TU WIEN Informatics

Symbolic Natural Language Inference for German Open Information Extraction

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

BSc Kristian Ristic
Matrikelnummer 51845332

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Gábor Recski, Univ.Ass.

Wien, 4. Mai 2025

Kristian Ristic	Gábor Recski

TU WIEN Informatics

Symbolic Natural Language Inference for German Open Information Extraction

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

BSc Kristian Ristic
Registration Number 51845332

to the Faculty of Informatics

at the TU Wien

Advisor: Gábor Recski, Univ.Ass.

Vienna, May 4, 2025

Kristian Ristic	Gábor Recski

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

BSc Kristian Ristic

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, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 4. Mai 2025

Kristian Ristic

Danksagung

Zunächst möchte ich allen Professoren, die mich auf dieser Reise begleitet haben, für ihre Leidenschaft und ihr Engagement in der Lehre danken. Jeder von ihnen ist Teil dieses kleinen, für mich aber großen Erfolgs.

Ein besonderer Dank gilt meinem Betreuer Gábor Recski, der mich schon lange vor dieser Arbeit inspiriert hat. Seine Leidenschaft und sein Zugang zu sprachwissenschaftlichen Phänomenen haben mich inspiriert, und ich bin dankbar, dass ich meine eigenen Interessen in dieser Arbeit vereinen konnte. Besonders dankbar bin ich für die schnelle und umfassende Unterstützung, wann immer ich Hilfe brauchte.

Nichts davon wäre ohne die Menschen möglich gewesen, die mir in guten wie in schwierigen Momenten zur Seite standen. Mein herzlichster Dank gilt meinen lebenslangen Freunden und den neuen, die ich auf meinem Weg kennengelernt habe, meiner Partnerin und meiner Familie.

Acknowledgements

First of all, I would like to thank all the professors I have met on this journey for their passion and dedication to teaching. Each of them is part of this small, yet for me big, success. A special expression of gratitude goes to my supervisor, Gábor Recski, who was an inspiration long before this thesis. His passion and approach to linguistic phenomena inspired me, and I am grateful to have been able to combine my own interests in this work. I am particularly thankful of the prompt and exhaustive support whenever I needed guidance. None of this would have been possible without the people who stood by me in both the good and the more challenging moments. My warmest appreciation goes to my lifelong friends and to the new ones I have met along the way, to my partner, and to my family.

Kurzfassung

Die automatisierte Bewertung offener Antworten von Studierenden anhand von Musterlösungen im Deutschen, insbesondere im juristischen Bereich, erfordert präzise und interpretierbare Methoden. Diese Arbeit begegnet dieser Herausforderung, indem sie die Bewertung von Antworten anhand von Musterlösungen als offene, durch textuelle Implikation validierte Informationsextraktion konzipiert. Wir nutzen Implikation als Verifizierungsschritt: Ein Antwortabschnitt eines Studierenden gilt nur dann als korrekt, wenn er die entsprechenden erforderlichen Informationen aus der Musterlösung enthält.

Diese Arbeit verwendet ein symbolisches Framework für die validierte Extraktion und leistet wichtige Beiträge zur Anpassung der grafischen Wissensrepräsentation an das Deutsche und zur Modellierung des Negationsumfangs. Der Kern des Frameworks basiert auf einer grafischen Wissensrepräsentation, die sowohl für Prämissenmuster aus Musterlösungen als auch für Sätze aus Antworten von Studierenden erstellt wurde. Diese Repräsentation baut auf Dependency Parsing auf, erweitert dieses jedoch erheblich durch die Einbeziehung mehrerer Ebenen linguistischer Details. Diese Ebenen umfassen eine lexikalische Ebene für Synonyme und hierarchische Beziehungen, eine Eigenschaftsebene zur Erfassung morphologischer Merkmale des Deutschen und eine Kontextebene mit regelbasierter Negationserkennung zur präzisen Modellierung des Negationsumfangs.

