TU WIEN Informatics

Rule-Based Open Information Extraction on German Legal Domain

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Master programme Data Science

by

Zsombor Iszak, BSc.
Registration Number 11709501

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Gábor Recski, PhD.

Signed by:	Zsombor Iszak
Date:	24.01.2025 10:28:36

This document is digitally signed!
This document bearing a qualified electronic signature has the same legal standing as a handwritten document in accordance with Article 25 (2) of Regulation (EU) No. 910/2014 of 23 July 2014 ('eIDAS-VO')
Validation: Information for checking the electronic signature can be found at: www.a-trust.at/pdf

SIGNATURINFORMATIONEN	Signatory	Gabor Andras Recski
	Date/Time-UTC	2025-01-23T20:37:44+0100
	Verification	Information about the verification of the electronic signature can be found at: https://www.signaturpruefung.gv.at
Note	This document is signed with a qualified electronic signature. According to Art. 25 para. 2 of the Regulation (EU) No 910/2014 of 23. July 2014 ("eIDAS-Regulation") it shall have the equivalent legal effect of a handwritten signature.	

Vienna, January 27, 2025

Zsombor Iszak	Gábor Recski

Erklärung zur Verfassung der Arbeit

Zsombor Iszak, 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, 27. Jänner 2025

Zsombor Iszak

Danksagung

In erster Linie möchte ich meinem Betreuer, Gábor Recski, für seine kontinuierliche Unterstützung, seinen unermüdlichen Einsatz und seine Anleitung danken, die es mir ermöglicht haben, diese Arbeit zu schreiben. Ihre Anleitung und Ihre Leidenschaft für die Forschung und das Forschungsthema waren für mich eine unschätzbare Inspiration. Ich konnte mich immer an Sie wenden, wenn ich vor Herausforderungen stand. Ich schätze es sehr, dass Sie mir schnell und umfassend Feedback gegeben haben, sogar auch am Wochenende oder in den Ferien. Ich bin Ihnen unendlich dankbar für Ihre Unterstützung und Anleitung.

Mein aufrichtiger Dank gilt auch meiner Familie, meiner Partnerin und meinen Freunden. Eure unschätzbare Unterstützung, die gemeinsam verbrachte Zeit und unsere gemeinsamen Erlebnisse waren eine Quelle der Kraft während meiner gesamten Ausbildung und darüber hinaus. Danke, dass Ihr immer für mich da seid und mir erlaubt, meine Ziele zu verfolgen.

Abschließend möchte ich mich bei meinen Kollegen und meinem Vorgesetzten dafür bedanken, dass sie mir eine flexible Arbeitszeitgestaltung ermöglicht haben, die es mir erlaubt hat, auf meine akademischen Bemühungen, insbesondere auf meine Masterarbeit, zu konzentrieren.

Acknowledgements

First and foremost, I would like to thank my supervisor, Gábor Recski, for his continuous support, relentless effort, and guidance under which I was able to write this thesis. Your guidance and passion for research and the research topic have been an invaluable inspiration to me. I could always turn to You for assistance whenever I faced challenges. I very much appreciate the quick and extensive feedback You gave me, even during the weekend or the holidays. I am immensely grateful for your support and guidance.

In addition, I would like to express my sincere gratitude to my family, my partner, and my friends. Your invaluable support and shared experiences have been a source of strength throughout my education and beyond. Thank you for always being there for me, and for allowing me to pursue my goals.

Finally, I would like to thank my colleagues and my manager for allowing me a flexible working schedule, which has enabled me to focus on my academic endeavors, specifically on my thesis.

Kurzfassung

In der Vergangenheit wurden Informationen in schriftlicher Form als Text gespeichert. Text ist im Allgemeinen eine unstrukturierte Form von Information, die zwar für den menschlichen Gebrauch geeignet ist, aber aufgrund ihres unstrukturierten Zustands nicht die effizienteste Art der maschinellen Informationsextraktion darstellt. Die Arbeit basiert auf einem Geschäftsfall, der darauf abzielt, die Lösungen von Jurastudenten zu bewerten und Feedback zu Fehlern zu geben. Bewertungsrichtlinien werden von Fachexperten bereitgestellt, die erleuchten, wie die Versuche zu bewerten sind und welche Informationen relevant sind. Ein regelbasiertes System zum Open Information Extraction (OIE) wurde entwickelt, um Informationssegmente aus den Lösungen der Studenten zu extrahieren. OIE leitet strukturierte Informationen aus unstrukturiertem Text ab, ohne Einschränkung durch den Beziehungstyp. Die regelbasierte OIE basiert auf der Kombination eines Regelwerkes und eines Matching-Algorithmus und bietet Erklärbarkeit, was eine transparente Entscheidungsfindung ermöglicht, die sowohl für den juristischen Bereich als auch für den vorliegenden Geschäftsfall entscheidend ist. Ein genau definiertes Regelwerk führt zu einer erklärbaren Informationsextraktion in der Zieldomäne, schränkt aber die Fähigkeit des Modells ein, sich zu verallgemeinern, wenn das Vokabular oder die Formulierung geändert wird. Das wissenschaftliche Ziel ist die Verallgemeinerungsfähigkeit von graphenbasierten Universal Dependency (UD) Regelsystemen über ber Rechtstexte zu untersuchen, die zu verschiedenen Rechtsfällen gehören.

In dieser Arbeit wird eine Kombination eines Regelwerkes und eines Matching-Algorithmus erstellt, die perfekt auf den ausgewählten Rechtsfall passt und sowohl einen Recall als auch eine Präzision von 1 erreicht. Ausgehend von dieser sehr fallspezifischen Kombination werden Verallgemeinerungsschritte unternommen, um die optimale Kombination von Regeln und Matching-Algorithmus zu finden, die nicht nur bei Rechtstexten des Zielfalls, sondern auch bei solchen aus anderen Rechtsfällen gut funktioniert. Sowohl der Matching-Algorithmus als auch das anfangs definierte Regelwerk werden im Rahmen der Generalisierungsschritte angepasst. Die Generalisierungsfähigkeit wird qualitativ und quantitativ bewertet. Der definierte Themenbereich präsentiert themenspezifische Herausforderungen. Die Auswirkungen dieser Herausforderungen werden erörtert, und potenzielle Lösungen werden vorgeschlagen, die gleichzeitig die Richtung der zukünftigen Arbeit definieren. Die Umsetzung dieser Lösungen im Rahmen dieser Arbeit ist jedoch nur eingeschränkt möglich, da weiteres Domänenwissen erforderlich ist.

Abstract

Historically, information has been stored in written form as text. Text is generally an unstructured form of information that, while suitable for human use, is not the most efficient way to extract information by machines due to its unstructured state. The thesis is based on a business case, which aims to evaluate the solutions of legal students and provide feedback on errors. Evaluation guidelines are provided by domain experts who know how to evaluate the attempts and what information is relevant. A rule-based Open Information Extraction (OIE) system is designed to extract information segments from the student attempts. OIE derives structured information from unstructured text, unrestricted by relation type. Rule-based OIE is based on the combination of a set of rules and a matching algorithm and provides explainability, enabling transparent decision making that is critical to both the legal domain and the business case at hand. A strictly defined set of rules leads to an explainable information extraction on the target domain but limits the ability of the model to generalize if the vocabulary or phrasing is changed. The scientific aim is to investigate the generalization capability of the Universal Dependency (UD) graph-based rule systems over legal texts belonging to diverse legal cases.

In this study, a set of rules is created with a combination of a matching algorithm, that works perfectly on the target legal case achieving both a recall and precision of 1. Starting from this highly case-specific combination, generalization steps will be taken to find the optimal combination of rules and matching algorithm, that can perform well not only on legal texts belonging to the target case but also on those belonging to different legal cases. As part of the generalization steps, both the matching algorithm and the initial set of rules will be adjusted. The generalization ability is evaluated both qualitatively and quantitatively. The specified domain poses challenges unique to the legal field. The implications of these are discussed and potential solutions are proposed outlining future work.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Historically, information is stored in a written form as text. Text is in general an unstructured form of information, which, although suitable for human use, is not the most efficient way of information extraction for computers given its unstructured state. A technology, that aims to provide structured information based on unstructured text, hence more efficient access to information, is called Information Extraction (IE) as part of the field of Natural Language Processing (NLP) (Jiang, 2012). There are multiple techniques to extract structured information, each having different strengths and weaknesses. Therefore the requirements of the specific domain influence the choice of suitable approaches. Open Information Extraction is a technique, that extracts structured information from an unstructured text without requiring a predefined set of relations. Rule-based OIE is more rigid compared to general OIE models, but can achieve high accuracy for target texts, for which the rules are well-defined, making it useful in controlled, domain-specific scenarios. The business case the research is created for aims to establish an Artificial Intelligence (AI) based feedback system for law students. The students are required to explain and solve a legal case and the system is supposed to evaluate the students' attempts based on the completeness of the solution. The system needs to be able to decide, whether a student mentioned the requested information segments or not. For each case, a detailed solution guideline will be provided describing the case-relevant information. Therefore extraction and generalization are not expected to work for all kinds of information from any legal text. The application is limited to the business case. This also limits the domain of the generalization.

The thesis aims to examine an explainable Open Information Extraction (OIE) approach relying on linguistic rules. On one hand, the usage of linguistic rules to extract patterns and information from written text enables the user to trace back, why a specific section of text is extracted. This is a significant advantage over state-of-the-art deep-learning

(DL) OIE models. On the other hand, the rule-based approach does not require a large amount of data, only a set of rules. How this set of rules is created, in combination with the matching algorithm, determines both the approach's information extraction capability and the generalization capability over multiple legal texts belonging to different legal cases. The thesis focuses on the domain of German legal texts. The aim is to investigate the generalization capability of rule-based Open Information Extraction (OIE) over diverse texts of various legal cases provided for the business case.

In the following section 1.2 the scientific problem and the research questions are introduced. Chapter 2 aims to provide an overview for the reader in the field of rule-based Open Information Extraction, by introducing the used definitions, the related work, and the used framework. In chapter 3 the used data, the business case, and the derived use case are explained. The used metrics, the node matching algorithm, and the methodology are defined in chapter 4, finally in chapter 5 the results are presented and discussed outlining the future work.

1.2 Problem Statement

The legal field is a particularly delicate domain, as decisions depend on arguments, subtle differences in wording, and interpretation of the situation. Hence, explainability and transparency are crucial in this field. Deep Learning (DL) models, such as Large Language Models (LLM) deliver good results regarding information extraction. Despite their success, deep networks are used as black-box models with outputs that are not easily explainable during the learning and prediction phases. This lack of interpretability is significantly limiting the adoption of such models in domains where decisions are critical such as the medical and legal fields (Zini and Awad, 2022). An alternative technique is the supervised Relation Extraction (RE) approach. RE is based on annotated labels, so it requires large-scale human-annotated data, which is expensive and does not scale to new domains or many relations (Vania et al., 2022). This is an issue especially if the amount of data available is limited.

Open Information Extraction converts the unstructured text to semi-structured tuples of parts of the text (Niklaus et al., 2018), typically of the format predicate(subject, object). OpenIE tuples have shown utility in various downstream tasks (Mausam, 2016) like Question Answering (Khot et al., 2017), Machine Reading (Poon et al., 2010) and Multi-Document Summarization (Fan et al., 2019). With the widespread adoption of Deep Learning in NLP, OpenIE systems have gone through a paradigm shift from using rule-based, statistical systems to supervised neural models. Both types of OIE systems have been limited to only a few languages. Rule-based systems require language-specific OpenIE insights and a set of rules. DL systems require even a larger amount of annotated training corpus that poses a barrier (Kolluru et al., 2022), not to mention the training cost of large models and their environmental impact. The rule-based approach is a better option for the legal domain, as it does not require a large amount of annotated German legal corpus and the extractions are explainable thanks to the defined rules the extraction is based on.

The scientific problem of rule-based OIE is, that its generalization capability is limited (Yu et al., 2022). A precisely defined set of rules leads to an explainable information extraction on the target domain with potentially high recall but also limits the capability of the model to generalize on other domains. The examined business case is in the domain of German legal texts involving various legal cases, which differ from one another as they each describe a different topic and consequently use different vocabulary. Therefore, one can test the generalization capability of rule-based OIE on this set of legal cases.

To the best of our knowledge, rule-based OIE in the domain of German legal corpus is not a widely researched field. It is unclear, what set of rules performs well on German legal cases and how well a set of rules developed on one case generalizes over multiple, unfamiliar legal cases. Therefore the scientific question, my thesis aims to investigate, is about the generalization capability of rule-based Open Information Extraction extraction over multiple legal texts belonging to diverse legal cases.

Accordingly, the following three research questions were defined for my thesis:

- What are the syntactic patterns, that characterize those text fragments, that pose the main challenge for rule-based Open Information Extraction on the set of legal cases in the German language?
- How does the rule system developed on one legal case generalize over multiple cases in terms of recall and precision?
- How much can the model's accuracy be improved by domain-specific methods, such as incorporating abbreviations and synonyms?

Hence the task is to find a set of rules that enables a reliable, explainable, and accurate information extraction on the specified domain, i.e. on a chosen legal case. Then it is tested how well this rule set developed for one legal case generalizes over text belonging to other, unfamiliar legal cases using alternative matching algorithms.

CHAPTER 2

Background

2.1 Open Information Extraction

OIE differs from other Information Extraction (IE) approaches. Traditional Information Extraction, such as Relation Extraction (RE), depends on predefined categories to identify relationships (Pai et al., 2024). As shown in Figure 2.1, OIE operates without such constraints. This allows it to extract a wide range of diverse and unforeseen relationships. OIE does not specify a precise set of relations and arguments, it only requires that the extracted words must be derived from the text. OIE does not tell what the exactracted relation is, it does not tell what entities are contained in the extracted structured tuple, but merely extracts elements, and phrases, that are part of the text. At the same time, these extracted elements can be in any kind of relation with each other, the relation does not need to be defined beforehand. Therefore the challenge is that all the relations must be extracted from the text because there are no predefined relations. These extracted relations do not require further deduction to a canonical form.

To showcase an example output from an example sentence: *Johann Strauss was born in Vienna.* A widely used IE, like RE, outputs the following structured extraction:

- Relation: born_in
- Subject: Johann Strauss
- Object: Vienna

For this precise, structured extraction, RE relies on the predefined relation *born_in*. On the contrary, OIE does not require any predefined relation but also does not track back the extracted structure to any canonical form. The OIE's output might be as simple (Johann Strauss, born in, Vienna).

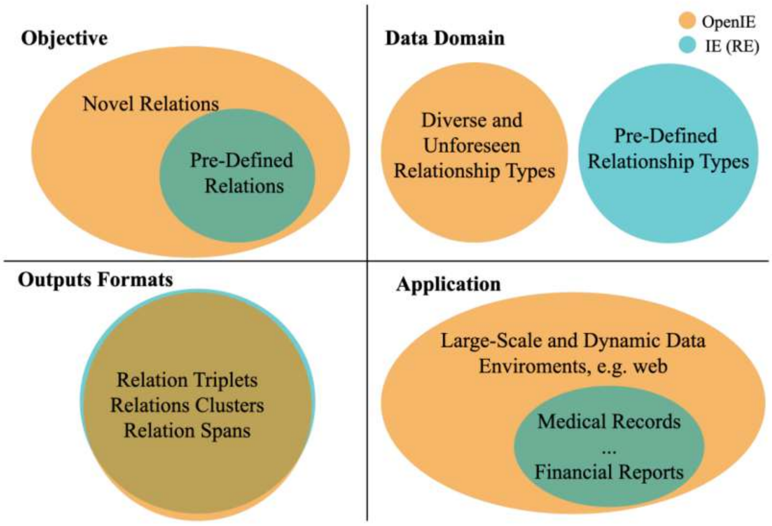


Figure 2.1: Comparison of OIE and standard Relation Extraction (RE). Source: Pai et al. (2024)

One type of Open Information Extraction is called rule-based OIE. Rule-based OIE utilizes linguistic tools and integrates more complex syntactic and semantic features while preserving the intuitive task of directly extracting relational triplets from text Pai et al. (2024). The extraction relies on linguistic patterns created by dependency parsing, part-of-speech tagging, or semantic roles. However, the exact relations are still not defined, which is a differentiating factor compared to RE. Rule-based OIE uses for example a rule describing a subject-verb-object structure, hence extracts all those matching structures from the text and outputs a tuple, like it was shown before: (Johann Strauss, born in, Vienna). The output is still not matched to an ontology or canonical form.

The rule-based OIE used in this thesis is based on patterns over Universal Dependency (UD) parsed representations of sentences. UD (Nivre et al., 2018) is a project that develops cross-linguistically consistent treebank annotation for many languages, to facilitate multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective. The annotation scheme is based on an evolution of (universal) Stanford dependencies (de Marneffe et al., 2014), Google universal part-of-speech tags (Petrov et al., 2012).

