correct but separate facts: [Paula | ist nicht | Eigentümerin des Fahrrades] and [Paula | kann nicht verkaufen | diese], potentially losing the explicit causal link ("Da..." - Since...) between them in a single structure.

### 5.1.3 Open Information Extraction Method Characterization

The qualitative analysis, framed by these common error types, reveals distinct behavioral patterns for each method. These characteristics directly influence the quality of input for the subsequent triplet matching phase.

**GPT-4o** GPT-4o typically prioritized contextual coherence, often merging related information into holistic triplets. While this captures broad semantics well, this tendency to condense sometimes led to a Loss of Information and Context, such as omitting the statutory basis for a legal rule. It also showed some weakness in handling lists or alternatives, occasionally producing illogical triplets. While its handling of negation was generally competent, it could separate causally linked negative facts, requiring the matching process to synthesize information from multiple candidates.

**GPT-4.1-mini** GPT-4.1-mini showed a preference for granular extraction, dissecting sentences into multiple, detailed propositions. This detail is useful but sometimes led to a Loss of Information and Context by creating fragmented triplets where, for example, a causal clause was omitted. When handling lists, it tended to collapse multiple distinct conditions into a single triplet with a very long object. It also occasionally struggled with the Inconsistent Handling of Legal and Source Qualifiers, misattributing roles or creating awkward constructions.

**DeepSeek-V3** DeepSeek-V3 used a highly explicit and often verbose style. While detailed, it was prone to producing triplets that suffered from a Loss of Information and Context, similar to GPT-4.1-mini. A notable weakness was its Difficulty Identifying the Correct Subject or Reference, for instance by misinterpreting pronouns or treating a source qualifier like "SV" as an actor. Its approach to lists was to expand them into many separate triplets, which, though verbose, could be semantically accurate for individual items.

**Rule-Based Method** The rule-based method serves as an important baseline, and its performance illustrates a clear trade-off between deterministic precision and flexible coverage. As shown in the qualitative examples, its predefined rules meant it struggled with the complex and nested grammar common in legal text, leading to incomplete or nonsensical outputs for those specific sentences. Its limitations were most apparent in the Loss of Information and Context and Difficulty Identifying the Correct Subject or Reference when sentence structures deviated from simple patterns. While its rigid nature makes it less suitable for comprehensively parsing the full range of legal language compared to LLMs, its effectiveness on simpler structures provides a valuable benchmark for the task.

#### 5.1.4 Performance Summary

Qualitative analysis highlights that LLMs show considerable potential for OIE in the complex domain of German legal text. Compared to the rule-based system evaluated, LLMs generally demonstrated a greater capacity to handle semantic nuances, contextual dependencies, and complex sentence structures. They often succeeded in extracting meaningful relationships from sentences involving negation, conditional logic, and indirect statements, an area where the rule-based system showed its limitations on complex sentence structures. This allowed the LLMs to provide a richer and more comprehensive input for downstream tasks like triplet matching.

Despite these observed advantages, the performance of LLMs is not flawless and varies considerably across different models. Common challenges persisted, as detailed in the analysis of common errors. These include the Loss of Information and Context by oversimplifying or fragmenting triplets, Difficulty Identifying the Correct Subject or Reference in complex sentences, the Mishandling of Negation, and the Inconsistent Handling of Legal and Source Qualifiers. Each of these failings directly impacts the potential success of subsequent semantic matching by either omitting crucial information or presenting distorted representations.

Individual LLMs demonstrated distinct OIE approaches, leading to different strengths and weaknesses as input for matching:

- GPT-4o leaned towards holistic, contextually coherent triplets. This can benefit matching of broad criteria but may obscure necessary details or create challenges if the matching target requires finer granularity.

- GPT-4.1-mini opted for detailed, explicit extractions. This provides granular candidates beneficial for specific criteria matching, but its tendency towards fragmentation or collapsing coordinated structures into long components can necessitate more complex parsing or synthesis by the triplet matcher.
- DeepSeek-V3, while attempting explicit and detailed representation, exhibited a distinct verbose style. Its expansion of lists can provide accurate individual candidates for matching, but its propensity for certain structural and semantic inaccuracies (like misidentifying subjects) can introduce flawed inputs to the matching stage.

In essence, this qualitative assessment demonstrates that while LLM-based OIE shows significant promise for extracting structured information from these legal texts, the quality and nature of the extracted triplets are highly model-dependent. The specific error patterns and behavioral characteristics observed directly inform their suitability as input for subsequent processes. For this use case, the analysis suggests GPT-4.1-mini's OIE outputs, despite some fragmentation, offered a comparatively strong balance of detail and semantic relevance for the subsequent matching task among the methods analyzed. The fidelity of these initial OIE outputs is fundamental, as the success of any downstream semantic comparison, like triplet matching, is inherently constrained by the accuracy and completeness of this foundational extraction stage.

## 5.2 Quantitative Analysis of Triplet Matching Performance

The evaluation of different triplet matching methods shows clear differences in how well they classify relationships. Table 5.1 compares the nine methods listed in Section 4.12.2, using the performance metrics described in 4.12.1. The LLM-based matching methods (first six rows) were evaluated using candidate triplets generated by their respective OIE counterparts (e.g., GPT-4o matching used triplets from GPT-4o OIE). End-to-end LLM was done with GPT-4.1-mini, the OIE step for String matching was done with GPT-4.1-mini, and the triplet matching step for Rule-based OIE was done with GPT-4.1-mini.

The results in Table 5.1 show that the top-performing methods are consistently LLM-based systems, which achieve the highest scores in balanced metrics like F1-score and MCC, though simpler heuristics can remain competitive with lower-performing LLMs. Among the LLM-based matching methods (first six rows), GPT-4.1-mini demonstrates the most robust overall performance. It achieves the highest Accuracy (0.8000), Balanced Accuracy (0.7604), F1-Score (0.8519), and Matthews Correlation Coefficient (MCC) (0.5893). This strong performance across multiple metrics, particularly B.Acc and MCC, suggests its effectiveness in providing reliable classifications for both positives and negatives. Its leading F1-score is a result of high Precision (0.7667) and an exceptional Recall (0.9583), having missed only one true positive match.