Diese detaillierten grafischen Wissensdarstellungen werden anschließend in einem mehrstufigen, graphenbasierten Matching-Prozess verglichen. Dieser Vergleich verifiziert die Implikation durch den Abgleich von Prämissen- und Hypothesengraphen, die Prüfung auf erforderliche Konzepte und Argumente und die Berücksichtigung der kodierten lexikalischen, morphologischen und Negationsbeschränkungen.

Die Evaluierung des Frameworks anhand der Bewertung deutscher Rechtsfalllösungen zeigt, dass die Einbeziehung externen lexikalischen Wissens und die explizite Negationsbehandlung sowohl die Präzision als auch den F1-Gesamtwert im Vergleich zu Basismethoden erhöhen. Der symbolische Charakter des Systems ermöglicht die Erklärbarkeit während des gesamten Extraktionsprozesses.

Abstract

Automated grading of open-ended student answers against sample solutions in the German language, particularly within the legal domain, necessitates methods that are both accurate and interpretable. This thesis addresses this challenge by framing the task of grading student answers against sample solutions as open information extraction validated by textual entailment. We leverage entailment as a verification step: a student's answer segment is considered correct only if it entails the corresponding required information from the sample solution.

This thesis employs a symbolic framework for validated extraction, featuring key contributions in adapting graphical knowledge representation for German and modeling negation scope. The core of the framework uses a graphical knowledge representation constructed for both premise patterns derived from sample solutions and sentences from student answers. This representation builds upon dependency parsing but significantly enriches them by incorporating multiple layers of linguistic detail. These layers include a lexical layer for synonyms and hierarchical relations, a property layer capturing morphological features specific to German, and a context layer featuring rule-based negation detection to accurately model the negation scope. These detailed graphical knowledge representations are then compared using a multi-stage, graph-based matching process. This comparison verifies entailment by matching premise and hypothesis graphs, checking for required concepts and arguments, and respecting the encoded lexical, morphological, and negation constraints.

Evaluating this framework on the task of grading German legal case solutions demonstrates that incorporating external lexical knowledge and explicit negation handling increases both precision and overall F1 score compared to baseline methods. The symbolic nature of the system allows for explainability throughout the entire extraction process.

Contents

CHAPTER 1

Introduction

Natural language understanding (NLU) involves drawing conclusions from incomplete or ambiguous information, relying on context, shared knowledge, and implicit assumptions. This allows humans to interpret implications even when some information is not explicitly stated.

In contrast, machines typically rely on patterns and correlations in training data. While advances in Natural Language Processing (NLP), such as Natural Language Inference (NLI) and NLU, have improved language tasks, machines still lack the common-sense reasoning humans use effortlessly.

This ability to reason with knowledge and context is particularly important when evaluating complex responses, such as grading open-ended exam questions in domains like law. Determining if a student's answer is correct requires understanding if it accurately reflects, or *entails*, the key information defined in a sample solution. Such task is known as *recognizing textual entailment* (RTE), a subtask of NLI focused on identifying this entailment relationship.

This thesis focuses on a specific use case from an Austrian legal publisher: developing a system to provide feedback on students' German-language legal exam answers. The core challenge involves accurately comparing student responses against detailed expert solutions and guidelines. Students may express correct legal concepts using varied phrasing, synonyms, or sentence structures, while legal language itself presents complexities like domain-specific terminology, abbreviations, and even Latin phrases. Therefore, simple keyword matching is insufficient. Furthermore, the educational and legal context demands high transparency, explainability, and reliability in the assessment, often without the large annotated datasets needed for conventional machine learning models. RTE becomes relevant in this scenario. It allows the system to assess whether segments of a student's answer, despite linguistic variations, logically entail the specific legal concepts and reasoning outlined in the expert guidelines.

Based on these requirements, this thesis models the task of grading legal exam answers in the German language as Open Information Extraction (OIE) validated by RTE. We treat essential information extracted from the sample solution as premises and sentences from the student's attempt as hypotheses. The goal is to use OIE to find potential answer segments and RTE to verify if they logically entail the required premise content.

To achieve this, we employ and evaluate a symbolic framework centered around knowledge graphical representations tailored for German. This framework utilizes graph-matching algorithms to detect entailment between premise patterns and hypothesis sentences. The approach is designed to be interpretable, leverage linguistic resources, handle specific German features, and operate effectively without large training datasets.