Sentences of the text are parsed using a custom-modified parser built on top of Stanza pipeline (Qi et al., 2020). The parser returns a UD-parsed representation of the sentence as a directed graph, which represents the syntactic structure of the sentence. A good example of a parsed tree is Figure 2.2. The nodes of the parsed graph are the words of the sentence, while the edges of the graph connecting the nodes represent the grammatical relations between the connected words. These grammatical relations are categorized

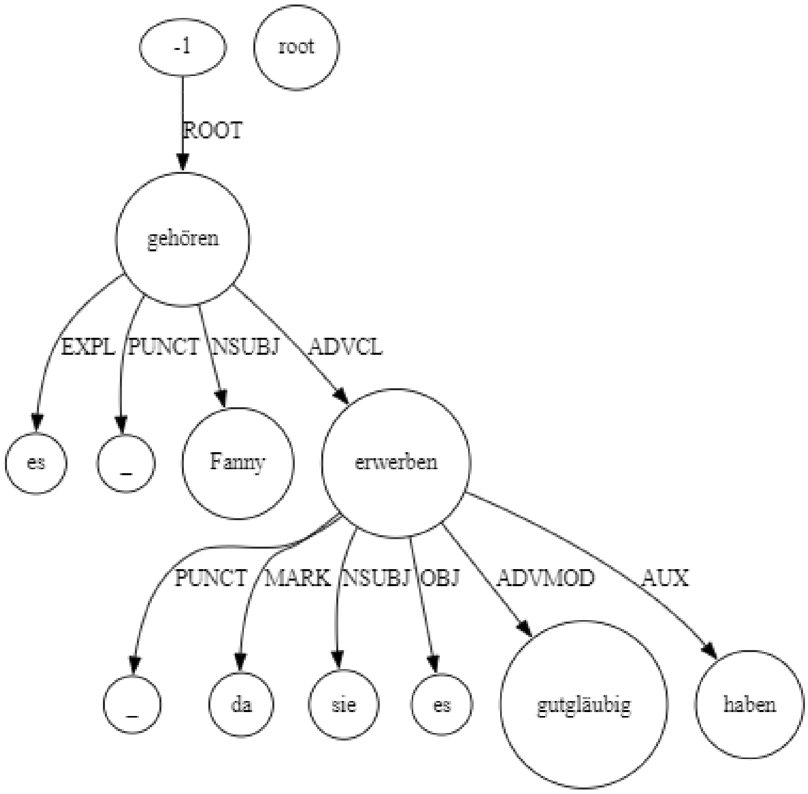


Figure 2.2: UD parsed tree representation of the German sentence: *Es gehört Fanny, da sie es gutgläubig erworben hat.* The sentence translates to *It belongs to Fanny, as she acquired it in good faith.*

according to the UD relations.

This universal dependency graph could be also written in Penman notation (Kasper, 1989). Penman notation is a serialization format for the directed, rooted graphs used to encode semantic dependencies. A simple example for this notation is if we take a subgraph of the sentence graph shown in Figure 2.2: Two connected words, *erwerben* and *gutgläubig*, are picked. The edge between them is labeled as *ADVMOD* referring to the adverbial modifier UD relation. The subgraph could be written in Penman notation as *(u_8 / erwerben :ADVMOD (u_7 / gutgläubig))*. This penman notation equals to the graph representation shown in Figure 2.3.

2.2 Definitions

Some definitions used later need to be defined. Rule-based Open Information Extraction requires a set of rules to extract triplets, i.e. the information. A triplet is a structured representation of information. It contains a predicate and arguments. The number of

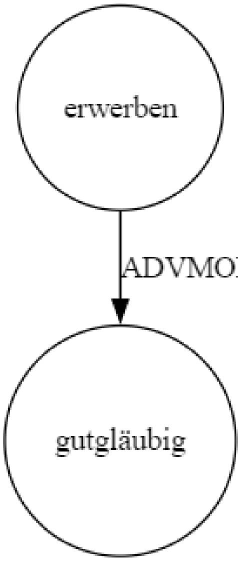


Figure 2.3: Subgraph of the UD parsed tree representation of the German sentence: *Es gehört Fanny, da sie es gutgläubig erworben hat.* The subgraph written in Penman notations looks as follows: *(u_8 / erwerben :ADVMOD (u_7 / gutgläubig))*

Der	derivative	Erwerb	von	Eigentum	scheidet	aus	,...
The	derivative	acquisition	of	ownership	ruled	out	,...

Table 2.1: German sentence: *Der derivative Erwerb von Eigentum scheidet aus,...*, that translates to *The derivative acquisition of property is ruled out,...*

arguments can range from zero to many. An example triplet is *gutgläubig(erwerben)*. In this example, the word before the parenthesis is the predicate, the German adjective *gutgläubig*. In the parenthesis the arguments are listed, this time only one, the German verb *erwerben*. Both the predicate and an argument can consist of multiple words or a phrase. *scheidet_aus(derivative_Erwerb)* from the sentence shown in Table 2.1 is a valid triplet, where the single argument is the phrase*derivative_Erwerb* and the predicate is *scheidet_aus*, a German verb with its preposition.

Triplets are extracted based on rules. Rules can have many forms but must consist of essentially 2 things: a graph and an indication of what is the predicate and what is an argument. An example rule looks as follows:

(("(u_8 / erwerben :ADVMOD (u_7 / gutgläubig))",), (),"attempt21"):((2,),(1,)).

Above the Penman notion form of the subgraph 2.3 is used. The structure is as follows: the graph information is stored before the colon, and the triplet structure is stored after the colon. The graph is the so-called (extraction) pattern. The words represent the nodes of the graph and the edges are the UD relations among them. This graph is a subgraph of the UD graph representation of the parsed sentence (see in Figure 2.2) the triplet was annotated on. In the example, the edge *:ADVMOD* is the UD relation adverbial modifier, between the two nodes: *erwerben* and *gutgläubig*. The previous rule is created on a sentence from attempt 1 of case 1 shown in Figure 2.2. The triplet annotation is *gutgläubig(erwerben)* and involves two words, i.e. the two nodes, *gutgläubig* and *erwerben*. The shortest path between the words of a triplet forms a subgraph and that subgraph is the pattern for the annotated triplet. In this example, the shortest path between the two words defines the subgraph shown in Figure 2.3.

The triplet structure after the colon is the additional indication of which words are predicates and which are arguments: ((2,),(1,)). The first parentheses contains the indices of the words of the predicate and the following parentheses contain the indices of the words of each argument. In the above example, the number 2 in the first parentheses means, that the second word in the graph is the predicate. In the example, this is *gutgläubig*. There is only one following parentheses meaning, that triplet has only one argument. The number 1 in the parenthesis indicates, that the first word in the graph, *erwerben*, is the only argument of the single triplet.

Creating rules, and patterns can be done manually, but in NEWPOTATO (see section 2.4 an annotation algorithm is implemented. This annotation enables one to select words from a sentence in a specific order and by selecting the words one defines the predicates and arguments. Based on this selection and the parsed representation of the sentence the pattern is extracted and the rule is created.

When a triplet is extracted in rule-based Open Information Extraction, the extraction pattern, i.e. the subgraph, is compared to the parsed representation of the target sentences. If the extraction pattern matches a part of the sentence graph, then the matched part is extracted from the sentence. The precise matching logic is described in section 2.4.

Rules are created on a text or a domain, on which they perform well, reaching high recall and even high precision. The question of generalization is the capability of how the information extraction performs on a text, on a domain it was not created for. The generalization is characterized by the metrics of recall and precision with the combination of the number of extracted triplets. The two factors limiting generalization are the diverse used vocabulary and the diverse formulation and phrasing of the language.

2.3 Related Work

The domain of interest for information extraction is German legal text. Legal field requires transparent decision making (Zini and Awad, 2022), hence an explainable feedbacksystem is required for the business case. This explainability is a crucial restricting condition for the extraction algorithm.

Designing a Relation Extraction (RE) could be a feasible approach. However, high-performance RE models require large-scale human-annotated data, which is expensive and does not scale to a large number of relations or new domains (Vania et al., 2022). As shown in Figure 2.1 and described in section 2.1, OIE does not require predefined relations and structures. This adaptability makes OIE particularly valuable for rapidly evolving Natural Language Processing (NLP) applications such as question-answering (Han et al., 2020), making OIE well suited for the business case.

Open Information Extraction focuses on extracting structured information from unstructured text sources (Niklaus et al., 2018). It typically represents relationships as triplets in the form of (arg1, rel, arg2). Open Information Extraction was initially introduced in the form of a system called TextRunner in 2007 Yates et al. (2007). Yates et al. (2007) defines open information extraction as an unsupervised task, that extracts structure, i.e. triplets from a vast corpus of unstructured web text. Since its inception in 2007, OIE has seen continuous progress. Early approaches relied on simple linguistic tools, but modern OIE models have increasingly incorporated advanced syntactic and semantic features while maintaining the straightforward task of extracting relational triplets directly from the text. The introduction of neural models in 2019 marked a significant turning point, with Transformer-based architectures like BERT (Devlin et al., 2018) dramatically improving feature extraction capabilities. To align with these technological advancements, diverse methods, and task settings have emerged within the evolving landscape of OIE approaches. Applications, such as RnnOIE formulate the OIE as a sequence tagging problem, addressing challenges such as encoding multiple extractions for a predicate (Stanovsky et al., 2018). Another typical approach is the usage of an encoder-decoder model generating a sequence of relation triplets conditioned by the input sentence. A good example is NeuralOIE (Cui et al., 2018). Recent studies, such as Ling et al. (2023), have employed LLMs for OIE tasks by transforming input text through specific instructions or schemas. The recent advancements in Open Information Extraction deliver top results on benchmarks, but the underlying issue is the black-box nature of neural network models, which contradicts the requirement of the business case, i.e. the transparent and explainable feedback system.

Therefore, the older, but explainable approach of rule-based Open Information Extraction is applied. The initial work, Yates et al. (2007), was a self-supervised method based on a heuristical approach. Shortly after, REVERB (Fader et al., 2011) was introduced, REVERB did not require a pre-specified vocabulary anymore. Its improvement was the introduction of two simple syntactic and lexical constraints on binary relations expressed by verbs. The syntactic limitation is expressed in terms of POS-based regular expressions, covering about 85% of verb-based relational phrases in English text, as Fader et al. (2011) revealed in linguistic analysis. The lexical constraint is based on the idea, that a valid relational phrase should take many distinct arguments in a large corpus. This constraint helps to avoid overspecified relational phrases. The drawback of REVERB is, that it focuses on binary relations and suffers a significant quality loss for the task of extracting higher-order N-ary facts. This quality loss may not only affect the correctness but also the completeness of an extracted fact. Akbik and Löser (2012) presented Kraken, which is an approach built for extracting complete, N-ary facts from sentences by gathering the full set of arguments for each relational phrase within the sentence. The extraction is based on hand-written rules over typed dependency parsers. KRAKEN is a high-precision OIE approach that captures more facts per sentence at greater completeness than existing OIE approaches at the time but is vulnerable to noisy and ungrammatical text.

EXEMPLAR, introduced in Mesquita et al. (2013), works both for binary and n-ary relations. A key idea in semantic approaches is to identify the precise connection between the argument and the predicate words in a relation. Mesquita et al. (2013) applies this key ides over a dependency parse tree. The goal is to achieve the higher accuracy of the semantic approaches at the lower computational cost of the dependency parsing approaches.

Stanovsky et al. (2016) argues, that while much semantic structure is expressed by syntax, many phenomena are not easily read out of dependency trees, often leading to further ad-hoc heuristic post-processing or to information loss. As a solution Stanovsky et al. (2016) proposes PROPS, which relies on a more semantically-oriented representation of a sentence. The dependency parsed tree is transformed by a proposed rule-based converter into a directed graph which is tailored to directly represent the proposition structure of an input sentence.

PredPatt (White et al., 2016) is built on a similar idea. It employs a set of unlexicalized rules defined over Universal Dependency parses (de Marneffe et al., 2014) to extract predicate-argument structures. In doing so, PredPatt constructs a directed graph, where a special dependency argument is built between the head token of a predicate and the head tokens of its arguments, while the original UD relations are preserved within predicate and argument phrases. PredPatt uses language-agnostic patterns on UD structures, hence, it is one of the few Open IE systems that work across different languages (Niklaus et al., 2018).

To the best of our knowledge, there are no datasets suitable for information extraction on the German legal domain besides GerDaLIR (Wrzalik and Krechel, 2021). GerDaLIR is designed for a specific use case: extract references among legal cases. Passages containing one or more references to known cases become queries while the referenced cases are labeled as relevant. It has a large corpus size having 123K queries, each labeled with at least one relevant document out of the collection of 131K case documents. While the dataset is suitable for extracting passage references, it is not our primary task, hence the dataset is unsuitable for our goal.

Rule-based OIE systems were initially developed for the English language, however, there are approaches to overcome the language barrier. OIE models were developed for multi-language usage, that can also work on German language (Kotnis et al., 2022). Kotnis et al. (2022) has a modular, iterative structure and integrates a rule-based extraction system into a neural end-to-end system. An alternative approach is to extend an existing English-based model to the German language. PropsDE (Falke et al., 2016) is an adaption of the original PropS system (Stanovsky et al., 2016) to the German language and is a good example of this approach. It is shown, how an English rule-based system, such as PropS, can be adapted to the German language. Finally, there are experiments to identify linguistic concepts unique to the German language and build an OIE model dedicated to the German language (Bassa et al., 2018). Bassa et al. (2018) is based on the output of a German dependency parser and several handcrafted rules to extract the propositions.

Finally, I would like to address the question of evaluation, as it is not straightforward for Open Information Extraction. To evaluate the extraction, a decision needs to be made, on what counts as a useful extraction, i.e. as a true positive. The essence of Open Information Extraction is, that a wide range of diverse and unforeseen relationships gets extracted, hence each extracted triplet needs to be evaluated one by one. To enable a reliable and reproducible evaluation, precise and detailed guidelines needs to be defined specifying what is a true positive extraction and what counts as false positive extraction. The requirements for the evaluation need to be pinpointed.

The decision of true positive extractions can be made on the token and on the fact level. Token-level scorer is Lechelle et al. (2019). It penalizes the verbosity of automated extractions as well as the omission of parts of a gold triplet by computing precision and recall at the token level in WiRe57. Their proposed precision is the proportion of extracted words that are found in the gold triplet, while recall is the proportion of reference words found in the system's extractions. To improve token-level scorers, CaRB (Bhardwaj et al., 2019) computes precision and recall pairwise by creating an all-pair matching table, with each column as an extracted triplet and each row as a gold triplet.

Fact-level scorers, such as Sun et al. (2018), measure to what extent gold triplets and extracted triplets imply the same facts and then calculate precision and recall. Sun et al. (2018) propose a similarity measure between the predicted fact and the ground truth fact. Sun et al. (2018) defined two quantitative conditions for comparing facts. if one of the conditions is met, the fats are labeled as equal. The differences of the token-level and fact-level approaches are highlighted in Table 2.2 and in Table 2.3. The Table 2.2 shows a weakness of token-level extraction: If the arguments are shuffled token-level labels them as different, while fact-level evaluation compares the underlying facts and scores the evaluation as ideal. The weakness of fact-level extraction is shown in Table 2.3. Fact level scorer compares the extracted tuple to the ground truth, however, there is no direct match with any of the ground truth tuples, hence a fact-level scorer does not recognize the extraction as valid. While toke-level evaluation awards the extraction

Sentence	I ate an apple.	(recall,precision)	(recall,precision)
Ground truth	(I; ate; an apple)	Toke-level	Fact-level
Extraction 1	(I; ate; an apple)	(1,1)	(1,1)
Extraction 2	(ate; an apple, I)	(0,0)	(1,1)

Table 2.2: Token match versus Fact match on a simple shuffled extraction (Bhardwaj et al., 2019),(Sun et al., 2018)

Sentence	I ate an apple and an orange	(recall,precision)	(recall,precision)
Ground truth	(I; ate; ane apple), (I; ate; ane orange)	Toke-level	Fact-level
Extraction 1	(I; ate; an apple and an orange)	(1,0.57)	(0,0)
Extraction 2	(I; ate; an apple)	(0.87,1)	(1,1)

Table 2.3: Token match versus Fact match for multiple golden triplets (Bhardwaj et al., 2019), (Sun et al., 2018)

2.4 Used Framework

To develop the set of rules the NewPotato[1] framework is used. NewPotato is an extension of the POTATO framework (Kovács et al., 2022). POTATO is a language-independent framework for human-in-the-loop (HITL) learning of rule-based text classifiers using graph-based features, similar to HEIDL (Sen et al., 2019) and GrASP (Lertvittayakumjorn et al., 2022). NewPotato is aimed at implementing Open Information Extraction (OIE) principles with a Human-in-the-Loop (HITL) approach. The project consists of a backend API built with FastAPI, and a frontend built with Streamlit. The package, named newpotato, contains the core extraction logic of the project. NewPotato includes an experimental OIE extension, which enables rule creation by annotation and information extraction by the created rules. NewPotato is based on the idea, that humans can annotate a text by creating triplets of form: predicate(argument1, argument2). It is not binned to binary relations and can handle N-ary relations. The number of arguments can vary from 0 to many. If the triplet has zero arguments, the annotation consists of only one predicate. Depending on the matching algorithm this might be an equivalent of a word search. Based on the annotation and the UD parsed representation of the annotated graph, a triplet is created in a form similar to the one described in section 2.2. The created rule contains a pattern over the Universal Dependency graph representation of the parsed sentence.