Method	TP	TN	FP	FN	Acc.	Prec.	Rec.	Spec.	B.Acc.	F1	MCC
GPT-4o	13	<u>13</u>	3	11	0.6500	<u>0.8125</u>	0.5417	<u>0.8125</u>	0.6771	0.6500	0.3542
GPT-4o-mini	19	9	7	5	0.7000	0.7308	0.7917	0.5625	0.6771	0.7600	0.3638
GPT-4.1-mini	23	9	7	1	<u>0.8000</u>	0.7667	0.9583	0.5625	<u>0.7604</u>	<u>0.8519</u>	<u>0.5893</u>
Llama-3.3	23	8	8	1	0.7750	0.7419	0.9583	0.5000	0.7292	0.8364	0.5377
Llama-4	<u>24</u>	4	12	0	0.7000	0.6667	<u>1.0000</u>	0.2500	0.6250	0.8000	0.4082
DeepSeek-V3	19	9	7	5	0.7000	0.7308	0.7917	0.5625	0.6771	0.7600	0.3638
End-to-end LLM	21	8	8	3	0.7250	0.7241	0.8750	0.5000	0.6875	0.7925	0.4114
String matching	16	11	5	8	0.6750	0.7619	0.6667	0.6875	0.6771	0.7111	0.3474
Rule-based OIE	15	12	4	9	0.6750	0.7895	0.6250	0.7500	0.6875	0.6977	0.3679

Highest value in each column underlined; TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives, Acc.: Accuracy, Prec.: Precision, Rec.: Recall, Spec.: Specificity, B.Acc.: Balanced Accuracy, F1: F1-Score, MCC: Matthews Correlation Coefficient

Table 5.1: Performance metrics for the nine evaluated methods

Other LLMs exhibit distinct strengths and weaknesses. Llama-4 achieves perfect Recall (1.0000) by identifying all 24 true positive matches (0 False Negatives). However, this comes at the cost of lower Precision (0.6667) and the lowest Specificity (0.2500) among all methods, due to a high number of False Positives (12 FPs), indicating a strong positive bias. Conversely, GPT-4o records the highest Precision (0.8125) and Specificity (0.8125), making it highly reliable when it predicts a match and effective at identifying true negatives. Its performance is tempered by a lower Recall (0.5417), as it failed to identify 11 true matches, leading to a modest F1-score of 0.6500.

Llama-3.3 also delivers strong results with a high Recall (0.9583), comparable to GPT-4.1-mini, and a robust F1-score (0.8364). Its performance is slightly behind GPT-4.1-mini due to a marginally lower Precision. The GPT-4o-mini and DeepSeek-V3 models present identical performance profiles, achieving an F1-score of 0.7600 and a Balanced Accuracy of 0.6771, showing medium performance among the tested LLMs for this task.

The performance of the alternative methods, as defined in Section 4.9, provides further insights:

- The **End-to-end LLM** method, which bypasses intermediate triplet extraction and matching by directly assessing text with an LLM, specifically GPT-4.1-mini, achieves an F1-score of 0.7925 and a Balanced Accuracy of 0.6875. While not surpassing the modular approach using GPT-4.1-mini, its competitive performance indicates that direct LLM assessment can be a viable, simpler alternative, though potentially sacrificing the interpretability and fine-grained control of a structured pipeline. At the same time, this result demonstrates that our modular framework is able to provide not only an intermediate structured representation as an advantage but also improved performance.

- The **String matching** method (using LLM-based triplet extraction but replacing LLM-based matching with string similarity) yields an F1-score of 0.7111 and an MCC of 0.3474. While its Precision (0.7619) is respectable, the Recall (0.6667) is lower than most dedicated LLM triplet matchers. This suggests that while string similarity can identify exact or near-exact matches, it struggles with the semantic nuances and paraphrasing that LLM-based triplet matchers are designed to handle, thus underscoring the value of LLMs for the matching component (relevant to RQ3).
- The **Rule-based OIE** method, when combined with an LLM matcher, yielded a surprisingly strong F1-score of 0.6977, notably outperforming GPT-4o. This performance profile, high Precision (0.7895) alongside low Recall (0.6250), is explained by the qualitative analysis (Section 5.1). The rule-based system excelled at accurately extracting triplets from simple grammatical structures but struggled to parse more complex sentences. This establishes the method as a highly precise baseline but shows its limited recall in achieving comprehensive coverage, a task where LLMs demonstrate a clear advantage.

In summary, the quantitative results show clear differences in model performance, with GPT-4.1-mini leading in key balanced metrics. When interpreting these results for practical application, however, it is crucial to consider factors like efficiency and cost. For instance, the rule-based system offers the highest efficiency, while large models like GPT-4o are the most resource-intensive. From this vantage point, the success of a highly efficient model like GPT-4.1-mini becomes even more significant. It not only achieves the highest accuracy in this study but does so without the computational demands of its larger counterparts, making it the most well-rounded and practical solution evaluated.

### 5.3 Qualitative Analysis of Triplet Matching Performance

This section qualitatively examines the performance of different triplet matching approaches: three LLM-based methods (using GPT-4o, GPT-4.1-mini, and DeepSeek-V3 as the matching LLM, as well as the models providing the candidate triplets from the OIE step) and a String-Matching-based method (using candidate triplets extracted by GPT-4.1-mini). The analysis focuses on the ability to identify semantic equivalence, handle paraphrasing, and avoid common pitfalls, drawing upon representative matching examples and a categorization of observed error patterns. The goal is to understand the nuances behind the quantitative metrics presented in Section 5.2.

#### 5.3.1 Comparative Analysis of Triplet Matching Methods on Representative Sentences

The effectiveness of any text comparison system heavily relies on the accuracy of its underlying triplet matching component. Errors in determining semantic equivalence be-

tween a target (criterion) triplet and candidate (extracted) triplets can lead to incorrect assessments of whether a text fulfills specific criteria.