1.1 Motivation

NLI is a core task in NLP, serving as a foundation for downstream applications such as text summarization, question answering, and information extraction. While recent Large Language Models (LLMs) demonstrate impressive capabilities across many NLP tasks, LLMs act as a black-box and depend on statistical methods rather than explicit reasoning through problem understanding ([Zini and Awad, 2022, Nie et al., 2020, McCoy et al., 2019]), limiting their interpretability in scenarios requiring transparent logic, a critical requirement in educational and legal contexts.

Motivated by these challenges, our work models NLI using a symbolic framework that parses text into structured graphs and employs graph-matching algorithms to determine entailment. This method integrates trusted lexical and morphological resources, providing full explainability throughout the inference process. Additionally, a key motivation for this work is to develop a framework tailored to the German language. While many NLP tools and frameworks exist primarily for English, language-specific approaches for German are less common, especially in the domain of symbolic NLI.

1.2 Problem statement

The task of automatically grading student exams in German requires a system that can evaluate open-ended answers against detailed expert-authored guidelines and solutions. This evaluation process needs to be transparent and explainable at every step, allowing users to understand how the system reached its conclusions. The system must work without a large training dataset, instead relying on the expert guidelines to detect correct answers on the fly.

German language processing adds complexity to this task. The language's features like gendered articles, case-based grammar, and compound words may require specific handling. Symbolic approaches for NLI necessitate external language resources for effective implementation. Consequently, the development of an effective system requires the integration of German-specific language resources. The challenge is to combine these

resources in a way that maintains transparency while accurately processing German texts.

1.3 Research questions

In order to evaluate the effectiveness of the proposed framework, our experiments are guided by the following research questions:

1. Can symbolic rules grounded in German morphology and semantics increase the recall and precision in extracting correct answers compared to two different baseline approaches?
2. How do unique linguistic features of German, such as compound nouns, case-based syntax, and gendered articles, affect the robustness of rule-based inference?
3. Can explicit negation detection and morphological consistency checks reduce false positives in entailment extraction, a common pitfall for statistical methods?

1.4 Structure of the thesis

The thesis is structured as follows:

- Chapter 2 provides the theoretical foundation for our work. It begins with an overview of OIE techniques. The chapter then explores NLI in detail, including its role in language understanding and current challenges. Finally, it presents the specific use case of automatic exam grading for legal texts, highlighting the unique requirements and constraints of this application.
- Chapter 3 reviews existing approaches to symbolic NLI as well as state-of-the-art methods based on deep learning models.
- Chapter 4 details our employed framework. The first section describes the adaptation of the graphical knowledge representation for German, explaining how we parse and structure the sentences. The second section presents our entailment detection method, including the methods and resources used for determining correct answers.
- Chapter 5 outlines our evaluation methodology. It covers potential data preprocessing steps and describes the experimental settings, including baseline comparisons and evaluation metrics.
- Chapter 6 presents and analyzes our findings. It includes a detailed error analysis that examines the strengths and limitations of our approach.
- Chapter 7 summarizes our contributions and findings, provides the answers to our research questions, and suggests directions for future work.

CHAPTER 2

Background

In this chapter, we provide theoretical background for the key concepts used in our work. We will cover a wide range of topics within the NLP field, including semantic parsing, OIE, and NLI, providing the necessary theoretical foundation to follow the work presented in this thesis.

2.1 Introduction to NLP

Natural language processing, in literature often abbreviated as NLP, is a field of computer science and artificial intelligence that deals with the interaction between computers and human language. The overall goal of NLP is to enable machines to understand, interpret, and generate human language, both written and spoken, a task that is challenging due to the complexity and variability of human language, and the fact that humans can implicitly understand language without being aware of the rules that govern it or without being able to express it explicitly. In the last decades, different types of computational methods have been developed and applied to address the challenges of NLP. Nowadays, these NLP methods have become an integral part of widely used applications such as speech assistants, search engines, chatbots, and translation systems.