Information is extracted from an input sentence if a single pattern(, i.e. a graph), matches the input graph(, i.e. the parsed graph representation of the sentence). Match is found if and only if the pattern graph is a subgraph of the input graph. The nodes and the edges of the graph may have string labels describing the word, the part of speech (POS), and the type of the UD relation. If a pattern graph contains regex labels, then the pattern matches an input graph if and only if the pattern graph is contained in the input graph such that the regex labels of the pattern graph match the string labels of corresponding nodes and edges in the input graph.

NewPotato includes also a custom-modified parser built on top of Stanza pipeline (Qi et al., 2020). This modification allows some adaptations. An especially useful feature is a list of abbreviations. This enables to define abbreviations, such as mentioned in section 3.2, along which the sentence does not need to be segmented. This solves the issue

[1]The repository is available under `https://github.com/adaamko/newpotato`

encountered initially, when the text got segmented not just along punctuation marks, such as period, question mark, and exclamation point, but also along the abbreviations.

CHAPTER 3

The AI legal tutor case

3.1 Business case

A hands-on business case was presented by a publisher of legal textbooks. The business case is based on a legal exercise: A legal case and a corresponding question are presented to students. The scholars are supposed to answer the question by analyzing the legal case. The ultimate goal of the business case is the creation of an Artificial Intelligence (AI) backed feedback system for law students, that provides them insights into their solution. The feedback is supposed to describe whether the solution is complete and if not, then what is missing. Therefore, the feedback system needs to be transparent and easy to follow.

The way legal texts are drafted is complex. It is crucial to pay particular attention to the meanings of words, abbreviations, and potential synonyms, especially those in Latin. For us to know what is important and what information needs to be extracted a detailed guideline is provided for 2 out of the 10 legal cases. These detailed guidelines list all the relevant information the students need to include in their results. This list of information is also the focus of the work as the feedback system is supposed to evaluate the occurrence of these pieces of information in the students' attempts. If a student makes an expected statement, the system should notice it, regardless of the wording. However, if an important piece of information is missing from a student's response, the system should also detect the absence of information and indicate this accordingly.

Jedoch	liegt	kein	gültiger	Titel	vor.
However	is	no	valid	title	there

Table 3.1: The German sentence *Jedoch liegt kein gültiger Titel vor.* and its corresponding English translation *However, there is no valid title.*

3.2 Data

For the business case, data was provided including in total 10 different legal cases. For each legal case, a sample solution written by domain experts and eight solutions written by law students are provided. The sample solutions written by domain experts incorporate all the expected elements for the corresponding legal cases.

For two of the ten legal cases, an additional detailed guideline is also provided. This guideline contains a detailed description of the use case it was created for. The information is color-coded into three categories:

- A color marks the theoretical knowledge, the legal base. An example for the legal base is, that the student mentions, that a legal title and modus exist for case 1.
- The second category marks the application of the theory knowledge to the specific case. This must show, why the student thinks, that the previously mentioned legal base is relevant to the case. It is essentially a reason for the existence of the legal base. An application to the case of the previously mentioned legal base is, that the student states, that the legal title is given in the form of a purchase contract, while the modus is given due to the handover in case 1.
- The last category describes, some relevant additional information. For case 1, such information is the following statement: A bona fide acquisition of ownership must be examined to answer the question. This last category is generally a summary or overview of the legal case.

A point is rewarded if the legal base and its application to the legal case are correctly described. The additional information is rather recommended to mention but no point is rewarded for the mentioning. However, if the requested additional information is missing, then the feedback system needs to warn the student to include it in the solution. The guideline states, that it is enough to find, if a statement, such as *"legal title and modus exist"* occurs in the text and the system needs to be able to differentiate if the statement is affirmed or denied by the student. If a legal base has multiple applications on the case, it is enough to state one, and not all need to be mentioned by the student. As an example from case 1: if *a legal transaction for consideration exists*, then there are 3 ways to underline this theoretical knowledge by a case-specific application: 1) *A purchase agreement has been concluded*; 2) *A purchase price has been paid* or 3) *100 euros have been paid*. A student is required to state at least one of these applications and then the point is awarded. For some cases, an additional note is also provided, such as, that the terms honesty (in German: *Redlichkeit*) and good faith (*Gutgläubigkeit*) are synonyms. However, listing synonyms is not all-embracing, Latin phrases and paragraph references are not explained. Lastly, the guideline also shows an example evaluation, by correcting some student attempts by marking the required relevant information statements in the attempt and listing how many points the student scored and what statements may be left out.

The guideline lists the exact pieces of information that need to be mentioned in the students' answers, i.e., the information that should be extracted by the set of rules if they occur in the text. We only know what statements need to be extracted thanks to the detailed guidelines. For now the guideline is available only for two cases however, for the business case, we can assume, that a detailed guideline will be provided for each legal case. But for now, the thesis is limited to only these two legal cases with the available guidelines. The detailed guidelines contain the example evaluation for 6 attempts from case 2. Based on these guidelines the recall of the extraction can be calculated as described in section 4.2.

In an ideal scenario, all 10 cases would be involved in the evaluation, however, I would argue, that these 2 cases are sufficient for the problem statement. The two factors limiting generalization are the different vocabulary and the diverse formulation and phrasing of the language. The two cases examined consist of several student attempts. A total of 18 texts are reviewed. The large number of documents guarantees a diversity of phrasing and formulations in the texts discussed. Therefore, one limiting factor can be sufficiently tested across the 18 texts. Case 1 and case 2 do not share the same vocabulary, therefore the second limiting factor of the generalization can be inspected also by using only these 2 cases. Using more legal cases would be beneficial as the amount of data available for testing would increase, but the generalization issues among cases already occur between 2 cases, hence 2 cases are enough to investigate the scientific problem.

Before using the written text, it needs to be analyzed. The text frequently contains semicolons, which disturbed the sentence segmentation, therefore these had to be replaced by commas. Similarly to semicolons, the legal text also contains abbreviations, such as *gem., Abs.,zB.*, which might issues with the segmentation. Thanks to the custom-modified parser used, this did not pose a significant challenge. Additionally, some of the students' attempts are not purely continuous text but occasionally include bullet-point listings, such as attempt 3 of case 1: *Entgeltlich: Kaufvertrag (100€)* or *Bewegliche Sache: Fahrrad*. Furthermore, additional notes in parenthesis, such as *Kaufvertrag (100€)*, are also common.

Rule-based OIE works also on bullet-point listings if the pattern is defined accordingly: In case of *Entgeltlich: Kaufvertrag (100€)*, the pattern graph should contain the UD relation appositional modifier, abbreviated as *APPOS*, because the parser recognizes this formulation as a link between key-value pairs and marks the relation as *APPOS*, shown in Figure 3.1. Such a working pattern is Figure 3.2. This specific extraction pattern is capable of extracting the relation of the targeted bullet-point listing involving the *APPOS* relation. Without creating this ad-hoc extraction pattern, the information would not be extracted from a bullet-point-like listing. This highlights the need to be aware of the structure of the text hence, why one needs to thoroughly analyze the text before starting to create the rule system.

Legal texts might incorporate references to paragraphs, abbreviations, and occasionally Latin phrases. Luckily, this domain-specific characteristic of the corpus seemed to be less of an issue during the testing, as these segments are less likely to be marked as important

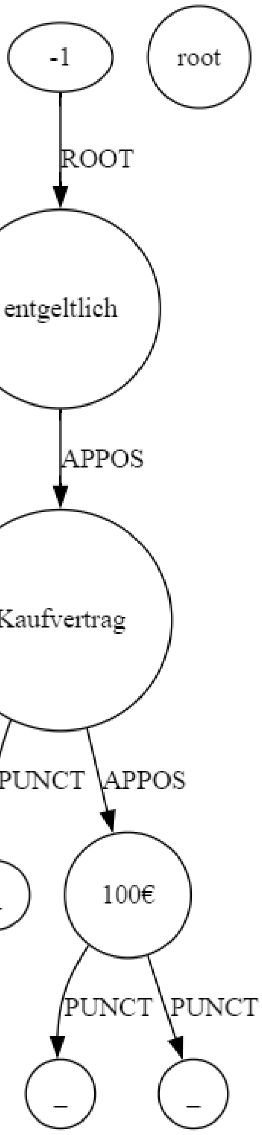


Figure 3.1: UD parsed tree representation of the German sentence: *Entgeltlich: Kaufvertrag (100€)*

by the domain experts. Nevertheless, some related issues were encountered, which will be discussed later.

3.3 Use case

The feedback system described in section 3.1 involves two modules: First, it needs to be based on an information extraction algorithm. That information extraction algorithm is

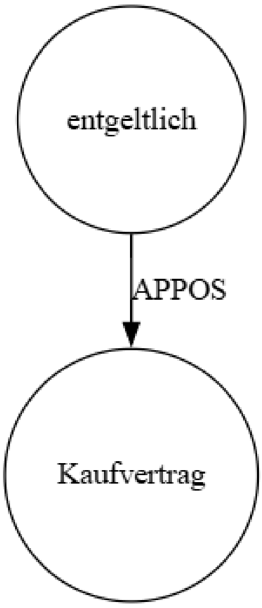


Figure 3.2: UD parsed tree representation of the matching pattern: *Entgeltlich(Kaufvertrag)*

supposed to discover and extract the pieces of information from the students' attempts. Second, based on the information extraction feedback needs to be provided for the students, which requires a corresponding interface and logical system, that evaluates the extracted elements and provides corresponding feedback. The focus of my thesis is the information extraction module of the use case. The requirements of the business case determine the following aspects from a technical perspective:

To design and test the information extraction system, one needs to know, what information needs to be extracted. This is discussed in the detailed evaluation guideline, which is provided for two legal cases each including 8 students' attempts and a sample solution. The limited amount of data restricts the range of feasible approaches, i.e. no DL could be trained based on the available data. Furthermore, there is no precise annotation available, forming a structured ground truth for the extraction. The domain experts corrected some attempts of case 1 and case 2. The correction is rather to indicate what is relevant information and how the feedback system needs to work. Therefore it is not directly usable as a triplet annotation for the rule-based open information system.

Due to the transparency requirement of the feedback systems, an explainable information extraction needs to be designed. Rule-based Open Information Extraction fulfills this criteria and is suitable. Additionally, rule-based OIE is also applicable even on a limited amount of data, making this extraction method a good match for the business case.

However, the potential differences in phrasing of the attempts pose a challenge: An example for the different wording is the sentence: *"Ein entgeltliches Rechtsgeschäft liegt* *vor."* (in English: A legal transaction against payment exists.) This statement is one of 8 elements the students should mention in their answers for case 1. The challenge of information extraction is, that this information can be expressed in multiple ways: *"Somit ist der Erwerb entgeltlich."* (in English: The acquisition is therefore against payment.) or *"Das Fahrrad wurde entgeltlich erworben."* (in English: The bicycle was purchased for a fee.). These sentences contain the same core information, that the deal was in return for payment. However, they are formulated differently, which poses the main challenge for the rule-based OIE as the set of rules should be defined in a way, that they can extract the same information despite the different formulations.

This poses a generalization challenge on the attempt level. However, the feedback system should work not just for one case with its numerous student attempts, but also for multiple legal cases. This requires a higher level of generalization. The advantage of rule-based OIE is, that if the rule definition is simple and enables post hoc modification, then the domain expert could adjust the rules for each case enabling wider generalization.

Therefore this use case is perfectly usable for the defined problem statement, namely what set of rules performs well on German legal cases and how a set of rules generalizes across legal cases.

CHAPTER 4

Methodology

4.1 Node matching algorithms

The essence of rule-based OIE is the matching of patterns to target sentences. The matching works as described in section 2.4. As part of the extraction, the pattern is compared to the UD parsed representation of a sentence. The extraction is made, if the subgraph, i.e. the extraction pattern, matches the parsed graph of the sentence. If the nodes have string or regex labels a node matching is done. In this section the node matching is discussed as the choice of the node matching algorithm is crucial for the generalization capability of the extraction model. Nodes represent the actual words of the pattern and the edges between them are the UD relations as described in section 2.2.

Es	gehört	Fanny,	da	sie	es	gutgläubig	erworben	hat.
It	belongs	Fanny,	as	she	it	in good faith	achquired	has.

Table 4.1: The German sentence *Es gehört Fanny, da sie es gutgläubig erworben hat.* and its corresponding English translation *It belongs to Fanny, as she acquired it in good faith.*. Its UD parsed tree representation is shown in Figure 2.2

Overall, three different node-matching algorithms are tested in combination with the defined rule sets. The three matching algorithms are the following:

- Lemma-based matching of the words of the patterns.
- Lemma-based matching for the predicate and POS-based matching for the arguments. Later referred to as Lemma-POS-based matching
- POS-based matching both for the predicate and the arguments.

Sentence	Lemma-based	Lemma-POS-based	POS-based
Es gehört Fanny, da sie es gutgläubig erworben hat.	gutgläubig(erworben)	gutgläubig(erworben)	gutgläubig(erworben)
Es gehört Fanny, da sie es gutgläubig gekauft hat.	NO extraction	gutgläubig(gekauft)	gutgläubig(gekauft)
Es gehört Fanny, da sie es redlich erworben hat.	NO extraction	NO extraction	redlich(erworben)
Es gehört Fanny, da sie es redlich gekauft hat.	NO extraction	NO extraction	redlich(gekauft)

Table 4.2: Extracted triplets from 4 example sentences using the different matching algorithms. The UD parsed representation of the sentences equal to Figure 2.2 with the only difference, that the adjective *gutgläubig* and the verb *erworben* are changed for some of the example sentences. The extraction pattern used is shown in Figure 2.3.

For showcasing, how each algorithm works the sentence discussed in section 2.1 is used. The sentence is shown again in Table 4.1. The parsed representation of the sentence is shown in Figure 2.2 and the matching pattern is Figure 2.3. If the annotated triplet is *gutgläubig(erworben)*, then *gutgläubig* is the predicate and *erworben* is the only argument.

Lemma-based matching requires, that the matched words need to have the same lemma. POS-based matching is more general, it requires only the same POS for the matched words. How the extraction works for different matching algorithms is shown in Table 4.2. Using only lemma-based matching is quite specific, it only extracts the triplet if it exactly matches the matching pattern and its words. Lemma-POS-based matching is more general. For argument matching it requires POS-based matching, therefore it extracts the triplet even from the second example sentence shown in Table 4.2. However, for the 3rd and 4th sentences, the word *gutgläubig* is changed, which is the predicate of the pattern triplet, therefore lemma-POS-based matching does not extract any triplet from the last two sentences. POS-based matching only requires matching POS for the words of the patterns. Therefore POS-based matching extracts a triple from all the sentences.

For lemma-based matching, one needs to be cautious and should consider which words are used in a pattern, especially in the German language. A German language-specific issue is, that the nouns might be slightly different depending on the gender. A good example of gender-specific nouns are the German words for owner: *Eigentümerin* and *Eigentümer*. Depending on the gender of the owner either the male version (*Eigentümer*) or the female version (*Eigentümerin*) is used. This might cause an issue for lemma-based matching if the used parser does not find the same lemma for the two forms of the noun.

Let us consider the sentence from the sample solution for case 1: *Paula gibt sich als Eigentümerin aus.* One can annotate the triplet: Eigentümerin(Paula). For now let us ignore the underlying pattern, which might be sentence-specific, and focus only on the node matching, i.e. the word matching. Due the annotated triplet one can extract the information from a sentence like *Fanny gibt sich als Eigentümerin aus.*, with the assumption, that the conditions for graph matching are met. However, what happens if the owner is a male: In German an example text may be as follows: *Paul gibt sich als Eigentümer aus.* In this case, the predicate becomes *Eigentümer* instead of *Eigentümerin*. This seems to be a marginal difference, but the used parser does not track back these two forms of the noun to the same lemma. The parser differentiates between the lemma *Eigentümer* and *Eigentümerin*. Accordingly, even if the subgraphs match, the triplet is not found due to the lemma-based node matching. Nevertheless, this issue could be overcome by a synonym list or an ad hoc rule.

4.2 Metrics

The evaluation of Open Information Extraction is based on comparing the extracted triplets to a ground truth, i.e. golden triplets. How this comparison is done is not trivial. It can be both on token- and fact-level as described in section 2.3.

A detailed guideline was provided for the first two legal cases how to evaluate the students' attempts. This guideline is critical because it defines which pieces of information are important and need to be extracted for each legal case. During evaluation, this detailed guideline is also used to decide which extracted triplet contains relevant information and should be considered a true positive. For each tested extraction precision and recall were calculated. These metrics are calculated on the attempts of the cases as these are the target texts the extraction needs to work on.