To illustrate this, the following sections show representative examples. Each example first presents the **Target Triplet** and the **Relevant Student Text Snippet**. After this, we compare how different matching methods performed, noting whether errors occurred during the OIE stage or the subsequent matching phase. This regular structure helps to clearly compare how each method deals with various semantic challenges.

#### Straightforward Match (Text 1, Criterion 4)

**Target Triplet:** [Fahrrad | ist | eine bewegliche Sache] **Relevant Student Text Snippet:** "...beim Fahrrad handelt es sich um eine bewegliche Sache."

- **GPT-4.1-mini, GPT-4o, and DeepSeek-V3** All three LLM-based OIE methods successfully extracted semantically equivalent candidate triplets, such as [beim Fahrrad | handelt es sich um | eine bewegliche Sache]. Their respective matching components correctly identified these as a MATCH. This aligned with the expert annotation (True Positive) and demonstrates that all three LLMs can handle minor lexical and structural variations.
- **String Matching**  
Using triplets from GPT-4.1-mini, the candidate [beim Fahrrad | handelt es sich um | eine bewegliche Sache] achieved a similarity score of 0.613. This was above the threshold, resulting in a correct MATCH (True Positive). The high score was driven by strong lexical overlap in the subject and object components.

#### Challenging Semantic Equivalence (Text 1, Criterion 3)

**Target Triplet:** [Vorliegen | von | Kaufvertrag und Übergabe] **Relevant Student Text Snippet:** "...Voraussetzungen hierfür ein gültiger Titel und ein Modus sind..." and "...Trotz Modus (Übergabe) an Fanny scheitert der Erwerb."

- **GPT-4.1-mini** The OIE component extracted relevant context, including [Trotz Modus (Übergabe) an Fanny | scheitert | der Erwerb]. The matching component then correctly identified these candidates as fulfilling the criterion, resulting in a successful MATCH (True Positive). This ability to handle a complex, paraphrased criterion where information is spread across the text distinguishes its performance.
- **GPT-4o and DeepSeek-V3**  
Both methods failed at the OIE stage for this criterion. Their OIE components read the text but did not extract any candidate triplets related to "Titel," "Modus," or "Übergabe." Since no relevant information was extracted, the matching component was never presented with a candidate to evaluate, leading to an inevitable NO MATCH decision and a False Negative.

- **String Matching**

This method also produced a False Negative. The candidate triplet with the highest lexical similarity, [die Voraussetzungen hierfür | sind | ein gültiger Titel und ein Modus] (extracted by GPT-4.1-mini), achieved a score of only 0.333. This low score, due to the lack of direct lexical similarity with the target, demonstrates the limitations of string-based methods on paraphrased content.

### Potential False Positive due to Keyword Overlap (Text 2, Criterion 6)

**Target Triplet:** [Fanny | ist | redlich oder gutgläubig] **Relevant Student Text Snippet:** "Wenn Fanny gutgläubig ist, d.h., sie wusste nicht und konnte nicht wissen, dass Paula nicht die Eigentümerin war..."

- **Analysis of the Error** This example illustrates a common challenge where methods produce a False Positive. The systems identified candidate triplets like [Fanny | ist | gutgläubig] within the text based on a strong keyword match. However, the expert annotation for this criterion was 'False', indicating that this simple statement was insufficient to meet the full legal requirement without further context or justification. This over-reliance on a partial lexical match is the source of the error.
- **GPT-4.1-mini, GPT-4o, DeepSeek-V3, and String Matching** All four methods identified a match based on the strong lexical overlap with "gutgläubig," resulting in a MATCH decision. In each case, this contradicted the expert annotation, producing a False Positive for the reason described above.

#### 5.3.2 Common Error Classes in Triplet Matching

The qualitative review of matching decisions revealed several recurring patterns that led to errors. These challenges can be grouped into three main categories, helping to explain the performance variations observed in the quantitative results.

**Matching on Keywords Without Full Context** This was a frequent cause of False Positives. In these cases, the matching component correctly identified a strong lexical overlap between a candidate and the target triplet but failed to account for the broader context that would invalidate the match. A clear example is Text 2, Criterion 6, where the target was [Fanny | ist | redlich oder gutgläubig]. All LLM-based methods found a candidate like [Fanny | ist | gutgläubig] and flagged it as a match. However, the ground truth required additional justification which was absent from the student's text, making the simple assertion insufficient. This shows a tendency for models to be swayed by salient keywords, even when the complete semantic meaning is not fulfilled.

**Mismatch in Relational Nuance** Another common source of False Positives was the misinterpretation of nuanced relationships, especially those involving causality. For Text

1, Criterion 7, the target was [Erwerb vom Vertrauensmann | liegt vor | weil Paula Pfandgläubigerin von Erwin ist]. Both GPT-4.1-mini and DeepSeek-V3 incorrectly flagged a match based on simpler candidate triplets like [P | gilt als | Vertrauensmann des E]. While the candidate correctly identified "Vertrauensmann," it completely missed the crucial causal condition ("weil..."). The matching models treated the partial overlap as sufficient, demonstrating a difficulty in strictly enforcing the full relational structure of the target.

**Upstream Failures in Triplet Extraction (OIE)** Perhaps the most critical finding is that many apparent matching failures are actually caused by errors in the preceding OIE stage. If the OIE component does not extract the necessary information from the source text, no matching algorithm can succeed. This was precisely the cause of the False Negatives for GPT-4o and DeepSeek-V3 on Text 1, Criterion 3. For this complex criterion ([Vorliegen | von | Kaufvertrag und Übergabe]), their OIE modules extracted no relevant candidate triplets. The failure was not in matching, but in the lack of input to the matcher. This highlights the critical dependency of the entire pipeline on a robust and comprehensive OIE phase, as errors in this initial step are irrecoverable.

### 5.3.3 Triplet Matching Method Characterization

The qualitative analysis of triplet matching decisions, supported by quantitative metrics, reveals distinct behavioral characteristics for each pipeline, often highlighting the critical role of the initial OIE stage.