2.1.1 History of NLP

While tracing the precise origins of NLP is difficult, significant early efforts emerged during the 1960s and 1970s, generating considerable optimism, especially in machine translation. This was driven by the ambition to automatically translate languages, such as translating Russian to English during the Cold War. Despite initial excitement, these early approaches largely failed to deliver on their promises.

Between the 1970s and the 1990s, the field saw a shift towards more structured symbolic representations, emphasizing knowledge-based methods and conceptual ontologies. The

use of ontologies, formal representations of knowledge domain enabled explicit encoding of semantic information, helping NLP systems achieve notable successes in limited and well-defined scenarios. Despite these achievements, purely rule-based symbolic approaches struggled when applied to unrestricted, real-world text due to the vast complexity of language and the difficulty to scale the development of rules for every linguistic phenomenon.

In the late 1980s and early 1990s, statistical methods started to gain traction. These methods leveraged large corpora of text data to derive patterns statistically rather than relying solely on explicitly programmed logical rules. Techniques such as n-gram models [Brown et al., 1992] emerged, which predicted words based on the frequency of word sequences seen in training data. Statistical NLP quickly became dominant because it significantly improved performance in practical tasks, such as speech recognition and text classification, by being more robust to linguistic variations and ambiguities.

The rise of deep learning around 2010 marked the next paradigm shift in NLP. Neural network-based models, particularly word embeddings like Word2Vec [Mikolov et al., 2013], captured semantic relationships by mapping words into continuous vector spaces. These approaches allowed NLP models to learn richer semantic information directly from data without explicit feature engineering.

The introduction of the Transformer architecture by Vaswani et al. in 2017 [Vaswani et al., 2023] represented another breakthrough. Transformers leverage self-attention mechanisms, enabling models to efficiently capture context from large text sequences. This innovation led to the development of large-scale pretrained language models such as BERT [Devlin et al., 2019], GPT [Brown et al., 2020], and RoBERTa [Liu et al., 2019], which currently represent the state-of-the-art in numerous NLP tasks, achieving remarkable performance levels across diverse benchmarks.

2.1.2 Approaches in NLP

While it is clear that research in NLP has moved from rule-based to statistical and now to neural network-based approaches, it is important to note that the approaches are not mutually exclusive and can be combined in different ways.

NLP techniques can be primarily categorized into symbolic, statistical, and neural network-based methodologies. Each approach presents distinct advantages and limitations, making them suitable for different scenarios and requirements within NLP applications.

The **symbolic approach** relies on explicitly defined rules and knowledge bases. Its primary strength lies in interpretability and transparency, allowing developers to precisely understand why and how certain outputs are generated. Symbolic methods are often chosen in scenarios where explainability is crucial, such as legal, medical, or regulatory contexts, where decisions must be justified explicitly. However, the symbolic approach typically struggles with scalability and generalization due to the extensive effort required to manually encode rules for each linguistic phenomenon. This approach is thus often confined to highly structured or domain-specific tasks.

The **statistical approach** utilizes probabilistic models derived from large, annotated text corpora. These models learn statistical patterns directly from this data. By learning these patterns, they are applied to tasks such as speech recognition, spam detection, and language modeling, attempting to handle linguistic variations, ambiguous cases, and noisy input like misspellings. Their effectiveness may be limited in low-resource languages or highly specialized domains where sufficient training data might not be available.

The **neural network-based approach**, particularly deep learning, has become dominant. These models learn complex patterns and can generalize effectively across diverse linguistic contexts, largely thanks to transfer learning. Transfer learning allows pre-trained models to be adapted quickly to various NLP tasks with minimal additional training. However, neural network approaches are often criticized for being "black box" models that lack interpretability, making them less desirable in applications where transparency and reliability are mandatory. Additionally, these models are computationally intensive, consuming significant energy resources, which poses sustainability challenges, particularly for large-scale language models (LLMs).

In practice, NLP solutions frequently integrate multiple approaches, leveraging symbolic rules for preprocessing tasks like tokenization or normalization, statistical methods for tasks requiring robustness to variation, and neural networks for tasks needing deep semantic understanding or generalization. Ultimately, the choice of approach involves trade-offs based on specific task requirements, including interpretability, scalability, available resources, and computational efficiency. Hybrid models, which combine symbolic transparency, statistical robustness, and neural generalization capabilities, often provide optimal solutions in real-world NLP applications.