The ground truth for a case is defined based on the sample solution and the guidelines provided for the case. For case 1 the sample solution looks as follows:

Fanny erwirbt das Fahrrad von jemandem, der nicht darüber verfügungsberechtigt ist, ein ***derivativer Erwerb*** *des Eigentums* ***scheidet*** *damit* ***aus****.* ***Zu prüfen*** *ist daher* ***ein gutgläubiger Erwerb*** *vom Nichtberechtigten (§ 367 f ABGB). Für den Eigentumserwerb der Fanny liegen* ***Titel („verkauft", Kaufvertrag)*** *und* ***Modus („übergibt")*** *vor. Das* ***Fahrrad*** *ist eine* ***bewegliche Sache****. Hier* ***liegt*** *ein* ***entgeltliches Rechtsgeschäft vor*** *(„****verkauft****").* ***Fanny*** *ist* ***redlich*** *(angemessener Preis,* ***Paula*** *gibt sich als* ***Eigentümerin*** *aus, keine gegenteiligen Angaben im Sachverhalt).* ***Paula*** *ist als* ***Pfandgläubigerin Vertrauensmann*** *des Eigentümers Erwin. Damit liegen alle Voraussetzungen vor und Fanny* ***erwirbt originär Eigentum*** *am Fahrrad.*

The relevant information segments according to the evaluation guideline are highlighted by bold format. The triplets created based on the highlighted parts form the ground truth. These golden triplets are the following:

- prüfen(gutgläubiger_Erwerb)
- scheidet_aus(derivativer_Erwerb)

- Titel(verkauft)
- Titel(Kaufvertrag)
- Modus(übergibt)
- bewegliche(Sache, Fahrrad)
- entgeltliches(Rechtsgeschäft)
- Rechtsgeschäft(verkauft)
- redlich(Fanny)
- Eigentümerin(Paula)
- Vertrauensmann(Paula)
- Pfandgläubigerin(Paula)
- originär(Eigentum, erwirbt)

The extracted triplets are evaluated against these triplets by comparing the information they carry, hence on fact-level. Some formulations found in the students' attempts deviate from the phrasing of the sample solution. For example, the evaluation guideline states, that there might be some synonyms words as mentioned in section 3.2. Consequently, some attempts, such as the 3rd attempt of case 1, use the synonym phrase *gutgläubig* instead of *redlich*, that was initially used in the sample solution. Therefore, the evaluation can not rely only on these golden triplets but also needs to consider the evaluation guidelines provided by the domain experts. Due to this business case requirements, the calculation of the metrics is the following:

Recall is calculated as the relation of the point rewarded based on the extracted triplets to the points rewarded based on reading the actual text. Therefore, triplets are evaluated based on the information they represent. The rule set was created for case 1 following the detailed description written by the domain expert. Each annotated triplet covers a piece of information relevant for awarding a point. Consequently, the previously described logic of recall equals to the relation of found triplets to the expected (,i.e. annotated,) triplets for case 1. The evaluation is made on all 8 attempts of case 1, as the annotation was made on all attempts of the case. For case 2, we rely purely on the detailed guidelines as no annotation was created for the attempts. Domain experts only evaluate 6 attempts for case 2, therefore the evaluation is done only based on these 6 attempts. In total 13 points are rewarded for these 6 attempts. It is examined how many points can be rewarded based on the extracted triplets and this is compared to the 13 points awarded by the domain experts.

The precision is calculated as the ratio of useful triplets to all extracted triplets. The question of usefulness is judged based on the expert's guidelines. The decision is made

Fanny	ist	redlich.
Fanny	is	honest.

Table 4.3: The German sentence: *Fanny ist redlich.* and its corresponding English translation: *Fanny is honest.*

purely on triplet-level: A triplet is considered useful, i.e. true positive, if it contains information relevant to at least one of the expected statements listed in the evaluation guidelines.

Compared to this implemented logic for precision, the evaluation from the business case's point of view deviates slightly. The false positive extractions for the business case are only those, that lead to false evaluation of the attempt. All the meaningless extractions are irrelevant to the business case as they do not influence the scoring of the students' attempts, therefore they are not labeled as false positives. Consequently, from the business case's perspective, the real number of false positive extractions is smaller, i.e. the precision is better, compared to the evaluation logic used in my thesis. Attempt 1 of case 1 contains the sentence shown in Table 3.1. If the triplet liegt_vor(Jedoch) is extracted from this sentence, this triplet is a noise in the results but does not influence the scoring of the attempt. Therefore strictly speaking it does not count as a false positive extraction. The crucial false positives are only the extractions, based upon a point is awarded, while no point should have been granted. If the extraction on the same sentence finds the triplet liegt_vor(gültiger,Titel), but not the denial and a point is awarded based on the extracted triplet, then this triplet is considered a false positive for the business case. The reason is, that the statement is wrongly denied in the sentence, i.e. no points should be awarded but the extraction gives the impression, that the statement is correctly affirmed, hence the point is awarded.

4.3 Workflow

The NewPotato framework is used for rule creation and extraction. NewPotato is initially based on the Lemma-POS matching: This means, that for the predicate lemma-based matching is used and the arguments are found by POS-based matching. If there is a sentence like shown in Table 4.3, then a triplet like redlich(Fanny) extracts also redlich(Peter) from the sentence *Peter ist redlich.* The reason is, that the predicates have the same lemma, and for arguments only the POS is compared: Fanny and Peter are different names, but they both have the same POS: proper noun (PROPN).

The first step is to experiment with the NewPotato matching algorithm and create rules to see how they generalize initially among the multiple attempts of case 1. The aim is to test how the matching works, how complex triplets should be, and what structure a triplet should have. Complexity relates to the number of words that should be included in the triplet. This impacts the pattern created based on the triplet annotation. The structure is a follow-up consideration. If it is decided that the triplet consists of three words, then structure determines which word is the predicate and which are the arguments.

The complexity of a triplet is essential for the long-term goals of the business case because the points are awarded if a legal base and its application to the legal case occur in the students' attempt, thus the criteria for a point might be complex. Consequently, one could create either one complex triplet covering the statement and the argument for the point or multiple simpler ones by splitting the legal base and the argument. One complex triplet is better as it is enough to check the occurrence of one triplet for a point, but it might be disadvantageous for generalization. Multiple simpler triplets may generalize better, but to award a point, the occurrence of multiple triplets needs to be checked.

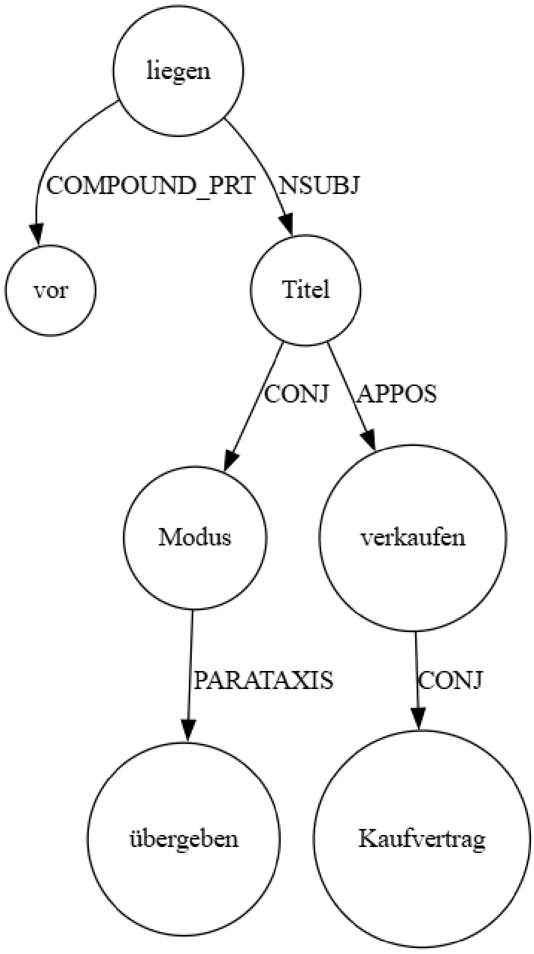
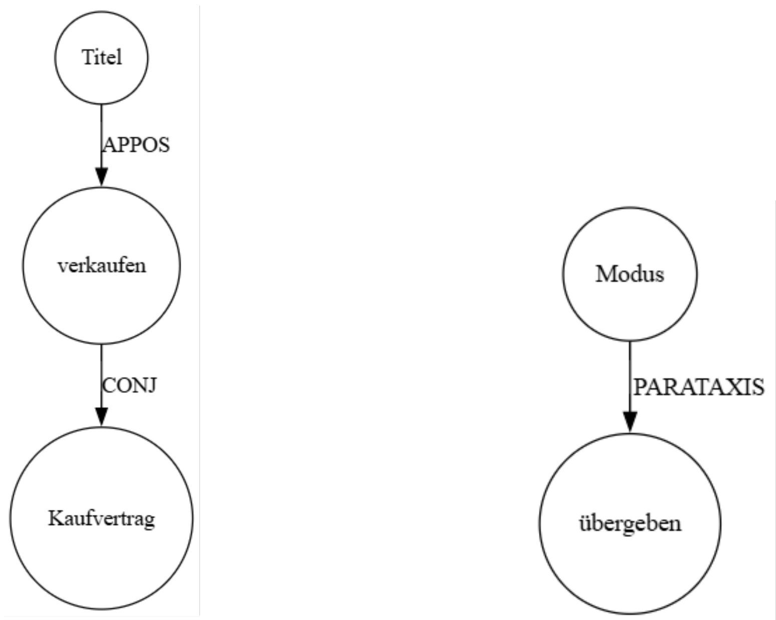


Figure 4.1: The matching pattern for the triplet annotation *liegt_vor(Titel,Kaufvertrag,Modus,übergibt)* from the sentence *Für den Eigentumserwerb der Fanny liegen **Titel („verkauft“, Kaufvertrag)** und **Modus („übergibt“)** vor.*

One can take a sentence from the previously quoted sample solution of case 1: *Für den*

Eigentumserwerb der Fanny liegen Titel („verkauft“, Kaufvertrag) und Modus („übergibt“) vor. Two separate annotations were created on this sentence for the ground-truth: *Titel(Kaufvertrag)* and *Modus(übergibt)*. In order to award a point, it is necessary that both triplets are extracted. Therefore, it may be beneficial to create one triplet out of the two, especially since both originate from the same sentence. It would be also useful to include *liegt vor* to underline the affirmation of the statement. Therefore one could create a triplet like *liegt_vor(Titel,Kaufvertrag,Modus,übergibt)*. This complex triplet would cover the statement, the argument and even an affirmation. Therefore, only this triplet could make the decision about scoring. However, let us compare the extraction patterns for this complex triplet. The complex annotation results in a complex extraction pattern shown in Figure 4.1. This pattern structure involves in total 7 nodes, making it extensive and highly sentence-specific. To the contrary this complex pattern could be split up into two simpler patterns *Titel(Kaufvertrag)* (shown in Figure 4.2a) and *Modus(übergibt)* (shown in Figure 4.2b). Splitting the complex triplet into two simpler annotations and ignoring the affirmation lead to a less sentence- and phrasing-specific extraction patterns. Having simpler triplets might lead to easier extraction, however for the evaluation of the scoring, the occurrence of both triplets need to be examined.



(a) Pattern for the triplet annotation *Titel(Kaufvertrag)*

(b) Pattern for the triplet annotation *Modus(übergibt)*

Figure 4.2: Two matching patterns from the sentence *Für den Eigentumserwerb der Fanny liegen* ***Titel („verkauft“, Kaufvertrag)*** *und* ***Modus („übergibt“)*** *vor.*

An other aspect of triplet complexity relates to the generalization capability. For case 1 one point is rewarded if the student mentions, that the case is about a paid transaction and underlines the statement with a reasoning. In total 3 different reasons were mentioned

Das	Fahrrad	ist	eine	bewegliche	Sache.
The	bike	is	a	movable	object.

Table 4.4: The German sentence: *Das Fahrrad ist eine bewegliche Sache* and its corresponding English translation: *Bike is a movable object.*

by the domain expert: a purchase price was paid or 100EUR was paid or a purchase agreement was concluded. For the point both the statement and one of the reasons need to be mentioned. One can intuitively already see that combining the statement and the reasoning in one triplet is not beneficial, as the reasons can vary. A student might mention 100EUR but the other may refer to a paid purchase price instead. Therefore, this statement is better covered by multiple simpler triplets instead of one complex. Another less intuitive example is the sentence from the sample solution for case 1 shown in gloss 4.4. The student is expected to mention, that the bike is a movable object. The initial thought is to create triplet annotation like Fahrrad(bewegliche,Sache). However, after analyzing the student results, this seems to be the wrong way. The findings are shown in subsection 5.1.1.

Once the complexity of a triplet is found, it is also checked, how a triplet needs to be structured. Let us take the previous example with the bike and assume it was found, that the ideal complexity includes all three words in one triplet. The question is which triplet works best across the 8 student attempts: Fahrrad(beweglich,Sache) or bewegliche(Sache, Fahrrad) or eventually Sache(bewegliche, Fahrrad). The question of the ideal triplet structure depends mainly on what information needs to be extracted and what node-matching algorithm is used. The goal is to find a structure, that is beneficial for generalization. Therefore, the sentences are taken from the students' attempts, which contain the information of interest. It is compared, how the same information was formulated in each sentence and how the parsing trees of the sentences look like to find the optimal structure of the triplet.

The second step is to create a rule set that works perfectly for case 1. The rules created based on the annotation are supposed to cover all relevant pieces of information in each attempt, reaching a recall of 1 and possibly a precision of 1. This can be achieved if all attempts of the case are annotated one by one. This might lead to a situation, when the number of rules is high, including even some duplicates. But the goal is to reach a recall of 1, even if it means that the rules become highly case- and sentence-specific, resulting in low generalization capabilities. To validate the initial set, extraction is done on case 1, first using a lemma-based matching. Then recall and precision are both calculated with the logic described in section 4.2.

Once it is ensured, that the initial set of rules works as intended on case 1. Then the same set of rules is applied to the new case, to the 6 attempts of case 2. These 6 attempts are not yet examined so far, no rule was created for them, but thanks to the domain expert's evaluation guidelines, it is precisely known, which information needs to be extracted from these 6 texts. This makes them ideal to test the generalization capabilities. Recall and precision are calculated similarly as before.

The following steps aim to generalize the extraction step by step. This can be done by generalizing both the rule set and the matching algorithm. The first generalization step is to use a more general matching algorithm, lemma-POS-based matching, with still the same initial set of rules. The second generalization step is to extend the initial set of rules created for case 1: to incorporate case 2 relevant vocabulary for lemma-based and lemma-POS-based matching, the sample solution of case 2 is annotated, i.e. a set of rules is created for it. The initial rule set is combined with the rules created on the sample solution of case 2. The combined set of rules is tested on the attempts of both legal cases first using the initial lemma-based and then the lemma-POS-based matching. The third generalization step is to use an even more general matching algorithm, namely the POS-based matching. Knowing, that the combined rule set is an extended version of the initial rule set, only this combined rule set is used for the POS-based matching.

In total 6 scenarios are tested. A scenario is determined by the combination of the set of rules and the matching algorithm (described in section 4.1). Each scenario is applied on the attempts of case 1 and case 2. For each scenario, recall and precision are calculated and the number of extracted triplets is noted. The scenarios are the following:

- Scenario 1: Lemma-based matching with the initial rule set created for case 1. Later referred to as *lemma_initial*
- Scenario 2: Lemma-POS-based matching with the initial rule set. Later referred to as *lemma_POS_initial*
- Scenario 3: Lemma-based matching with the combination of the combined data, consisting of the initial set of rules and the rules created on the sample solution of case 2. Later referred to as *lemma_combined*
- Scenario 4: Lemma-POS-based matching with the combination of the combined data, consisting of the initial set of rules and the rules created on the sample solution of case2. Later referred to as *lemma_POS_combined*
- Scenario 5: POS-based matching with the combination of the combined data, consisting of the initial set of rules and the rules created on the sample solution of case 2. Later referred to as *POS_combined*

CHAPTER 5

Results and Discussion

5.1 Results

First, the overall results are presented in Table 5.6. It must be noted, that the metrics are calculated on students' attempts, with the constraint, that for case 2 only 6 attempts are analyzed as expert evaluation is only provided for those for case 2. The strength of rule-based OIE is, that high recall can be achieved if the rules and the matching algorithm are defined correctly. In our case, the rules were defined primarily on case 1. As a result, a recall of 1 is achieved in case 1, in addition to a precision of 1 in scenarios lemma_initial and lemma_combined, when lemma-based matching is used.

The recall on previously not-seen data is a good indication of generalization. Here case 2 attempts represent this formerly not-seen data. The recall for case 2 varies significantly across the scenarios. Precision is also a significant indicator of the quality of the data extraction. Having a low precision and a high number of extracted triplets despite a high recall creates noise in the result leading to worse usability.

In the following, each examined setup will be discussed in detail.

5.1.1 Initial testing

The initial testing on the case level has the purpose of analyzing how an optimal triplet looks like. How many words does it include and which words are chosen as predicates. The question of triplet structure, i.e. what is the predicate, is especially significant when the Lemma-POS node matching algorithm is used because in that algorithm the predicate is matched based on lemma, but the arguments are matched on POS of the words.

The complexity of a triplet is examined first. The chosen words in a triplet determine the pattern upon which the extraction relies. These patterns are created from the parsed graph representations of the annotated sentence as described in section 2.2. To briefly recap: the shortest path between the words in a triplet creates a subgraph of the parsed sentence and that subgraph becomes the pattern for the annotation. This is why the complexity of the pattern only depends on the words the triplet annotation consists of. The structure of the triplet, i.e. what is the predicate and what are the arguments, does not influence the complexity of the extraction pattern. Consequently, patterns are subgraphs of the parsed sentences, therefore the parsing of a sentence is essential to understand the mechanism of rule-based OIE.