**GPT-4o** The GPT-4o pipeline's performance was defined by a sharp contrast between its components. Its leading Precision (0.8125) and Specificity (0.8125) suggest a reliable and cautious matching component when provided with relevant information. However, the system's overall effectiveness was significantly hampered by its OIE component. The low Recall (0.5417) and resulting modest F1-score (0.6500) were largely due to failures in the extraction stage. As seen in the "Challenging Semantic Equivalence" example, the extractor failed to capture key information from complex sentences, leading to unavoidable False Negatives before the matching stage was even attempted.

**GPT-4.1-mini** The GPT-4.1-mini pipeline emerged as the most effective system in this analysis. Qualitatively, it demonstrated a superior ability to extract relevant information even from complex and paraphrased text, and its matching component was able to correctly identify semantic equivalence from that information. This robust end-to-end performance, particularly its success on challenging cases where other models failed at the OIE stage, translated into the highest scores across balanced metrics like F1-Score (0.8519) and MCC (0.5893). Its high Recall (0.9583) shows its comprehensive coverage, though this was paired with a slightly higher tendency for False Positives on nuanced criteria compared to the most precise models.

**DeepSeek-V3** The DeepSeek-V3 pipeline exhibited a mixed performance stemming from weaknesses at both the OIE and matching stages. Similar to GPT-4o, its OIE component struggled with complex sentences, failing to extract relevant information in the challenging equivalence case, which contributed to its False Negatives. However, when it did extract triplets, its matching component was sometimes overly permissive, leading to False Positives by misinterpreting relational nuance or relying too heavily on keyword overlap (e.g., Text 1, Criterion 7). This combination of extraction gaps and matching inaccuracies explains its moderate overall performance.

**String-Matching-Based Method** The String Matching method behaved predictably as a primarily lexical similarity approach. Qualitatively, it was effective only when there was substantial surface-level overlap between target and candidate triplet components. This was reflected in a respectable Precision (0.7619) when such overlap existed. However, its significant struggle with semantic variations and paraphrasing its inability to handle conceptual equivalence when lexical forms differed substantially was its primary qualitative limitation, leading to a lower Recall (0.6667). It was also easily misled into False Positives by strong keyword alignment without true semantic equivalence. This method primarily serves as a baseline, with its performance underscoring the substantial advantage LLMs offer in interpreting semantic meaning beyond superficial string similarity.

### 5.3.4 Performance Summary

This qualitative analysis, contextualized by the quantitative results, reveals that LLM-based pipelines generally offer a more nuanced and effective method for determining semantic equivalence compared to simpler approaches like string matching. Their ability to handle significant lexical and structural variations is a key advantage, leading to stronger overall performance in balanced metrics like the F1-score and MCC.

However, significant performance differences exist among the LLMs, with distinct error profiles emerging. A common challenge leading to False Positives was an over-reliance on keywords; several models identified matches based on salient terms while missing the full contextual or justificatory scope of a criterion. This suggests that even advanced models can be swayed by superficial lexical signals over deeper semantic consistency.

Conversely, False Negatives were also a major issue, primarily stemming from failures at the OIE (extraction) stage. Models like GPT-4o, despite its high-precision matcher, and DeepSeek-V3 often failed to extract any relevant information from complex, paraphrased sentences. This rendered a subsequent match impossible and was the main driver of their lower recall scores. In contrast, the GPT-4.1-mini pipeline proved more robust, successfully handling these difficult extraction and matching tasks, which explains its top-tier F1-score and exceptional recall. This starkly illustrates that the performance of the matching component is fundamentally dependent on the quality of the upstream OIE component.

In essence, this analysis indicates that while LLMs significantly advance beyond purely lexical methods, their effectiveness is not uniform across the entire pipeline. It is also important to note that this study used a single, standardized prompt for all LLMs to ensure a fair comparison. It is possible that model-specific prompt engineering could have further optimized the performance of each individual LLM. Nonetheless, based on the results of this evaluation, the choice of an LLM pipeline involves navigating these varying strengths. The optimal system depends not just on the matching component's precision, but critically on the entire pipeline's end-to-end ability to handle the anticipated linguistic complexity.



# Discussion

## 6.1 Addressing the Research Questions

This research set out to explore the efficacy of LLMs in the context of OIE, semantic triplet matching, and ultimately, text comparison, with a specific application to a German legal use case. The findings presented in Chapter 5 offer direct insights into the research questions posed.

**RQ1: To what extent does LLM-based Open Information Extraction serve as an effective foundation for semantic text comparison in German legal domains?** LLM-based OIE is an effective foundation for this task, particularly due to its ability to produce semantically rich outputs from complex legal language where simpler methods fail. The structured triplets extracted by top-performing models like GPT-4.1-mini, while not flawless, were accurate enough to support a high-performing downstream matching task. This was evident even when accounting for model-dependent variations in extraction style and specific error tendencies.

This effectiveness is primarily supported by the performance of the end-to-end framework, where the configuration utilizing GPT-4.1-mini for both OIE and matching achieved 80.0% accuracy and an MCC of 0.5893 (Table 5.1). The qualitative analysis of OIE outputs (Section 5.1) further corroborated this, illustrating that LLMs could often successfully capture complex relations and nuanced legal concepts from the source texts. While challenges such as the "Loss of Information and Context" were observed, the extracted triplets, on balance, provided a more informative basis for the subsequent matching stage compared to methods that either bypassed structured extraction or relied on simpler techniques. This affirms the foundational utility of LLM-based OIE, suggesting that its strengths in semantic understanding outweigh its current imperfections for this task.

**RQ2: How do LLM-derived triplets compare to dependency graph rule-based triplet extraction when used for semantic text comparison in legal assessment?** When used for semantic text comparison, triplets derived from the top-performing LLMs, such as GPT-4.1-mini, provided a more effective foundation than those from the rule-based method. This is evidenced by the final task performance, where the best LLM-based pipeline (GPT-4.1-mini OIE + matching) achieved a notably higher F1-score of 0.8519 and MCC of 0.5893 compared to the Rule-based OIE pipeline (F1-score of 0.6977, MCC of 0.3679), as shown in Table 5.1.