2.2 Relational Tuples

A common format for representing factual information extracted from text is the use of triplets, also known as relational tuples, which typically take the form *(subject, relation, object)*. This representation format offers a structured yet flexible way to extract and store semantic content from natural language text. It is particularly useful for downstream applications such as knowledge base construction, question answering, and information retrieval.

The triplet format allows for a sentence to be segmented into its core semantic components. For instance, the sentence *"A black cat is eating the tuna"* can be abstracted as the triplet *(cat, eat, tuna)*. This simplified representation captures the essential elements of the sentence, "who" is doing "what" to "whom or what", a structure often used for conceptual modeling and logical reasoning [Niklaus et al., 2018].

Triplets may be binary, where each relational unit connects only two arguments (subject and object), or they can be extended to n-ary or nested structures to encode additional information such as time, place, or manner [Bhutani et al., 2016]. The decision to use

binary versus more complex structures typically depends on the task at hand and the expressiveness required in the target representation.

One advantage of the triplet representation is its interpretability and compatibility with symbolic reasoning systems. However, its simplification can sometimes lead to loss of information, especially in sentences involving complex syntactic constructs or implicit arguments. For example, implicit causality, modal verbs, or conditional clauses may not be adequately captured by simple triplets.

Furthermore, the extraction of high-quality triplets requires accurate identification of the semantic roles of entities and the boundaries of meaningful phrases. This is often a non-trivial task, especially in languages with rich morphology and flexible word order, such as German. Nevertheless, due to their simplicity and usability, triplets remain a simple yet effective method for representing structured information from text.

2.3 Open Information Extraction

Information Extraction (IE) refers to the automatic identification of structured information such as entities, relations, and events from unstructured textual sources. Traditional IE methods typically require predefined schemas or domain-specific ontologies, limiting their applicability to particular contexts and constraining their flexibility.

Open Information Extraction (OpenIE), initially proposed by Banko et al. [Etzioni et al., 2008], aims to address these limitations by eliminating the need for pre-specified relations or domain-specific knowledge. OpenIE extracts information, often represented as the relational tuples discussed previously, directly from text in an "open domain" setting. This enables it to process a wide range of texts from diverse domains without prior configuration. Unlike traditional IE methods that generate structured outputs aligned with fixed schemas, OpenIE methods can produce flexible relational tuples capturing the semantic relationships within sentences. The increase in data available on the web, characterized by its diversity, volume, and unstructured nature, has significantly driven the demand for OpenIE techniques.

OpenIE has become instrumental in various applications, including knowledge base population, question-answering systems, summarization, and semantic web applications. For instance, OpenIE can populate knowledge bases by extracting structured facts from large-scale news corpora, enabling automated updates and comprehensive coverage of new information.

Recent symbolic approaches to OpenIE often rely on linguistic analyses and handcrafted rules to identify meaningful relational phrases and arguments. These systems typically leverage syntactic structures to extract semantically coherent tuples, an approach also utilized in this thesis to detect German textual entailment.

Deep learning methods for OpenIE, such as neural OpenIE systems, have emerged in recent literature. Systems like RnnOIE [Stanovsky et al., 2018] employ neural architectures that learn directly from annotated datasets.

In this thesis, we implement OpenIE methods to extract potential correct answer passages from a given student attempt. We don't aim to define entities and relations for the legal domain, but rather use simple patterns from the golden solution, and provide them to the OpenIE system to extract matching patterns in the student's attempt. In line with this goal, we adopt the triplet format discussed in the previous section as a means of representing simplified correct answer content. This allows us to focus on extracting the core concepts in student responses and matching them to patterns derived from gold-standard solutions.

2.3.1 Dependency Parsing

Dependency parsing is a syntactic parsing approach that identifies grammatical relationships between words in a sentence, producing a tree-structured representation known as a dependency tree. In this tree, words are nodes, and the edges represent dependency relations that specify which word depends on which other word and in what way.