Hier	hat	sie	um	einen	angemessenen	Preis	(damit	ist
Here	has	she	for	a	reasonable	price	(thus	is
§ 368	Abs	2	ABGB	nicht	erfüllt)	und	somit	gegen
§ 368	Section	2	General Civil Code	not	fulfilled)	and	therefore	for
Entgelt	das	Fahrrad	gekauft	und	Eigentum	erworben.		
fee	the	bike	bought	and	ownership	acquired.		

Table 5.1: The German sentence: *Hier hat sie um einen angemessenen Preis (damit ist § 368 Abs 2 ABGB nicht erfüllt) und somit gegen Entgelt das Fahrrad gekauft und Eigentum erworben.*, and its corresponding English translation *In this case, she purchased the bicycle for a reasonable price (thus Section 368-2 of the General Civil Code is not fulfilled) and thus acquired ownership.*

Some sentences can be quite complex and the UD parsing might fail to correctly build the parsing tree. A good example of this issue is the following sentence from the 7th attempt in case 1 shown in Table 5.1. The corresponding parsing tree is shown in Figure 5.1. This sentence has a long additional note in the parentheses, which makes the parsed representation of the graph complex and less intuitive. The two verbs, *erfüllen* and *kaufen* are connected, even though they are logically independent. This was experienced not just for this one sentence. It seems, that extensive information, precisely if it includes verbs, written in parentheses tends to confuse sentence parsing. Consequently, a complex annotated triplet on this sentence, such as *gekauft(sie,Fahrrad, um_angemessenen_Preis)* results in the following highly specific syntactic matching pattern shown in Figure 5.2. Besides the words in the triplet, additional words, such as *haben* and *erfüllen* appear in the pattern due to the parsed structure of the sentence. This resulting pattern is highly specific to the formulation of the sentence and is disadvantageous for generalization.

There are some annotations, that seem to be simpler at first but the underlying subgraph is similarly sentence specific: In attempt 6 of case 1, there is the following sentence: *Es handelt sich bei dem Fahrrad um eine bewegliche Sache.* The UD parsed tree representation of the sentence is seen in Figure 5.3. The relevant information of this sentence is, that the *Fahrrad* (bike) is *bewegliche* (mobile). One could annotate a corresponding triplet: *beweglich(Fahrrad)*. However, due to the parsed graph representation of the sentence, the underlying subgraph of the annotation is more complex, than expected: Figure 5.5a. Between the two targeted words, the predicate (*beweglich*) and the argument (*Fahrrad*), there are 2 additional words in the pattern subgraph because the adjective (*beweglich*) is not referring to the target noun directly (see in Figure 5.3).

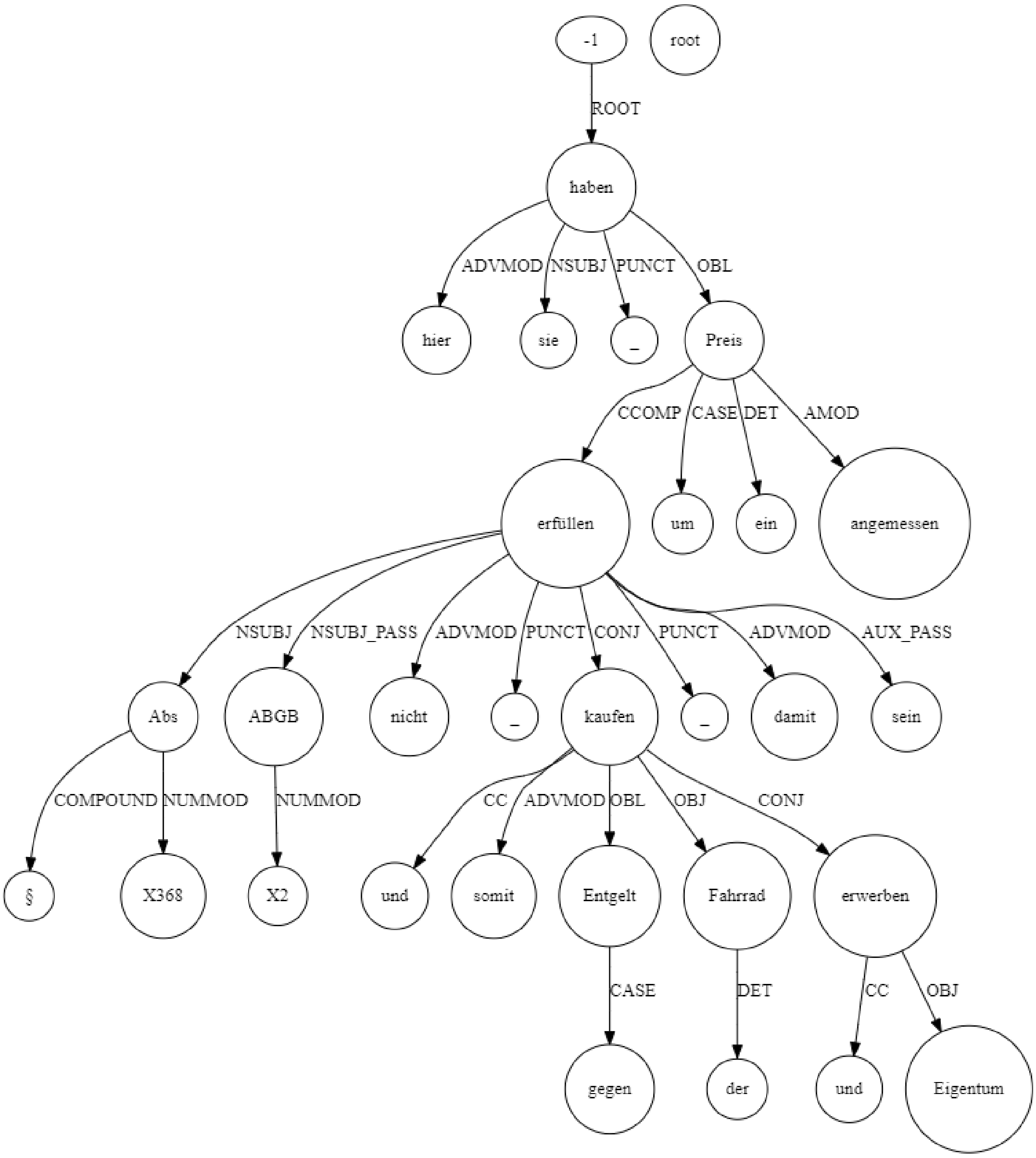


Figure 5.1: UD parsed tree representation of the German sentence shown in Table 5.1

Due to this phrase-specific complexity of the underlying subgraph of the annotation, the information of interest would not be extracted from a similar, but differently phrased sentence found in the sample solution of case 1: *Das Fahrrad ist eine bewegliche Sache.* Its parsed representation is shown in Figure 5.4

Therefore it is recommended to ideally split the annotation into two: *beweglich(Sache)* and *Sache(Fahrrad)*. This leads to two pattern subgraphs Figure 5.5b and Figure 5.5c. This

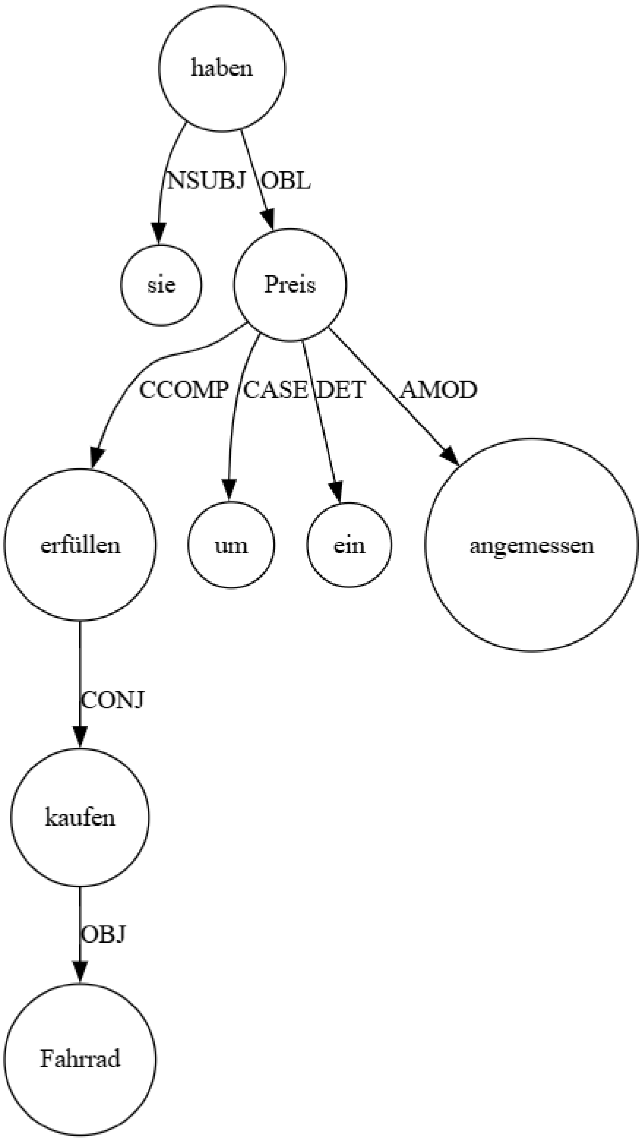


Figure 5.2: The matching pattern corresponding to the annotated triplet *gekauft(sie,Fahrrad, um_angemessenen_Preis)*, for sentence shown in Table 5.1

is beneficial for the generalization because the first pattern also appears in the alternative sentence (see in Figure 5.4), thus the triplet *beweglich(Sache)* is found. Although the second triplet, shown in Figure 5.5c, is still phrase specific, so the relation between *Sache* and *Fahrrad* is not found in the alternative sentence based only on the two patterns. Despite the fact that the relationship between the object and the bike is not missed, the information regarding the mobile object is correctly extracted.

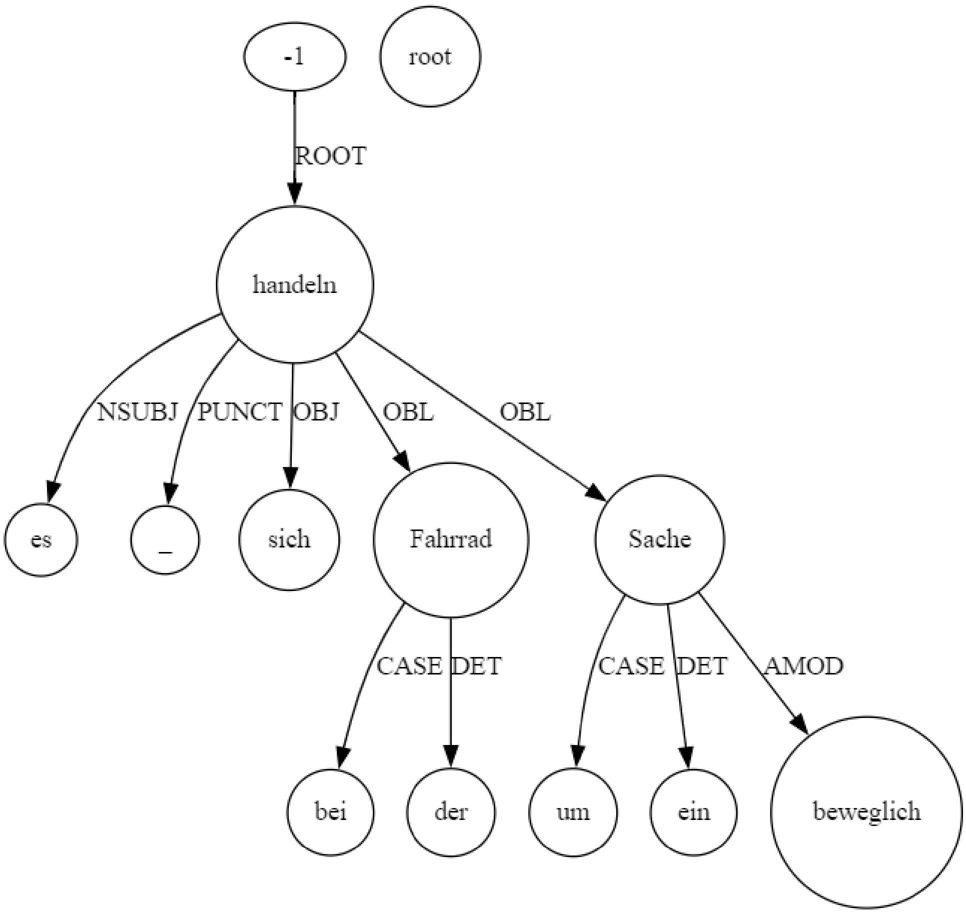


Figure 5.3: UD parsed tree representation of the German sentence: *Es handelt sich bei dem Fahrrad um eine bewegliche Sache.* A seemingly simple sentence containing the key information, that that the bike is a mobile object.

As the examples show, a simple extraction pattern is beneficial for the generalization capability. Therefore, an important conclusion of the initial step is that the annotation and most importantly the patterns need to be kept as simple as possible.

Once, the complexity of the extraction pattern is determined, the structure of the triplet needs to be tested. The structure of the triplet, i.e. the choice of the predicate and arguments, is relevant depending on how the nodes are matched. Therefore it is discussed along with the node-matching algorithms in the following sections.

5.1.2 Lemma-based matching

The most strict matching algorithm is lemma-based matching. This means, the words are matched based on their lemma as described in section 4.1. Pure lemma-based matching is

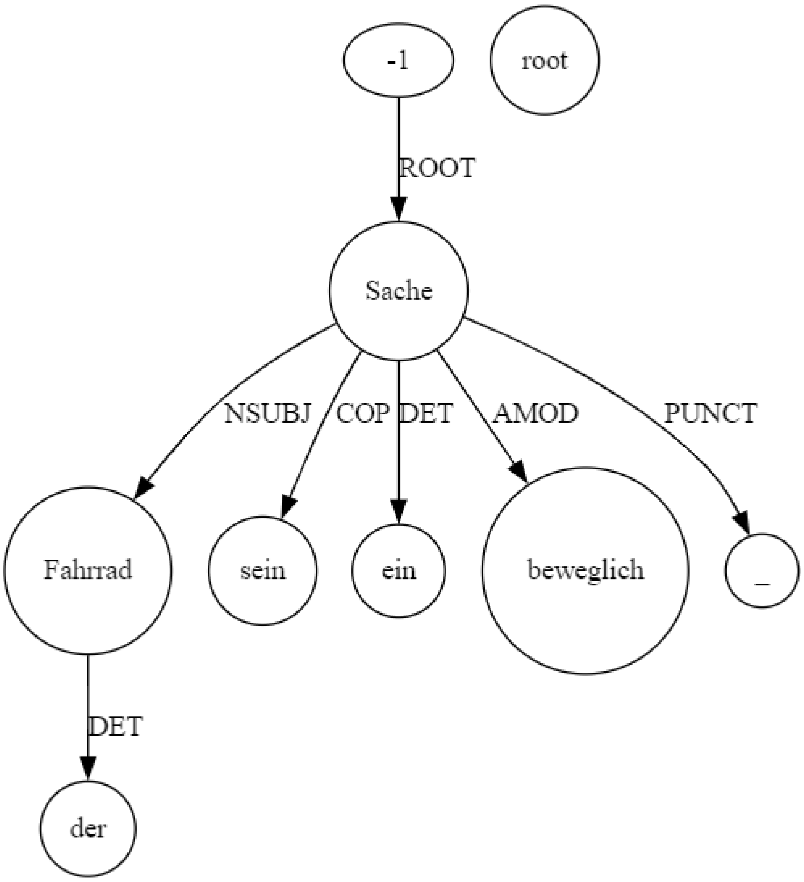
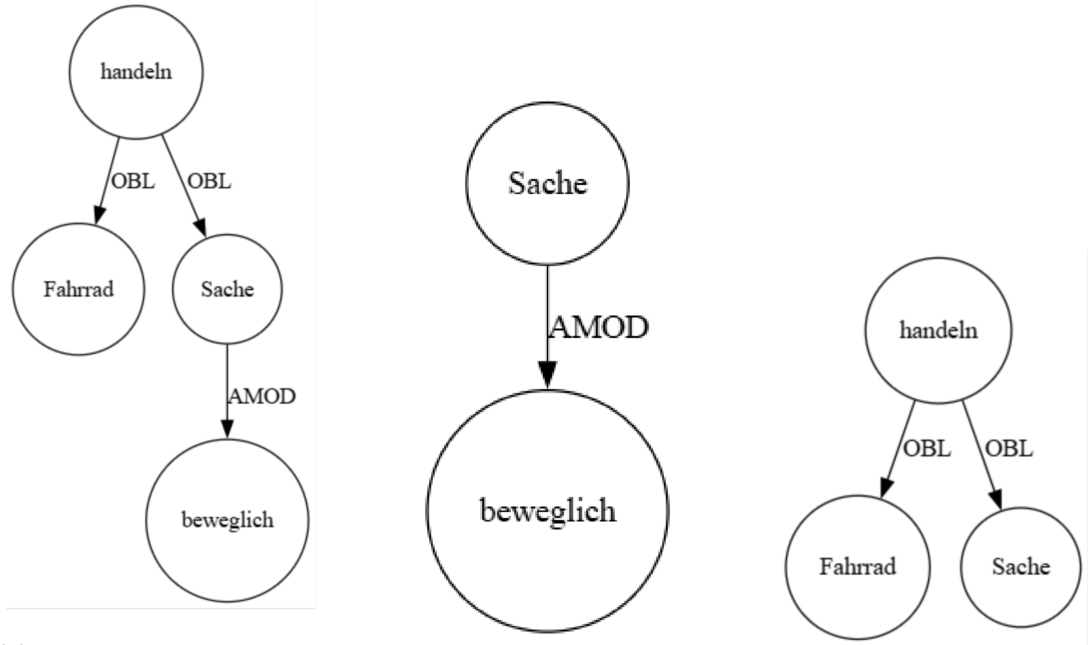


Figure 5.4: UD parsed tree representation of the German sentence: *Das Fahrrad ist eine bewegliche Sache.*

used in two scenarios, lemma_initial and lemma_combined. The results for the scenario are shown in Table 5.2.