However, the comparison is not entirely one-sided. The rule-based system proved to be a strong baseline, outperforming the GPT-4o pipeline on the F1-score. This indicates that for sentences with conventional grammatical structures, the deterministic rules can be highly precise. The primary advantage of the higher-performing LLMs stems from their ability to handle the linguistic complexity and diverse phrasing common in legal text, as detailed in the qualitative analysis (Section 5.1). The rule-based system often failed to parse these complex sentences, limiting its recall and overall effectiveness. This finding underscores that while a rule-based approach can be competitive, the superior extraction coverage of advanced LLMs like GPT-4.1-mini is critical for achieving the highest performance on this task, as even a sophisticated matching component is constrained by the quality of its input triplets.

**RQ3: How reliable is LLM-based triplet matching in identifying semantic equivalence between differently worded but conceptually similar information?** LLM-based triplet matching is substantially more reliable than lexical methods like string matching, particularly for handling paraphrasing. As demonstrated by models like GPT-4.1-mini, which achieved a recall of 0.9583, advanced LLMs can consistently identify conceptual similarity despite surface-level syntactic variations.

The qualitative examples (e.g., Section 5.3, "Challenging Semantic Equivalence") demonstrated instances where GPT-4.1-mini successfully identified a match in complex scenarios where other methods, including some LLMs and string matching, faltered. However, this reliability is not absolute. Precision scores varied across models (e.g., GPT-4.1-mini: 0.7667, GPT-4o: 0.8125), and the qualitative analysis revealed that LLMs could occasionally over-rely on keywords or misinterpret nuanced relational meanings, leading to False Positives (e.g., Section 5.3, "Potential False Positive"). Despite these limitations, the substantially lower F1-score (0.7111) and MCC (0.3474) of the String Matching method underscore the considerable advantage LLMs offer in navigating the semantic complexities inherent in this task.

**RQ4: What evaluation metrics best capture the performance of LLM-based OIE systems, particularly when human-annotated ground truth is available?** A comprehensive suite of evaluation metrics, rather than reliance on a single measure, is essential for adequately capturing the multifaceted performance of LLM-based OIE systems, especially when benchmarked against human-annotated ground truth. This is

crucial for understanding not only overall correctness but also specific behavioral aspects like precision-recall trade-offs and robustness to potential class imbalances in the dataset.

While overall Accuracy provides a general sense of performance, metrics like the F1-Score and particularly MCC proved more insightful for a balanced assessment, especially given that MCC is less susceptible to imbalanced classes and was effective in differentiating method performance in this study (Table 5.1). Furthermore, metrics such as Precision, Recall, Specificity, and Balanced Accuracy were indispensable for dissecting model-specific tendencies (e.g., Llama-4’s high Recall but very low Specificity, indicating a strong positive bias). Crucially, this study reinforces that quantitative measures, while vital, must be contextualized by thorough qualitative analysis (as conducted in Section 5.1 for OIE and Section 5.3 for matching). Such qualitative examination is indispensable for uncovering the nuances of error patterns, understanding model behaviors, and identifying limitations that aggregate scores alone cannot reveal, thereby providing a more holistic and actionable assessment of system performance.

## 6.2 Comparison with Existing Work

The findings of this thesis align with and extend the growing body of research into the application of LLMs for OIE. Traditional OIE systems, frequently reliant on rule-based methodologies or earlier machine learning paradigms, have historically grappled with challenges such as linguistic variation, effective domain adaptation, and the extraction of implicit information (Sarawagi et al., 2008; Niklaus et al., 2018). While the advent of neural network-based OIE systems offered improvements (Kolluru et al., 2020; Cui et al., 2018), the few-shot and zero-shot learning capabilities inherent in modern LLMs, as leveraged in this work (Section 4.4), significantly reduce the dependency on large, OIE-specific annotated datasets (Brown et al., 2020).

The systematic application of LLMs to OIE tasks, particularly for combined OIE and subsequent semantic triplet matching, remains an area of active development, as noted in Section 2.4. Existing work has primarily explored aspects such as dynamic prompt engineering. For instance, studies by Ling et al. (2023) and (Qi et al., 2023) investigated the performance of models like LLaMA-2 and GPT-3.5-Turbo for OIE, demonstrating their potential when guided by sophisticated prompting strategies. These studies indicated that LLMs could achieve competitive results, sometimes matching or exceeding specialized OIE systems, though they also highlighted challenges like output inconsistency and the risk of hallucination (Zhang et al., 2023; Huang et al., 2025).

This thesis builds upon these foundational explorations by offering several distinct contributions:

- **Modular LLM-based Pipeline for OIE and Triplet Matching:** A core distinction of this research is its focus on a *modular pipeline* that employs LLMs for both the initial OIE and the subsequent semantic triplet matching phase. This contrasts with studies that might focus predominantly on end-to-end LLM evaluation

for a single task or on the OIE step in isolation. The comparative analysis suggests that this structured, two-stage LLM-driven approach can outperform holistic end-to-end LLM methods and simpler matching techniques for the given task.

- **Evaluation of Newer LLMs:** While much of the cited prior work centered on models such as GPT-3.5-Turbo Ling et al. (2023); Qi et al. (2023), this thesis systematically evaluates a range of more recent and advanced LLMs, including GPT-4o, various GPT-4-mini iterations, Llama series models, and DeepSeek-V3, for their efficacy in both components of the proposed framework. The results provide new insights into the capabilities of these newer architectures for complex OIE and semantic comparison.
- **Application and Validation in a Complex, Non-English Domain:** The framework’s successful deployment and rigorous evaluation on a German legal education dataset directly addresses the noted difficulties and resource scarcity in multilingual OIE, particularly for languages other than English (Ro et al., 2020; Niklaus et al., 2018). The strong performance, especially of models like GPT-4.1-mini, suggests significant multilingual capabilities in current LLMs that can be effectively harnessed for domain-specific OIE and text comparison.