For example, in the sentence "The cat is sleeping in the living room," dependency parsing would identify "sleeping" as the main verb (or root), "cat" as its subject, and "room" as the object of the prepositional phrase introduced by "in." Each dependency relation (e.g., nsubj, prep, pobj) adds a layer of meaning that is critical for understanding sentence structure, as illustrated in Figure 2.1.

Dependency grammars are particularly useful in languages with flexible word order, such as German, since dependency relations explicitly define the sentence structure regardless of word sequence. Dependency parsing models are language-specific and are trained using manually annotated linguistic datasets known as "treebanks." Treebanks, developed by linguists, contain sentences annotated with dependency relations, serving as the basis for training parsing models. For example, German treebanks address complexities like morphological inflections, grammatical cases, and highly flexible word order, making them distinct from English treebanks, which handle a different set of linguistic characteristics.

The Universal Dependencies (UD) [Nivre et al., 2016] initiative plays a central role in providing such annotated resources across many languages. UD aims to create a standardized set of syntactic annotation guidelines and multilingual treebanks, promoting cross-linguistic consistency, facilitating cross-lingual learning, and supporting robust multilingual NLP applications.

Modern dependency parsers are often based on transition-based or graph-based algorithms, with many recent systems incorporating deep learning architectures for improved accuracy and generalization.

2.4 Natural Language Inference

NLI (Natural Language Inference) tries to determine the logical relationship between a pair of sentences, named *premise* and *hypothesis*. The premise serves as the context

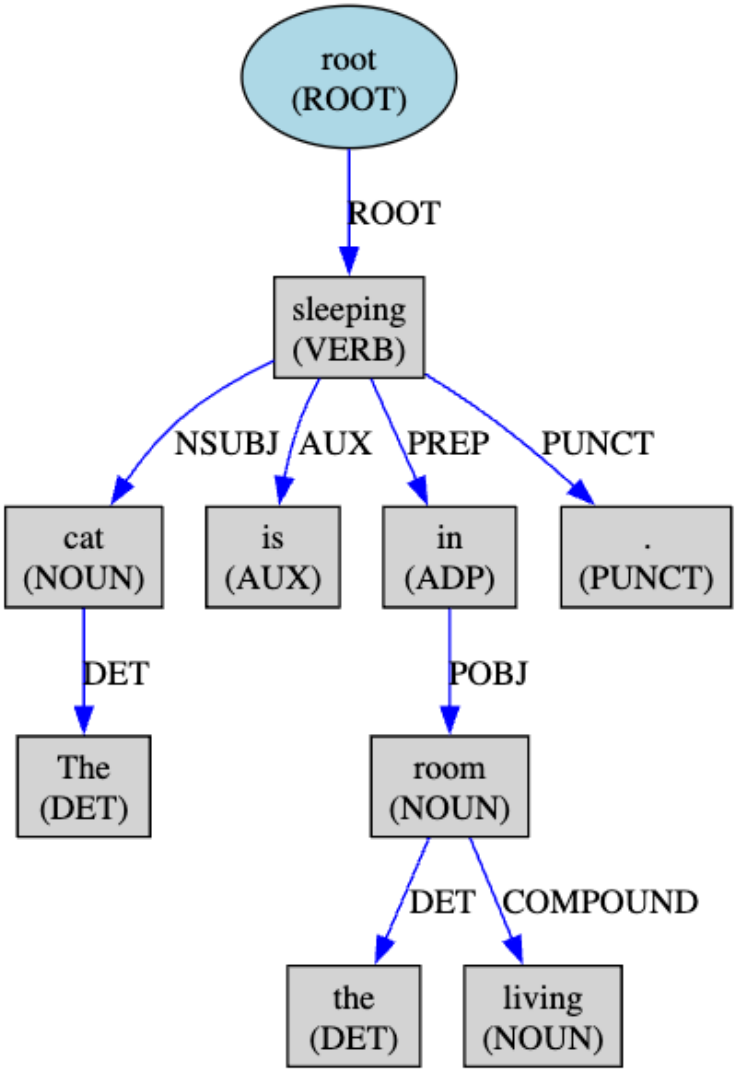


Figure 2.1: Dependency tree for the sentence "The cat is sleeping in the living room".

or the base for an argument, and defines our only knowledge about the world. The hypothesis is a statement that we want to evaluate against the premise.