Lemma_initial uses the initial set of rules. The initial ruleset was created based on the first case. This set of rules includes 79 pattern graphs created by manually annotating the case and the corresponding triplet indication. Consequently, the initial set of rules is expected to reach a recall and precision of 1 on case 1 in both scenarios.

It must be noted that not only the initially annotated triplets are extracted for case 1. In some cases, a rule created for one attempt extracts a triplet from another attempt. However, all these additionally extracted triplets are relevant: Let us consider the first attempt of this case: All the 11 initially annotated triplets are extracted thanks to the attempt-specific rules. Besides these initial 11 triplets, 4 additional triplets are found by rules defined for other attempts. These 4 additional triplets are the following: *Titel(gültiger)*, *gutgläubig(erworben, Rad)*, *gutgläubig(erworben, F)* and *Fahrrad(Sache)*.



(a) Pattern for the triplet annotation *beweglich(Sache,Fahrrad)*

(b) Pattern for the triplet annotation *beweglich(Sache)*

(c) Pattern for the triplet annotation *Sache(Fahrrad)*

Figure 5.5: Three matching patterns from the sentence *Es handelt sich bei dem Fahrrad um eine bewegliche Sache.* shown in Figure 5.3

Scenario	**Legal case**	**Recall**	**Precision**	**No. Extracted Triplets**
lemma_initial	1	1	1	86
lemma_initial	2	0	0	0
lemma_combined	1	1	1	86
lemma_combined	2	0.077	1	2

Table 5.2: Performance metrics by scenario and legal case using lemma-based matching algorithm: The metrics are calculated on the student attempts of the cases. Each scenario is determined by the combination of the ruleset and the matching algorithm. The scenarios are described in section 4.3

These triplets were initially not annotated for the first attempt but are still relevant pieces of information, hence they are considered true positive results and the precision is 1. Similarly, for other attempts of the case, a couple of not annotated triplets may get extracted. Due to the lemma-based matching, all these contain relevant information. These additional, not expected triplets are good examples: how one rule created for a specific sentence can result in initially not intended but useful extractions on other sentences. This kind of generalization is desired and worth investigating.

The above example triplets for the first attempt have one thing in common, all are relatively simple. The first contains only 2 words, a predicate, and an argument: *Titel(gültiger)*, the second is more complex having a second argument: *gutgläubig(erworben, Rad)*. These are extracted by the following subgraphs: *(u_5 / Titel :AMOD (u_4 / gültig))* and *(u_13 / erwerben :OBJ (u_11 / Rad) :ADVMOD (u_12 / gutgläubig))*. Hence even the more complex triplet: *gutgläubig(erworben, Rad)* is extracted by a simple subgraph of three words, which are exactly those forming the triplet. This finding further underlines the conclusion, that triplets need to be kept simple for generalization.

As the initial set of rules was created for case 1 and all words are matched based on their lemma, the results are expected to be poor on case 2, because the two cases have different vocabularies. As expected relying only on the initial set of rules, there are no triplets extracted from case 2.

To include case-specific rules and vocabulary for case 2 as well, we decided to create a set of rules for the sample solution of case 2 imitating the existence of gold triplets. This set of rules is combined with the initial set of rules. The combined ruleset consists of 97 rules in total compared to the initial 79. Thanks to the proposed extension 18 rules cover the vocabulary of case 2. If these rules created on the sample solution of case 2 can generalize well on the attempts of case 2, then we expect multiple found triplets also for case 2. Unfortunately, as the results for scenario lemma_combined show (see Table 5.2), only two triplets were found in the attempts of case 2. These triplets are *rechtswidrig(Bernds, Verhalten)* and *eingegriffen(geschütztes_Rechtsgut)*. The triplets are true positive extractions, i.e. the precision is 1, and even a point could be awarded based on these found triplets, but the number of extracted triplets is lower than expected. This also shows, how challenging the generalization is even on the attempt level.

As for case 1, the extension of the initial set of rules does not have an impact, the results stay the same. Thanks to the diverse vocabulary used, the newly introduced rules for case 2 do not extract any triplets from case 1.

5.1.3 Lemma-POS-based matching

A less strict and more general matching algorithm is lemma-POS-based matching While the predicates are still matched based on their lemma, the arguments are matched based on Part-of-Speech labels, as described in section 4.1. Lemma-POS-based matching is used in two scenarios, lemma_POS_initial and lemma_POS_combined. The results for the scenario are shown in Table 5.3.

In addition to all previously extracted triplets that still meet the matching criteria, additional triplets are found when looser, more general matching algorithms such as Lemma-POS-based are used. This means the initially obtained recall of 1 is still achieved for case 1 as all previously extracted triplets still occur among the higher number of extractions.

Using the initial set of rules in combination with lemma-POS-based matching leads an extraction of 183 triplets for case 1. This is a significant increase compared to the

Scenario	Legal case	Recall	Precision	No. Extracted Triplets
lemma_POS_initial	1	1	0.844	183
lemma_POS_initial	2	0	1	3
lemma_POS_combined	1	1	0.844	183
lemma_POS_combined	2	0.154	0.778	9

Table 5.3: Performance metrics by scenario and legal case using lemma-POS-based matching algorithm: The metrics are calculated on the student attempts of the cases. Each scenario is determined by the combination of the ruleset and the matching algorithm. The scenarios are described in section 4.3

previously found 86 triplets. The larger amount of extracted triplets results in some meaningless, i.e. False Positive, extractions reducing the initially high precision of 1 to 0.844 for case 1.

Thanks to the more general lemma-POS-based matching, even while the initial set of rule is used, the algorithm is capable of extracting triplets from the second case. In total 3 extracted triplets were found compared to the failed extractions with lemma-based matching:

- *vor_liegt(Vertrag)*,
- *liegt_vor(Beweislastumkehr)* and
- *liegt_vor(wissentliche_Gefährlichkeit)*.

These triplets are again true positives, hence the precision is 1 but no point could be awarded for any attempt, therefore recall remains 0. Nonetheless, the results highlight an important lesson: During the initial testing using lemma-POS-based matching, the realization was made, that the targeted words and phrases need to be chosen as predicates. The reason was that two sentences with the same content can be formulated differently: *Es ist eine bewegliche Sache* or *Es gibt eine bewegliche Sache.* If the annotation contained the predicate *beweglich* and the argument *Sache*, the triplets were extracted in both cases. However, if the verb phrase is part of the annotation, such as *ist(beweglich,Sache)*, then the extraction fails on a slightly different sentence, such as on *Es gibt eine bewegliche Sache*, because the verbs used and the parsed structure of the sentence differ. Consequently, using the phrases of interest as predicates helped the generalization on the attempt level. As soon as the same set of rules was applied to a different case, to case 2, we discovered the drawback of the previously described logic, that helped generalization on the attempt level: Only triplets were extracted, that used more general verbs as predicates, such as the verb *vorliegen.* The usage of case-specific phrases as predicates, such as *beweglich* fails to generalize well over multiple cases if cases use different vocabularies, and phrases of interest when lemma-based matching is used for the predicates. This indicates, that using general phrases instead of case-specific phrases as predicates is advantageous for overall generalization. However, this conclusion is contradictory to the observation on the attempt level, when the usage of phrases of interest as predicates helps the generalization.

In scenario lemma_POS_combined, similarly lemma-POS-based matching is used but this time with the combined rule set. The combined rulest introduced some case-specific vocabulary for case 2 thanks to the rules created on the sample solution of case 2. It expects to perform better on case 2 compared to the previous scenarios. As Table 5.3 shows, the performance increases in fact, but only slightly. In total 9 triplets are extracted from the 8 attempts, marginally improving the recall. The results are still not satisfactory. Using a more general lemma-POS-based algorithm only manages to extract 7 additional triplets compared to scenario lemma_combined. This again highlights the generalization challenge on the attempt level within a case. The rules created on the sample solution of case 2 introduce case-specific vocabulary. So strictly speaking the vocabulary barrier for generalization is partly eliminated. However, even in scenario lemma_POS_combined only 9 triplets from the attempts of case 2 were extracted due to the various phrasing of the attempts. Therefore the 18 case 2 specific rules generalized quite poorly on the attempts within the case.

5.1.4 POS-based matching

To expand the generalization capabilities POS-based matching is also tested in scenario POS_combined. The results are shown in Table 5.4. POS-based matching significantly increased the number of extracted triplets. The high number of extracted triplets has an impact both on recall and precision.

Scenario	Legal case	Recall	Precision	No. Extracted Triplets
POS_combined	1	1	0.524	576
POS_combined	2	0.846	0.507	353

Table 5.4: Performance metrics by scenario and legal case using POS-based matching algorithm: The metrics are calculated on the student attempts of the cases. Each scenario is determined by the combination of the ruleset and the matching algorithm. The scenarios are described in section 4.3

As hoped, the recall increases significantly for case 2. For 4 out of the 6 examined attempts, a recall of 1 is achieved (see in Table 5.5. For attempt 2 of case 2, a piece of relevant information (a Latin phrase, *ex delicto*) is missed in the extraction. Consequently, one finds only 2 points based on the extraction whereas an expert awarded 3 points for the student. Similarly, for attempt 1, a written paragraph number should have been extracted in combination with a Latin phrase. Yet, they were not found, hence no point could be awarded based on the extraction, while the expert awarded 1 point for the student. Due to these 2 missed points, only 11 points could be awarded based on the extraction compared to the 13 points rewarded by the domain expert. Thus, the recall is 0.846.

The combined ruleset consists mainly of nouns, adjectives, and verbs. Proper nouns, personal pronouns, and numbers are underrepresented as in the evaluation guidelines these types of POS are rarely highlighted by domain experts. The limitation of POS-based matching is displayed by the mentioned attempt 1 of case 2: The student received one point from the domain expert as a Latin phrase (*ex delicto*) was written in combination with a paragraph reference. However, neither the paragraph reference nor the Latin phrase were found by the algorithm. As stated previously, numbers are underrepresented in the set of rules, in fact, no pattern was composed for numbers as in the sample solutions and provided guidelines, they have not been relevant so far. Hence, the reference to a paragraph's number could not be found by the algorithm in the first attempt of case 2. By including ad hoc patterns, involving numbers, in to the ruleset, the paragraph references could be extracted leading to a higher quality of extractions. Handling occasional Latin phrases is more challenging for POS-based matching using a German parser. The parsing of the phrase *ex delicto* is correctly done (see in Figure 5.6). The German parser recognized, that the words of the Latin phrase belong together and they are grouped in the parsed tree with a compound relation in between. The POS of the words in the phrase are categorized as proper nouns (PROPN), which is wrong and misleading in a POS-based matching. Likewise, all the words of the Latin phrase of *conditio sine qua non*, from the sample solution of case 1, are categorized as PROPN.

Similarly to Latin phrases, abbreviations might be also wrongly categorized regarding POS. In the attempts, it is quite common to abbreviate the names of individuals. In the example sentence (Figure 5.6) *P* refers to Paula. Originally, it was expected, that the parser would fail to correctly identify the abbreviation, but it correctly identified as PROPN. Other specific characters and abbreviations, that are commonly used in legal text are § or ABGB. These frequently occurring characters have consistently the same POS: NOUN and PROPN respectively. These potential mistakes in the POS identification of the parser have consequences: These misleading labels result in worse precision as some unintended triplets get extracted, such as *§(fordern)* from attempt 1 in case 1 from the German sentence of *Sie kann auch von G auf Grund seines Auswahlverschuldens SE fordern (§ 1315 ABGB)*. The reason of the extraction is the following. The pattern shown in Figure 5.7a was created by annotating the sentence *Das Fahrrad wurde entgeltlich erworben, Titel ist der Kaufvertrag und wurde das Fahrrad laut SV auch übergeben (modus)* from attempt 5 of case 1. The pattern essentially translates to Figure 5.7b, when POS-based matching is used. For that pattern, the algorithm searches for nouns with the corresponding *APPOS* UD relation. The character § was identified as a noun and matches the pattern, hence the extraction. However, the extracted triplet is meaningless, i.e. False Positive, resulting in worse precision.

Nevertheless, one can cope with the misleading POS labels knowing that the given characters get consistently wrongly qualified: Latin words are identified as PROPN and § as a NOUN. If one writes patterns with that in mind or annotates Latin phrases, they

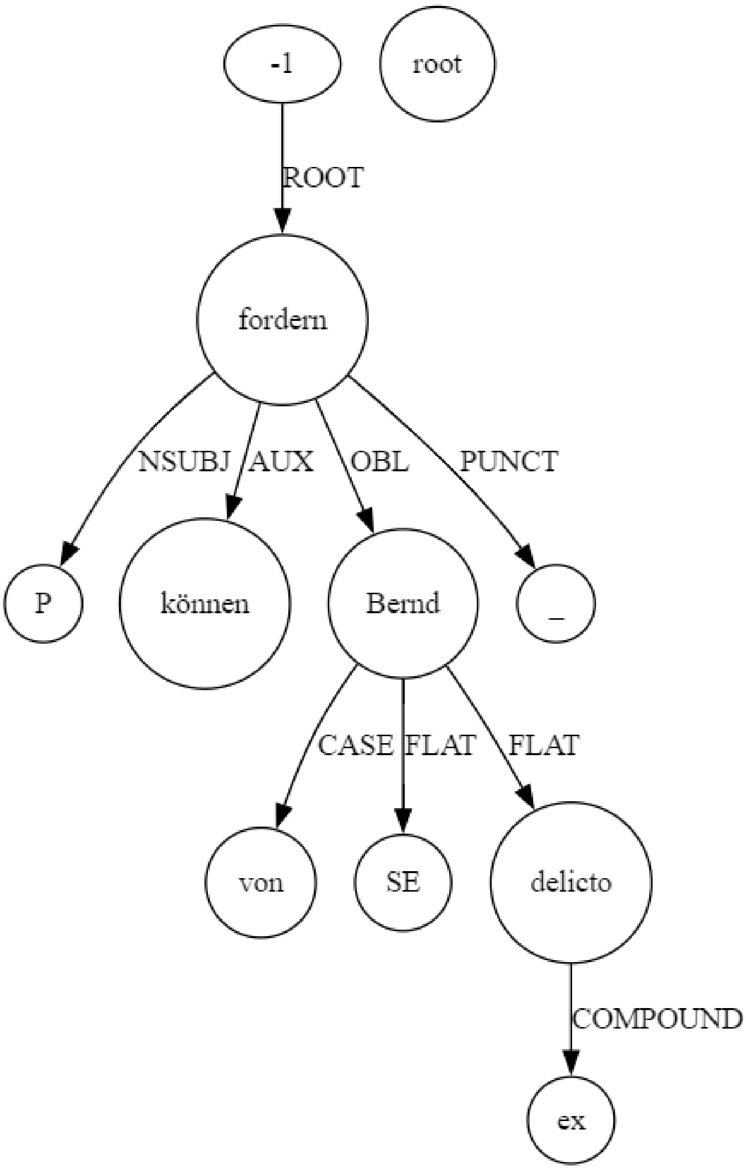
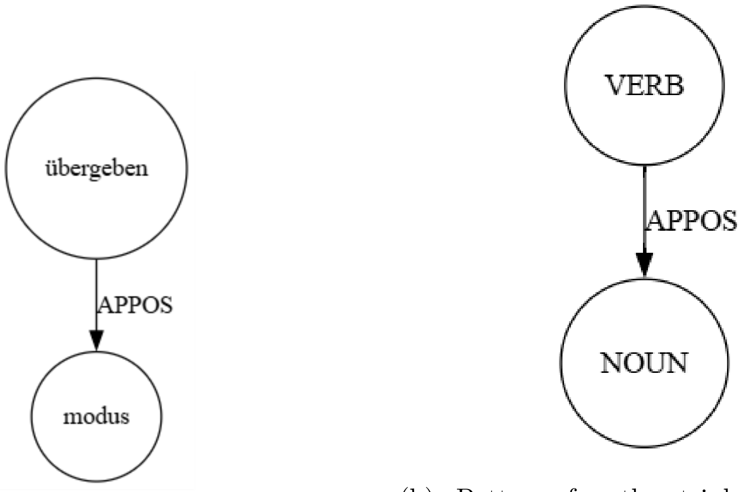


Figure 5.6: UD parsed tree representation of the German sentence: *P kann von Bernd SE ex delicto fordern.*

will be found and extracted. Proving this, the following pattern was created, shown in Figure 5.8, where both nodes are PROPN and the UD relation connecting them is compound, like in Figure 5.6. As expected, thanks to this pattern the triplet *ex_delicto()* was found and extracted. However, it must be noted, that each Latin phrase needs to be evaluated individually to see, how the German parser handles them. The previously mentioned other Latin phrase, *conditio sine qua non*, has a different parsed structure



(a) Pattern for the triplet annotation *modus(übergeben)*

(b) Pattern for the triplet annotation *modus(übergeben)*, when POS-based matching is used

Figure 5.7: The matching pattern for the annotation *modus(übergeben)* of the sentence *Das Fahrrad wurde entgeltlich erworben, Titel ist der Kaufvertrag und wurde das Fahrrad laut SV auch übergeben (modus).* Figure 5.7a translates to Figure 5.7b if POS-based matching is used.

with other UD relations between its words. Therefore it can not be assumed, that all the Latin phrases have the UD relation, *compound*. Nevertheless, if the Latin phrase and its parsed representation are known, then a dedicated pattern could be created for its extraction. This is why the potentially relevant Latin phrases need to be defined by the domain experts.