The challenges identified within this thesis, such as managing LLM output variability and ensuring factual accuracy, are consistent with broader concerns in the LLM literature (Ashok and Lipton, 2023; Huang et al., 2025). The mitigation strategies employed, including detailed few-shot prompting and structured output parsing, reflect common and evolving practices aimed at enhancing the reliability of LLM-based systems. This work, therefore, not only benchmarks performance but also contributes to the understanding of how to practically implement and refine LLM-driven solutions for nuanced information extraction and comparison tasks.

### 6.3 Implications of the Findings

The research carries several important implications:

- **Applications in Specialized Domains and Legal Tech:** The successful application to the German legal use case demonstrates the framework’s potential in both educational assessment and legal technology. In education, LLM-based systems can provide scalable evaluation of complex texts, alleviating manual grading burdens. In legal contexts, the ability to extract and compare structured information offers value for document review, case analysis, and compliance verification. This dual applicability highlights the framework’s adaptability to domains requiring nuanced textual understanding.
- **OIE Methodology for Text Comparison:** The research advocates the use of Open Information Extraction to generate structured (subject, relation, object)

triplets as a foundational intermediate step for effective semantic text comparison. This structured approach, particularly when the extracted triplets are subsequently semantically matched (as demonstrated by the primary LLM OIE + LLM Matching pipeline’s superior performance), allows for a more nuanced and accurate evaluation of content equivalence than direct end-to-end LLM comparison or methods relying on less granular, surface-level features.

- **LLM Development for Knowledge Extraction:** The study confirms LLMs as effective tools for OIE, particularly for tasks requiring semantic understanding across different linguistic expressions. The resulting triplet-based representation (as generated by OIE) demonstrates superior utility for fine-grained text comparison. This granular capability enables nuanced content validation that surface-level similarity metrics often miss.
- **Framework Flexibility and Modularity:** Our results validate the effectiveness of the modular design that allows interchangeable LLM components. This approach ensures resilience to model evolution while maintaining consistent performance across contexts. The framework remains viable as LLM technology advances, allowing integration of improved models without significant architectural modifications, while facilitating customization for specific domain requirements and computational constraints.

## 6.4 Limitations and Challenges

Despite the promising results, this study has several limitations:

- **Dataset Scope:** The evaluation was conducted on a dataset specific to one German legal case (the case *Fanny und das Fahrrad*) with a limited number of student responses (5 responses). While allowing for detailed qualitative analysis, this limits the generalizability of the findings to other legal areas, languages, or text types.
- **LLM Selection:** The study tested a specific set of LLMs available at the time of research. The field of LLMs is rapidly evolving, and newer or different models might yield varied performance.
- **Prompt Engineering:** The performance of LLMs is highly sensitive to prompt design. While efforts were made to create effective prompts (Section 4.4), further optimization or different prompting techniques might lead to improved results. Dynamic prompting based on the extraction or matching target seems especially promising.
- **Hallucination and Factual Accuracy:** Although the framework aimed to mitigate this through structured extraction and matching, LLMs can still hallucinate or misinterpret information. Ensuring the factual accuracy of extracted and matched triplets remains a critical challenge.

- **Computational Resources:** Deploying large LLMs can be computationally expensive and may present barriers for practical application in some settings, especially with larger datasets.
- **Evaluation of Extracted Triplets Themselves:** While the end-task of matching criteria was evaluated, a detailed, standalone evaluation of the OIE component’s triplet extraction quality (e.i. evaluation of individual triplet extractions against a gold standard of extracted triplets) was not the primary focus beyond qualitative analysis.
- **Manually Determined Target Triplets:** While manually defined to align with assessment criteria, the target triplets were not validated through systematic testing. Further improvements could be achieved by testing different methodologies for the target triplet representation of the solution scheme.
- **Rule-Based Sentence Segmentation Constraints:** The sentence segmentation approach employs basic heuristics that risk over-segmentation or under-segmentation when processing unconventional syntactic structures or discourse patterns in student writing. This could be mitigated by integrating a linguistically-informed segmentation model trained on educational legal texts.

### 6.5 Future Research Directions

Building on this work, several avenues for future research emerge:

- **Expanded Datasets and Domains:** Evaluating the framework on larger, more diverse datasets, including different legal problem types, other specialized domains (e.g., medicine, finance), and other languages, would enhance understanding of its robustness and generalizability.
- **Exploration of Newer LLMs and Architectures:** Continuously testing and integrating the latest advancements in LLM technology, including multimodal models or models specifically fine-tuned for information extraction or legal text.
- **Advanced Prompting and Fine-tuning Techniques:** Investigating more sophisticated prompt engineering strategies (e.g., chain-of-thought, self-consistency) or parameter-efficient fine-tuning of LLMs on domain-specific data to improve performance.
- **Hybrid Approaches:** Exploring hybrid systems that combine the strengths of LLMs with rule-based systems or traditional NLP techniques to improve accuracy, interpretability, and efficiency. For example, using rules to pre-process text or validate LLM outputs.

- **Factuality Verification and Explainability:** Developing more robust mechanisms for verifying the factual grounding of extracted triplets, prevent hallucination, and improving the explainability of the LLM's decisions in both extraction and matching stages.
- **Long-Context Handling:** Improving the ability to process and reason over entire documents rather than sentence-by-sentence for OIE, to capture inter-sentential relations more effectively.
- **User Interface and Integration:** Developing user-friendly interfaces for educators or legal professionals to interact with the system, review results, and provide feedback for continuous improvement.
- **More Sophisticated Definition of Target Triplets:** Developing advanced methodologies for defining target triplets that more precisely represent complex legal concepts and reasoning chains. This could involve hierarchical triplet structures, weighted importance metrics for different triplets, or collaborative approaches where legal experts iteratively refine triplet definitions to better capture nuanced assessment criteria.
- **Method Optimization:** Systematically exploring the impact of various system parameters (such as hyperparameters, LLM used for extraction, LLM used for matching) to identify optimal configurations for different assessment scenarios.