To further inspect generalization, the results for the attempts of case 2 are shown in Table 5.5. One might ask, how it is possible, that for attempt 1 the recall is 0, but the precision is 0.6 in Table 5.5. The explanation lies in the business case specific evaluation. For attempt 1 and attempt 3 I slightly deviated from the domain expert evaluation for the precision calculation: In attempt 1 multiple triplets about the alcohol problem were extracted. The domain expert did not award a point for the mention of the alcohol problem, as further information would be required to obtain the full point. Therefore the mention of the alcohol problem was not even highlighted directly in the text by the domain expert. However, in the detailed guidelines alcohol problem is an information segment of interest. Therefore, regarding IE and the feedback system, finding out, that the alcohol problem was mentioned in the text is relevant and the rule system is intended to extract it. Accordingly, I decided to count the triplets mentioning alcohol problems as true positives for the precision calculation. Similarly, for attempt 3 no points were given at all, but I marked some extracted triplets, such as *eine_Delikatshaftung()*, *Sohn(Erfüllungshilfe)* and *Schadenersatz(Paula, verlangen)* as true positive for the same reason: the information itself is not enough for awarding the attempt with a point but

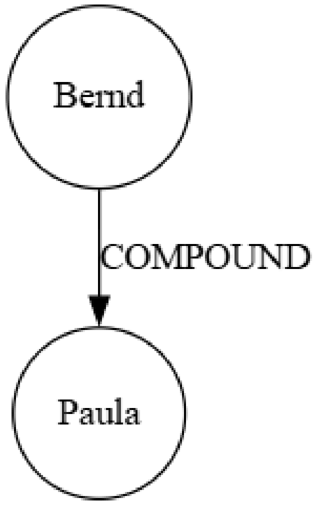


Figure 5.8: A dedicated pattern to extract the triplet *ex_delicto()*

the extractions represent useful information, that the feedback system needs to be aware of. Therefore I marked them as true positives.

Attempts	1st	2nd	3rd	4th	5th	6th
Recall	0	0.667	1	1	1	1
Precision	0.6	0.571	0.500	0.435	0.397	0.630
Points based on extractions	0	2	0	2	3	4
Points given by experts	1	3	0	2	3	4
No. extracted triplets	10	84	16	92	78	73

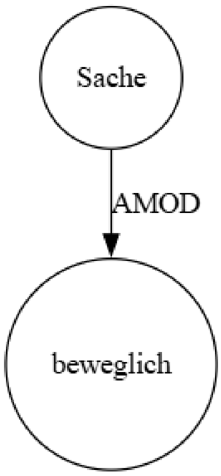
Table 5.5: Qualitative evaluation of students' attempts for case 2 using POS-based matching combined with the combined ruleset. As expert evaluation is provided only for 6 attempts, the other 2 attempts are ignored.

Another effect of the high number of extracted triplets is the noise in the results, i.e. a low precision, and a high number of extracted triplets. The precision both for case 1 and case 2 is slightly above 0.5. This result means, that nearly every second triplet was labeled as false positive.

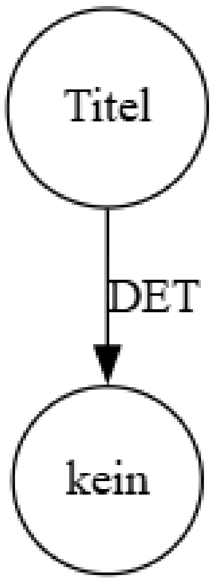
A good example for not desirable extractions besides the previously mentioned *§(fordern)*, is the pattern shown in Figure 5.9b for the triplet *Titel(kein)*, which will be mentioned again in subsection 5.3.1. This pattern with the combination of POS-based matching turns into *(u_24 / NOUN :DET (u_22 / DET))*, where DET stands for determiner. The issue with this pattern and especially with the determiner is, that not just *kein* (in English not) is a determiner but all the German articles, *der*, *die* and *das* are determiners. Consequently, the combination of this pattern with POS-based matching results in many extractions such as *Fahrrad(das)* or *Kauf(den)* etc. These extractions are useless False

Positive extractions leading to lower precision.

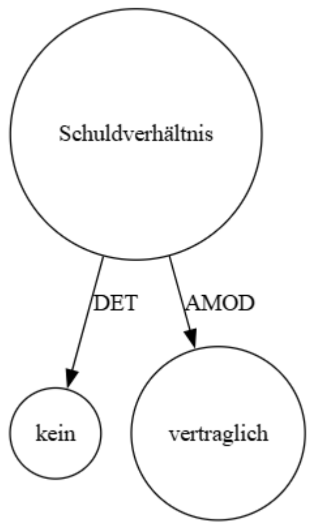
It must be also mentioned, that some highly similar triplets occur, such as *schweren(Schaden)*, *einen(Schaden)* and *einen(schweren, Schaden)* for attempt 4 from sentence *P erleidet einen schweren Schaden*. The reason for similar extractions is that some rules become very similar when POS-based matching is used. For the three similar examples mentioned above, the corresponding extraction patterns are shown in Figure 5.9. These are different rules as long as one considers the words. However, as soon as the matching algorithm is POS-based all the 3 rules become highly similar. The pattern, Figure 5.9c covers the information of the other two patterns. Therefore one could argue, that instead of using all three extraction patterns, one should only use the broadest one, i.e. the one shown in Figure 5.9c. While this seems to be a promising idea based on this example, let us recall the conclusion found in subsection 5.1.1: try to keep the annotation and most importantly the patterns as simple as possible. Let us consider a sentence on which the triplet *kein(Titel)* was created on: *Jedoch liegt kein gültiger Titel vor*. The sentence was already shown before in Table 3.1. If only the broadest extraction pattern would be used, then no extraction would be made on this simple sentence: The extraction pattern (Figure 5.9c) does not match the parsed sentence graph (Figure 5.10). Therefore all the simple extraction patterns need to be kept.



(a) Pattern for the triplet annotation *beweglich(Sache)*



(b) Pattern for the triplet annotation *kein(Titel)*



(c) Pattern for the triplet annotation *vertraglichen(keinem, Schuldverhältnis)*

Figure 5.9: Three matching patterns from different sentences, that cover nearly identical pattern structures, when POS-based matching is used.

Finally, I would like to note, that using POS-based matching transforms the initially written patterns, as the meaning of the words does not count anymore, only the part-of-

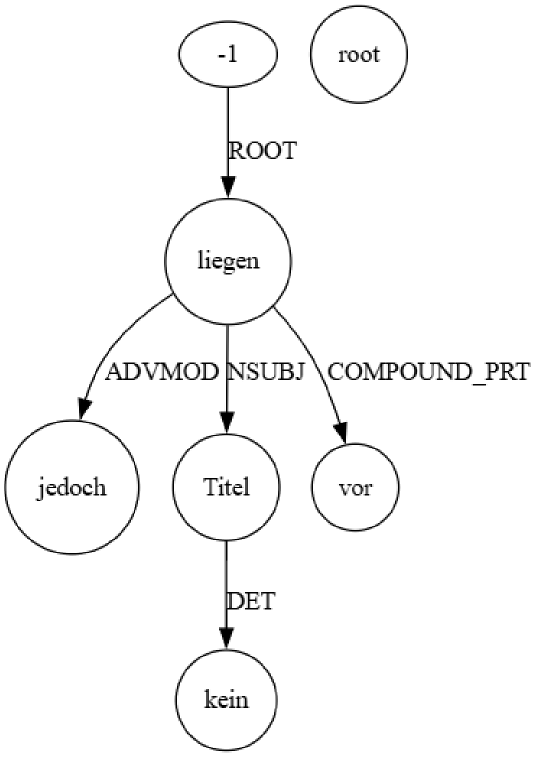


Figure 5.10: The UD parsed graph representation of the German sentence: *Jedoch liegt kein Titel vor.*

speech of the words in the pattern matters. For example initial set of rules consists of the following two patterns: *(u_6 / Sache :AMOD (u_5 / beweglich))* and *(u_5 / Titel :AMOD (u_4 / gültig))*. These patterns are different if lemma-based matching is involved. However, using only POS-based matching makes these two identical: *(u_6 / NOUN :AMOD (u_5 / ADJ))*, as *Sache* and *Titel* ar German nouns and *beweglich* and *gültig* ar German adjectives. The combined ruleset consists of 97 patterns in total, to compress the size of the ruleset a modification to the ruleset can be made. I manually filtered the ruleset to only include syntactically unique patterns. Thanks to this syntactical filtering the size of the ruleset is reduced to 65. To test if the reduced ruleset performs the same as the combined ruleset and no mistakes were made, the scenario POS_filtered was executed using POS-based matching with the combination of the filtered set of rules. This scenario is called *POS_filtered* in **??**. The results of scenario POS_filtered match with the results of scenario POS_combined, thus the filtering was correct, and the same triplets were found using the filtered, syntactically unique set of rules. This syntactically unique set of rules only works if POS-based matching is used. In all other scenarios, the combined set of rules needs to be used.

5.2 Overview

To summarize the findings, the results are combined in Table 5.6. The initial set of rules was created for case 1, hence does not include the vocabulary of case 2, but covers each relevant information of case 1. Therefore it was expected, that recall stays 1 for each scenario for case 1, while precision decreases during the generalization steps. The results confirm the expectation. Each generalization step leads to improved performance on Case 2, but at the same time, the precision of Case 1 decreases. Consequently, the more sentence-specific the rules are, the higher the precision on the target text, and the lower the generalization capability on an unseen text is .

Lemma-based matching (scenarios lemma_initial and lemma_combined) performs well on the data it was created for, achieving both a recall and precision of 1 on case 1. However, the generalization is poor: In scenario lemma_initial, no triplet is found for the previously not-seen case, i.e. case 2. Even after extending the initial set by the rules defined for the sample solution of case 2. Only 2 triplets are found on the attempts of case 2 in scenario lemma_combined.

Switching from lemma-based matching to Lemma-POS-based matching increases the extraction capability on case 1, but the performance is still poor on case 2 in scenarios lemma_POS_initial and lemma_POS_combined. However, the generalization step is observable compared to the previous lemma-based matching. It must be also noted, that for case 1, the number of extracted triplets significantly increases and the precision drops to 0.844 due to the generalization step.

Switching completely to POS-based matching for both predicate and arguments (scenario POS_combined) is beneficial for generalization. A recall of 0.846 is reached on the attempts of case 1. But at the same time, the number of extracted triplets skyrockets, reaching 576 extracted triplets for the 8 attempts of case 1. At the same time, the precision also drops to 0.524.

Finally, I would like to note, that initially, I believed POS-based matching would create too many false positive triplets, strongly limiting the precision. However, the strict syntactic patterns restrict the number of triplets that get extracted, therefore the achieved precision of roughly 0.5 with the help of the uniquely filtered syntactic patterns is better than the expectation. The actual precision for the business case will be even better, as precision was calculated by evaluating, if the extracted triplet contains useful information and if not they were marked as false positive reducing the precision score. However, from the business case perspective, as described in section 4.2, only those triplets are false positive, which lead to false scoring. Extractions like *Fahrrad(das)* or *Kauf(den)* are currently marked as false positives and reduce the precision. However, they do not influence the scoring, therefore from the business case perspective, they do not reduce the precision.

Nevertheless, to further improve the precision, one could use a list of synonyms for matching. Node matching based on a list of synonyms is more general than lemma-based matching but is more specific than POS-based matching. Matching based on a synonym

Scenario	Legal case	Recall	Precision	No. Extracted Triplets
lemma_initial	1	1	1	86
lemma_initial	2	0	0	0
lemma_POS_initial	1	1	0.844	183
lemma_POS_initial	2	0	1	3
lemma_combined	1	1	1	86
lemma_combined	2	0.077	1	2
lemma_POS_combined	1	1	0.844	183
lemma_POS_combined	2	0.154	0.778	9
POS_combined	1	1	0.524	576
POS_combined	2	0.846	0.507	353
POS_filtered	1	1	0.524	576
POS_filtered	2	0.846	0.507	353

Table 5.6: Performance metrics by scenario and legal case: The metrics are calculated on the student attempts of the cases. Each scenario is determined by the combination of the ruleset and the matching algorithm. The scenarios are described in section 4.3

list also eliminates the issue of gender-specific nouns, a German language-specific issue discussed in section 4.1. False extractions thanks to misleading POS labels would also disappear when a synonym-based matching is applied. The disadvantage of synonym-based matching is, that a synonym list needs to be defined and adjusted to the vocabulary of the corresponding legal case. The need for adjustment limits the generalization and requires human intervention for each different case. However, in the business case it can be assumed, that for each case a solution guideline will be provided in a requested form, therefore one could ask the domain expert to define synonym lists for each case to enable synonym-based matching. The currently provided evaluation guidelines do not yet include appropriate synonym lists, therefore this approach could not be tested yet.

5.3 Error analysis

As my experiments and results show, patterns and the matching algorithm determine the quality of the open information extraction. If one defines precise patterns and uses a matching algorithm based on a lemma, then both a high recall and precision can be reached. However, high-precision patterns and matching algorithms generalize poorly, as shown in Table 5.6 in scenarios lemma_initial and lemma_combined for case 2. If one desires better generalization capabilities, it usually comes at a cost: reduced precision and a high number of extracted triplets. This was proven on case 1, as changing to more general matching algorithms led to a reduced precision and a higher number of extracted triplets by each step. Finding the ideal balance of precision and generalization can be

Den	G	trifft	kein	Verschulden,	weil	es
The	G	meets	no	fault,	because	it
ihm	objektiv	und	subjektiv	nicht	vorwerfbar	ist.
him	objectively	and	subjectively	not	reproachable	is.

Table 5.7: The German sentence: *Den G trifft kein Verschulden, weil es ihm objektiv und subjektiv nicht vorwerfbar ist*, and its corresponding English translation *The G is not at fault because he cannot be blamed objectively and subjectively.* Here G stands for Gerhard.

done by defining the optimal combination of a ruleset and the corresponding matching algorithms.

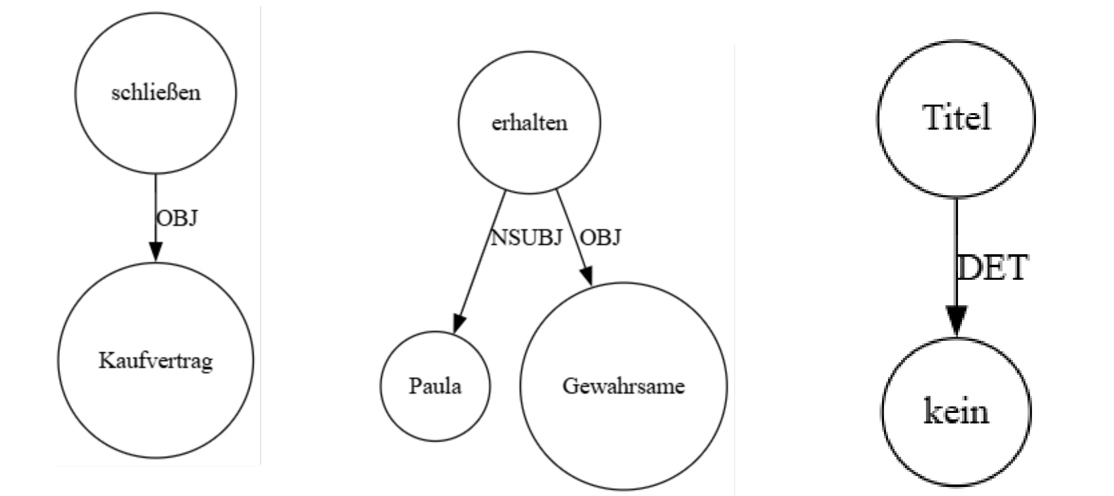
5.3.1 Affirmation, Denial, Questioning

An identified limitation of the rule-based OIE is, that it extracts structures described by the defined patterns. Consequently, for all possible outcomes, a rule needs to be created. If a meaning-modifier extension, such as a negation, is part of the target structure, it gets ignored unless it is also part of the pattern. Let us consider attempt 4 of case 2, which contains the following sentence shown in Table 5.7.