# Conclusion

## 7.1 Summary of Key Findings

This thesis presented and evaluated an LLM-based framework for OIE and semantic triplet matching, with a practical application to assessing student responses in a German legal education context. The research demonstrated that Large Language Models can be effectively employed to extract structured (subject, relation, object) triplets from complex, domain-specific text and subsequently determine the semantic equivalence between these extracted triplets and predefined criteria.

The key findings indicate that:

- LLM-based OIE, when coupled with LLM-based semantic matching, provides a powerful approach for nuanced text comparison in specialized domains like law, outperforming several alternative methods. The GPT-4.1-mini model configuration was particularly effective.
- LLM-derived triplets are a valuable intermediate representation for comparing textual content, enabling the identification of semantic similarities that go beyond simple keyword or string matching. While also enabling integration into systems requiring structured data, such as knowledge bases.
- LLMs exhibit a strong capability for recognizing semantic equivalence between differently worded statements, though their reliability can be influenced by model choice and prompt design, with challenges in precision and avoiding over-matching or omissions.
- A combination of quantitative metrics (Accuracy, F1-Score, MCC, Precision, Recall) and qualitative analysis is essential for a thorough evaluation of such systems, providing a comprehensive understanding of their performance characteristics.

## 7.2 Contributions of the Thesis

The primary contributions of this thesis are:

- **Contribution of a Validated, Open-Source Modular LLM Framework for Practical Application:** This thesis delivers a fully designed, implemented, and empirically validated open-source framework, accessible at <https://github.com/TamasCsakvari/oie-llm-framework>. This practical tool integrates LLMs for Open Information Extraction and semantic triplet matching, with a modular architecture enabling interchangeable components, adaptation to various LLMs, and serving as a valuable starting point or resource for researchers and practitioners exploring similar tasks.
- **Application and Evaluation in a Novel, Complex Domain:** The successful application and rigorous evaluation of this framework on a German legal dataset, demonstrating its utility in a challenging, real-world scenario. This extends the understanding of LLM capabilities in specialized, non-English contexts.
- **Comparative Analysis of LLMs and Methods:** A systematic comparison of various state-of-the-art LLMs (GPT-4 series, Llama series, DeepSeek) and different methodological approaches (end-to-end vs. modular, LLM-based vs. string-matching/rule-based) for the tasks of OIE and triplet matching.
- **Insights into LLM Performance for OIE and Semantic Matching:** The research provides valuable insights into the strengths and weaknesses of current LLMs for structured information extraction and semantic comparison, including error patterns and the impact of prompting strategies.
- **Foundation for Automated Assessment Tools:** This work lays a foundation for the development of more sophisticated AI-driven tools for automated assessment in education and for knowledge extraction in fields like legal document analysis.

In conclusion, this thesis underscores the transformative potential of Large Language Models in automating complex natural language understanding tasks. Crucially, this work delivers not just findings but also a validated, open-source framework, providing a practical and accessible tool for immediate use in Open Information Extraction, triplet matching, and semantic text comparison tasks. While challenges related to reliability, interpretability, and resource demands persist, the demonstrated capabilities of the proposed framework offer a promising step towards more intelligent and adaptable systems for extracting and comparing information within specialized domains. Future research, building upon the insights and directions identified herein, will be crucial in further advancing this field.





# Overview of Generative AI Tools Used

In accordance with principles of academic integrity and transparent research practices, I declare that several generative AI tools were employed during the preparation of this thesis.

For programming assistance, I utilized Cursor, an AI-enhanced code editor that provided support with code completion, debugging, and refactoring tasks. All code implementations were designed, reviewed, and validated by me to ensure functional correctness and adherence to best practices.

Throughout the research and writing process, I also made use of various large language models (Claude 3.7 Sonnet, GPT-4o, Gemini 2.5 Pro) as assistive tools. These models assisted with phrasing and language refinement, suggested relevant literature to review, and provided feedback on formatting and structuring content. Prompted in the following manner:

- "Please give critical feedback on the attached Chapter X."
- "How can I phrase X in a more straightforward way?"
- "How can I connect ideas X and Y with natural flow?"
- "Can you help me draft a transition paragraph between these two sections? X and Y"
- "What additional detail should I include on methodological choice X?"
- "Please translate sentence X to English."

It is important to emphasize that all AI-generated content was critically evaluated, edited, and thoughtfully integrated into the thesis by me. The conceptual work, analyses, conclusions, and scientific contributions presented in this work are my own original intellectual efforts. The AI tools served solely as assistive technologies to enhance productivity and language quality, not to generate original research insights or conclusions.

I affirm that I have maintained intellectual ownership of this work and that the use of these tools complies with the ethical guidelines for scientific work at the Vienna University of Technology.

# List of Tables

5.1	Performance metrics for the nine evaluated methods . . . . .	55
-----	--	----

# List of Figures

1.1	A high-level overview of the proposed LLM-based pipeline built in the proposed framework. The process begins with unstructured text, which is transformed into structured (subject, relation, object) triplets using an LLM. These extracted triplets are then semantically compared against target triplets to evaluate the text's content. . . . .	3
3.1	The representative case <i>Fanny und das Fahrrad</i> . . . . .	20
3.2	The solution scheme for the case <i>Fanny und das Fahrrad</i> . . . . .	22
3.3	An example student answer for the case <i>Fanny und das Fahrrad</i> . . . . .	24
4.1	A conceptual overview of the primary LLM-based pipeline. The flowchart illustrates the key processing stages from raw text input to matched triplet generation, including sentence segmentation, LLM-based Open Information Extraction (OIE), and triplet matching. . . . .	28
4.2	A detailed illustration of the pipeline's data flow using a concrete example. The flowchart traces a sample input text ("Good-faith acquisition...") through each processing stage. It shows the transformation from a sentence into candidate triplets, the comparison of these against target triplets, and the resulting interpretation into a cell of the predicted matching matrix, which is then evaluated against the reference matrix. . . . .	31

4.3	Comprehensive overview of the entire experimental setup. This diagram contrasts the primary LLM-based pipeline with the alternative methods evaluated in this study. It visualizes how the components for triplet extraction and matching are either replaced or bypassed in the End-to-end LLM, Rule-based OIE, and String Matching methods, providing a complete picture of the comparative analysis. . . . .	38
-----	---	----

## Listings

4.1	Few-shot LLM prompt used for triplet extraction . . . . .	32
4.2	Few-shot LLM prompt used for triplet matching . . . . .	33
4.3	Manually defined target triplets, one target triplet for each of the criteria shown in Figure 3.2 . . . . .	36
4.4	Few-shot LLM prompt used for end-to-end text-criterion assesment . .	42

# Bibliography

- Kiran Adnan and Rehan Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771, 2019.
- Alan Akbik and Alexander Löser. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, 2012.
- Dhananjay Ashok and Zachary C Lipton. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*, 2023.
- Akim Bassa, Mark Kröll, and Roman Kern. Gerie-an open information extraction system for the german language. *J. Univers. Comput. Sci.*, 24(1):2–24, 2018.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.
- Sangnie Bhardwaj, Samarth Aggarwal, et al. Carb: A crowdsourced benchmark for open ie. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, 2019.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. *arXiv preprint arXiv:1805.04270*, 2018.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Tianyu Ding, Tianyi Chen, Haidong Zhu, Jiachen Jiang, Yiqi Zhong, Jinxin Zhou, Guangzhi Wang, Zhihui Zhu, Ilya Zharkov, and Luming Liang. The efficiency spectrum of large language models: An algorithmic survey. *arXiv preprint arXiv:2312.00678*, 2023.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.

Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, pages 10–18, 2012.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ahmed Abdeen Hamed, Malgorzata Zachara-Szymanska, and Xindong Wu. Safeguarding authenticity for mitigating the harms of generative ai: Issues, research agenda, and policies for detection, fact-checking, and ethical ai. *IScience*, 27(2), 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl,

Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International conference on machine learning*, pages 4411–4421. PMLR, 2020a.

Xuming Hu, Chenwei Zhang, Yusong Xu, Lijie Wen, and Philip S Yu. Selfore: Self-supervised relational feature learning for open relation extraction. *arXiv preprint arXiv:2004.02438*, 2020b.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*, 2017.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. Imojie: Iterative memory-based joint open information extraction. *arXiv preprint arXiv:2005.08178*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. Evaluating chatgpt’s information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*, 2023.

- Tongliang Li, Zixiang Wang, Linzheng Chai, Jian Yang, Jiaqi Bai, Yuwei Yin, Jiaheng Liu, Hongcheng Guo, Liqun Yang, Hebboul Zine el abidine, et al. Mt4crossoie: Multi-stage tuning for cross-lingual open information extraction. *Expert Systems with Applications*, 255:124760, 2024.
- Zichao Lin, Shuyan Guan, Wending Zhang, Huiyan Zhang, Yugang Li, and Huaping Zhang. Towards trustworthy llms: a review on debiasing and dehallucinating in large language models. *Artificial Intelligence Review*, 57(9):243, 2024.
- Chen Ling, Xujiang Zhao, Xuchao Zhang, Yanchi Liu, Wei Cheng, Haoyu Wang, Zhengzhang Chen, Takao Osaki, Katsushi Matsuda, Haifeng Chen, et al. Improving open information extraction with large language models: A study on demonstration uncertainty. *arXiv preprint arXiv:2309.03433*, 2023.
- Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson DAutume, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR, 2022.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- Pai Liu, Wenyang Gao, Wenjie Dong, Songfang Huang, and Yue Zhang. Open information extraction from 2007 to 2022—a survey. *arXiv preprint arXiv:2208.08690*, 2022.
- Meta AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. Meta AI Blog, April 2025. URL <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>. Accessed: 2025-05-03.
- Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.
- Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. Open information extraction for knowledge graph construction. In *Database and Expert Systems Applications: DEXA 2020 International Workshops BIOKDD, IWCFSS and MLKgraphs, Bratislava, Slovakia, September 14–17, 2020, Proceedings 31*, pages 103–113. Springer, 2020.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.
- Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. *arXiv preprint arXiv:1806.05599*, 2018.

- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Ji Qi, Kaixuan Ji, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Lei Hou, Juanzi Li, and Bin Xu. Mastering the task of open information extraction with large language models and consistent reasoning environment. *arXiv preprint arXiv:2310.10590*, 2023.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- 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. *Journal of machine learning research*, 21(140): 1–67, 2020.
- Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi $\text{Q}$ oie: Multilingual open information extraction based on multi-head attention with bert. *arXiv preprint arXiv:2009.08128*, 2020.
- Swarnadeep Saha, Harinder Pal, et al. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, 2017.
- Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.
- Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.
- Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534, 2012.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.
- Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

- Jacob Solawetz and Stefan Larson. Lsoie: A large-scale dataset for supervised open information extraction. *arXiv preprint arXiv:2101.11177*, 2021.
- Linfeng Song, Ante Wang, Xiaoman Pan, Hongming Zhang, Dian Yu, Lifeng Jin, Haitao Mi, Jinsong Su, Yue Zhang, and Dong Yu. Openfact: Factuality enhanced open knowledge extraction. *Transactions of the Association for Computational Linguistics*, 11:686–702, 2023.
- Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, 2016.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, 2018.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Interspeech*, volume 2012, pages 194–197, 2012.
- D Thenmozhi and G Ravi Kumar. An open information extraction for question answering system. In *2018 International Conference on Computer, Communication, and Signal Processing (ICCCSP)*, pages 1–5. IEEE, 2018.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv e-prints*, pages arXiv–2302, 2023.
- Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357, 2024.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

- Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007.
- Dongxu Zhang, Subhabrata Mukherjee, Colin Lockard, Xin Luna Dong, and Andrew McCallum. Openki: Integrating open information extraction and knowledge bases with relation inference. *arXiv preprint arXiv:1904.12606*, 2019.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren’s song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.
- Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.
- Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, Haiyang Yu, Jian Sun, and Yongbin Li. A survey on neural open information extraction: Current status and future directions. *arXiv preprint arXiv:2205.11725*, 2022.