The sentence clearly states that G is not responsible. The extracted triplets (using POS-based matching) from this specific sentence containing the word *Verschulden* are the following: *Verschulden(trifft)*, *Verschulden(G, trifft)* and *Verschulden(kein)*. The first two extracted triplets are valid but misleading. The patterns (see Figure 5.11a and Figure 5.11b) extracted them do not contain any possible structure of negation. Therefore using only these two patterns results in a valid extraction, but the extracted triplets are misleading as the denial does not appear. Knowing this a simple negation pattern for nouns was implemented, the previously also shown Figure 5.11c. Thanks to this pattern the triplet *Verschulden(kein)* is also extracted, indicating the presence of a negation in the sentence. This example highlights the importance of considering all relevant extracted triplets for a sentence to get the entire picture. Additionally, it needs to be emphasized, that all 3 triplets are valid, but without creating a pattern for negation, the extraction would be highly misleading.

This behavior of rule-based OIE is critical for the business case as it needs to be examined in numerous situations, whether a mentioned statement is affirmed or denied. In the concrete business case, besides affirmation and denial, another aspect also appears: the domain experts also ask students to state, what needs to be investigated or questioned in a given legal use case. Such a questioning statement can be the formulation shown in Table 5.8.

To summarize a statement can have 3 forms: affirmed, denied, or questioned. Affirmation can be tested by the absence of both negation and questioning. Denial is somewhat easier to test as it is frequently expressed simply by using the words *kein* or *nicht*. However, in



(a) Pattern for the triplet annotation *Kaufvertrag(geschlossen)*

(b) Pattern for the triplet annotation *Gewahrsame(Paula,erhalten)*

(c) Pattern for the triplet annotation *kein(Titel)*

Figure 5.11: Three matching patterns from different sentences each, applied to the sentence of Table 5.7

Zu	prüfen	ist	daher	ein	gutgläubiger	Erwerb	vom
To	examine	is	therefore	a	because	acquisition	bona fide
Nichtberechtigten.	ihm	objektiv	und	subjektiv	nicht	vorwerfbar	ist.
non-entitled party	him	objectively	and	subjectively	not	reproachable	is.

Table 5.8: The German sentence: *Zu prüfen ist daher ein gutgläubiger Erwerb vom Nichtberechtigten.*, and its corresponding English translation *A bona fide acquisition from the non-entitled party must be examined.*

Fanny	erwirbt	das Fahrrad	von	jemandem,	ein
Fanny	acquires	the bike	from	someone	a
deriativer	Erwerb	des Eigentums	scheidet	damit	aus.
derivative	acquisition	of ownership	rule	so	out.

Table 5.9: The German sentence: *Fanny erwirbt das Fahrrad von jemandem, ..., ein derivativer Erwerb des Eigentums scheidet damit aus.*, and its corresponding English translation *Fanny acquires the bicycle from someone, ..., a derivative acquisition of ownership is therefore ruled out.*

some cases, negation is expressed by the usage of specific verbs, such as *ausscheiden* (in English *rule out*) as in the sentence in Table 5.9

To extract the negation information from such a sentence is more challenging as one needs to know the meaning of the verb. Finally, testing whether a statement is questioned may be the trickiest as there is no standardized simple way to do so. In the attempts, it was frequently expressed by the phrase: *ist zu prüfen* (in English: *is to be examined*) as in the sample solution of case 2: *Zu prüfen ist daher ein gutgläubiger Erwerb vom Nichtberechtigten (§ 367 f ABGB)*, also shown in Table 5.8. Hence a corresponding grammatical pattern can be defined. But that pattern would be still specific to the phrase *ist zu prüfen* and would not work for other formulations of questioning.

The use of POS-based matching and thoroughly defined grammatical structures should cover numerous ways to express negation and questioning. However, these are not part of the scope of my thesis but are rather highlighted for future work.

5.3.2 Research Questions

Finally, I would like to reflect on the formulated research questions and problem statement. By writing a dedicated pattern, any information could be extracted from any text using rule-based OIE, as it was shown for the Latin phrase *ex delicto* in subsection 5.1.4. The challenge for the extraction relies in its generalization capability.

Generalization can be discussed on two levels: the attempt level and the overall case level. On the attempt level, the generalization among attempts within a case was examined. These attempts have the same vocabulary as they each describe the same legal case. Case-level generalization is a higher level of generalization, as different legal cases may not necessarily share the same vocabulary. In subsection 5.1.3 the conclusion was made, that the usage of general phrases instead of case-specific phrases as predicates is advantageous for case-level generalization. However, this conclusion is contradictory to the observation on the attempt level, where the usage of phrases of interest as predicates helps the generalization.

Specific syntactic patterns were expected to be the main challenges for the rule-based Open Information Extraction. It was found, that truly as expected, complex patterns, i.e. patterns including multiple words, tend to be highly sentence-specific such as the mentioned example Figure 5.2 in subsection 5.1.1. The complex patterns depend highly on the formulation of the sentences and consequently are less suited for generalization.

It was discovered, that besides highly specific syntactic graphs, the main problem for generalization is the node-matching algorithm. The choice of the matching algorithm has an even more significant aspect. Using lemma-based matching is beneficial for precision, but it is highly limiting for generalization, as shown in Table 5.6. If the vocabulary of each legal case is different and lemma-based matching is used, then inevitably the set of rules needs to be adjusted for each new legal case. On the contrary POS-based matching generalizes better and does not require rule adjustment. But it has precision issues. Although for the presented business case, this is less of an issue as only those triplets are false positive extractions from the business case's perspective, which negatively impact the scoring.

There are multiple domain-specific complications. The way legal texts are formulated is complex, involving compound sentences, and extensive remarks in parenthesis, such as in

deliktischer	Schadenersatzanspruch
tortious	claim for damages

Table 5.10: The German phrase: *deliktischer Schadenersatzanspruch* , and its corresponding English translation *tortious claim for damages*

Table 5.1. These comprehensive notes in parenthesis lead to highly specific extraction patterns such as inFigure 5.1, resulting in sentence-specific patterns. Besides compound sentences, the usage of Latin phrases and potential references to paragraphs are also common in the legal domain. As discussed in subsection 5.1.4, Latin phrases can cause issues especially, when POS-based matching is used. The challenge of Latin phrases is to know what to look for. Once it is known what the relevant Latin phrase looks like and how it is handled by the parser, then an ad hoc extraction pattern could be defined, such as for the Latin phrase *ex delicto*. Without precisely knowing the Latin phrase of interest it is hard to create a corresponding pattern for its extraction because the way Latin phrases are parsed differs. Based on what we've seen, it's likely that each word will be identified as a PROPN, but the UD relation among the words might vary.

The phrase shown in Table 5.10 is a statement, that needs to be extracted. That information is expressed typically in German, however, there is also a Latin phrase for it, namely the previously mentioned *ex delicto* in the sentence in Figure 5.6. This phrase is equivalent to the German phrase. Additionally, the domain experts noted, that section 1315 also refers to the tortious nature, hence the combination of a claim for damage with the reference to § 1315 also carries the same information. A promising solution for the issue of interchangeable phrases could be something like a synonym list. With the help of a domain expert one could create a list for the information of interest: how that piece of information could be expressed differently. Latin phrases can be covered by dedicated rules, once the phrase and its parsing structure are known as was shown in subsection 5.1.4. As for the paragraph references, I believe a higher degree of autonomy is possible. One can create syntactic patterns to cover the numeric references without domain expertise. It is not necessarily required to know the reference in advance, but for interpretation of the results, domain knowledge is required.

By improving the extraction of Latin phrases and the numeric paragraph references, the system has the potential to obtain all the relevant information segments from the 6 attempts of case 2. Thus, all 13 points could be awarded based on the extraction, consequently, a recall of 1 could be reached instead of the current 0.846.

Considering precision I can refer to the character § and its connected issue discussed in subsection 5.1.4: The parser recognizes § as a NOUN. The misleading POS labels lead to False Positive extractions such as the mentioned example in subsection 5.1.4, *§(fordern)*- These extractions reduce the precision. However, this issue occurred rarely as misleadingly labeled POS words and characters are rare in the text. Therefore for solving this issue only a slight increase in precision is predicted.

The initial scientific question, my thesis aims to investigate the generalization capability of rule-based Open Information Extraction extraction over multiple legal texts corresponding to diverse legal cases. My thesis presents a combination of a ruleset[1] with a matching algorithm, that reaches recall, higher than 0.8 for the inspected legal cases while having a precision higher than 0.5. Additionally, the quantitative analysis provided describes the main challenges and considerations of the generalization and proposes corresponding solutions for the future work in subsection 5.3.3.

5.3.3 Future work

Based on the results, POS-based matching performed the best for generalization. It is recommended to extend the set of rules by numbers and patterns to find paragraph references and unique Latin phrases. This has the potential to increase the recall to 1. To improve the precision a new node-matching algorithm based on synonym lists seems to be a promising approach. The creation of the synonym lists requires clean instruction by domain experts, including Latin phrases and potential paragraphs corresponding to the legal statement. The underlying rule system for the synonym-based approach could be based on the syntactically unique set of rules.

A current limitation of my work is, that the extraction was only tested on attempts of two legal cases. While this is sufficient to make generally valid conclusions about generalization, as part of future work, it would be useful to extend the analysis to the attempts of the other 8 legal cases, as soon as detailed guidelines are provided for them, describing which information is relevant in those cases. Additionally, the question of affirmation, denial, and questioning of a statement, discussed in subsection 5.3.1, is highly relevant to the business case and needs to be investigated further.

[1]accessible under: `https://github.com/izsom/rbOIE_GerLig`

Overview of Generative AI Tools Used

In this thesis, generative AI tools such as Grammarly and DeepL were employed to refine and reformulate specific sentences. For translation tasks and to identify suitable synonyms, enhancing the precision and fluidity of the language DeepL was used. These tools served as assistants, aiding in the refinement of the text, but were never used to generate large blocks of content.

List of Figures

List of Tables

Bibliography

Akbik, A. and Löser, A. (2012). KrakeN: N-ary facts in open information extraction. In Fan, J., Hoffman, R., Kalyanpur, A., Riedel, S., Suchanek, F., and Talukdar, P. P., editors, *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montréal, Canada. Association for Computational Linguistics.

Bassa, A., Kröll, M., and Kern, R. (2018). Gerie - an open information extraction system for the german language. *Journal of Universal Computer Science*, 24(1):2–24.

Bhardwaj, S., Aggarwal, S., and Mausam (2019). CaRB: A crowdsourced benchmark for open IE. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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, Hong Kong, China. Association for Computational Linguistics.

Cui, L., Wei, F., and Zhou, M. (2018). Neural open information extraction. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia. Association for Computational Linguistics.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford dependencies: A cross-linguistic typology. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In Barzilay, R. and Johnson, M., editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Falke, T., Stanovsky, G., Gurevych, I., and Dagan, I. (2016). Porting an open information extraction system from english to german. pages 892–898.

Fan, A., Gardent, C., Braud, C., and Bordes, A. (2019). Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *CoRR*, abs/1910.08435.

Han, X., Gao, T., Lin, Y., Peng, H., Yang, Y., Xiao, C., Liu, Z., Li, P., Zhou, J., and Sun, M. (2020). More data, more relations, more context and more openness: A review and outlook for relation extraction. In Wong, K.-F., Knight, K., and Wu, H., editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

Jiang, J. (2012). *Information Extraction from Text*, pages 11–41. Springer US, Boston, MA.

Kasper, R. T. (1989). A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Khot, T., Sabharwal, A., and Clark, P. (2017). Answering complex questions using open information extraction. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Vancouver, Canada. Association for Computational Linguistics.

Kolluru, K., Mohammed, M., Mittal, S., Chakrabarti, S., and Mausam (2022). Alignment-augmented consistent translation for multilingual open information extraction. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland. Association for Computational Linguistics.

Kotnis, B., Gashteovski, K., Rubio, D., Shaker, A., Rodriguez-Tembras, V., Takamoto, M., Niepert, M., and Lawrence, C. (2022). MILIE: Modular & iterative multilingual open information extraction. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6939–6950, Dublin, Ireland. Association for Computational Linguistics.

Kovács, A., Gémes, K., Iklódi, E., and Recski, G. (2022). Potato: explainable information extraction framework. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4897–4901, New York, NY, USA. Association for Computing Machinery.

Lechelle, W., Gotti, F., and Langlais, P. (2019). WiRe57 : A fine-grained benchmark for open information extraction. In Friedrich, A., Zeyrek, D., and Hoek, J., editors, *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, Florence, Italy. Association for Computational Linguistics.

Lertvittayakumjorn, P., Choshen, L., Shnarch, E., and Toni, F. (2022). Grasp: A library for extracting and exploring human-interpretable textual patterns.

Ling, C., Zhao, X., Zhang, X., Liu, Y., Cheng, W., Wang, H., Chen, Z., Osaki, T., Matsuda, K., Chen, H., and Zhao, L. (2023). Improving open information extraction with large language models: A study on demonstration uncertainty.

Mausam, M. (2016). Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 4074–4077. AAAI Press.

Mesquita, F., Schmidek, J., and Barbosa, D. (2013). Effectiveness and efficiency of open relation extraction. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S., editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Seattle, Washington, USA. Association for Computational Linguistics.

Niklaus, C., Cetto, M., Freitas, A., and Handschuh, S. (2018). A survey on open information extraction. In Bender, E. M., Derczynski, L., and Isabelle, P., editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nivre, J., Abrams, M., Agić, Ž., Ahrenberg, L., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Basmov, V., Bauer, J., Bellato, S., Bengoetxea, K., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Blokland, R., Bobicev, V., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Burchardt, A., Candito, M., Caron, B., Caron, G., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., Diaz de Ilarraza, A., Dickerson, C., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Elkahky, A., Ephrem, B., Erjavec, T., Etienne, A., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ion, R., Irimia, E., Ishola, Ọ., Jelínek, T., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kahane, S., Kanayama, H., Kanerva, J., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Kopacewicz, K., Kotsyba, N., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lam, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Mendonça, G., Miekka, N., Misirpashayeva, M., Missilä, A., Mititelu, C., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, K. S., Mori, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peng, S., Perez, C.-A., Perrier, G., Petrov, S., Piitulainen, J., Pitler, E., Plank, B., Poibeau, T., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Rießler, M., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O., Rueter, J., Sadde, S., Sagot, B., Saleh, S., Samardžić, T., Samson, S., Sanguinetti, M., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shohibussirri, M., Sichinava, D., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Spadine, C., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Szántó, Z., Taji, D., Takahashi, Y., Tanaka, T., Tellier, I., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Wallin, L., Wang, J. X., Washington, J. N., Williams, S., Wirén, M., Woldemariam, T., Wong, T.-s., Yan, C., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zeman, D., Zhang, M., and Zhu, H. (2018). Universal dependencies 2.3. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Pai, L., Gao, W., Dong, W., Ai, L., Gong, Z., Huang, S., Zongsheng, L., Hoque, E., Hirschberg, J., and Zhang, Y. (2024). A survey on open information extraction from rule-based model to large language model. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9586–9608, Miami, Florida, USA. Association for Computational Linguistics.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Poon, H., Christensen, J., Domingos, P., Etzioni, O., Hoffmann, R., Kiddon, C., Lin, T., Ling, X., Mausam, Ritter, A., Schoenmackers, S., Soderland, S., Weld, D., Wu, F., and Zhang, C. (2010). Machine reading at the University of Washington. In Mulkar-Mehta, R., Allen, J., Hobbs, J., Hovy, E., Magnini, B., and Manning, C., editors, *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 87–95, Los Angeles, California. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Sen, P., Li, Y., Kandogan, E., Yang, Y., and Lasecki, W. (2019). HEIDL: Learning linguistic expressions with deep learning and human-in-the-loop. In Costa-jussà, M. R. and Alfonseca, E., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 135–140, Florence, Italy. Association for Computational Linguistics.

Stanovsky, G., Ficler, J., Dagan, I., and Goldberg, Y. (2016). Getting more out of syntax with props.

Stanovsky, G., Michael, J., Zettlemoyer, L., and Dagan, I. (2018). Supervised open information extraction. In Walker, M., Ji, H., and Stent, A., editors, *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, New Orleans, Louisiana. Association for Computational Linguistics.

Sun, M., Li, X., Wang, X., Fan, M., Feng, Y., and Li, P. (2018). Logician: A unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 556–564, New York, NY, USA. Association for Computing Machinery.

Vania, C., Lee, G., and Pierleoni, A. (2022). Improving distantly supervised document-level relation extraction through natural language inference. In Cherry, C., Fan, A., Foster, G., Haffari, G. R., Khadivi, S., Peng, N. V., Ren, X., Shareghi, E., and Swayamdipta, S., editors, *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 14–20, Hybrid. Association for Computational Linguistics.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., Rawlins, K., and Van Durme, B. (2016). Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Wrzalik, M. and Krechel, D. (2021). GerDaLIR: A German dataset for legal information retrieval. In Aletras, N., Androutsopoulos, I., Barrett, L., Goanta, C., and Preotiuc-Pietro, D., editors, *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 123–128, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yates, A., Banko, M., Broadhead, M., Cafarella, M., Etzioni, O., and Soderland, S. (2007). TextRunner: Open information extraction on the web. In Carpenter, B., Stent, A., and Williams, J. D., editors, *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Yu, B., Zhang, Z., Li, J., Yu, H., Liu, T., Sun, J., Li, Y., and Wang, B. (2022). Towards generalized open information extraction. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1439–1453, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zini, J. E. and Awad, M. (2022). On the explainability of natural language processing deep models. *ACM Comput. Surv.*, 55(5).