To illustrate the problem, let's define a simple example with the following premise and hypothesis:

- Premise: A Persian cat is napping on the couch.
- Hypothesis: A domestic animal is resting.

Determining the relationship requires understanding the meaning of the words involved. First, it involves *lexical inference*: recognizing that a "Persian cat" is a specific type of

"domestic animal". Second, it requires knowing that "napping" is a form of "resting". Since both the subject (Persian cat → domestic animal) and the action (napping → resting) in the hypothesis are broader categories or synonyms that encompass those in the premise, we can conclude that the premise entails the hypothesis. This relies on lexical-semantic knowledge, which for machines might come from resources like WordNet [Miller, 1992] or be learned from data.

In general, NLI is a backbone for different tasks that require some type of reasoning, providing a framework for machines to understand and process complex linguistic relationships.

2.4.1 History of NLI

The history of NLI begins with early symbolic approaches. One of the foundational works in this area was the use of logic-based systems, which relied on formal logic and inference rules to determine entailment. These systems, while precise, required extensive manual effort to craft rules and struggled with the variability of natural language.

A significant shift occurred with the introduction of statistical methods. The PASCAL RTE Challenge [Dagan et al., 2005] in 2005 provided a benchmark for evaluating entailment systems, encouraging the development of machine learning approaches. These methods treated entailment as a classification problem, using features extracted from text pairs to train models. However, they often lacked the ability to capture deep semantic relationships.

The advent of deep learning marked a transformative period for NLI. The introduction of a SNLI corpus [Bowman et al., 2015], a large-scale dataset that enabled the training of neural network models. These models, particularly those based on recurrent neural networks and later transformers, significantly improved performance by learning complex patterns in data.

The introduction of transformer-based models, such as BERT [Devlin et al., 2019], revolutionized the field. These models leveraged pre-training on vast corpora to capture language understanding, achieving state-of-the-art results on NLI benchmarks. The development of even larger models, like GPT-3 [Brown et al., 2020], further pushed the boundaries, demonstrating impressive generalization capabilities across diverse linguistic contexts.

Despite these advancements, challenges remain in ensuring robustness and interpretability in NLI systems.

2.4.2 Challenges of NLI

In this section, we highlight some key challenges of NLI, which arise from the complex nature of human language and the need for deep semantic understanding. These challenges are also present in our work on automatic grading of German legal texts, where we encounter them on a practical level.

- **Lexical and Structural Variability**: Often the same meaning can be expressed with different words or syntactic structures. For example, *"The cat chased the mouse"* and *"The mouse was chased by the cat"* are syntactically different but semantically equivalent.

- **Contextual Awareness**: Understanding the context is crucial for accurate interpretation. For instance, a common example used in the literature is the phrase *"Aspirin eliminates headaches"* where "eliminates" suggests a beneficial effect, while *"Aspirin eliminates patients"* implies harm [Levy and Dagan, 2016]. The context in which "eliminates" is used changes the entailment relationship, highlighting the importance of contextual awareness in determining meaning.

- **Distinguishing Contradiction and Neutrality**: One particularly challenging aspect of NLI is distinguishing between *contradiction* and *neutrality*. For example:

  – Premise: A Persian cat is napping on the couch.

  – Hypothesis: A dog is resting.

  Consider the premise about the napping Persian cat and the hypothesis about a resting dog. Does introducing a dog, absent from the premise, constitute a contradiction, or is it merely a neutral statement about an unrelated entity? The premise doesn't support the hypothesis (implying contradiction), yet the hypothesis doesn't directly negate the premise about the cat (implying neutrality). Resolving this ambiguity is a central challenge. NLI systems need clear definitions or learned criteria to reliably distinguish these similar outcomes.

- **Understanding Quantification**: Complex hypotheses involving quantifiers (*all*, *some*), negation, or conjunctions require compositional reasoning. For instance, the premise *"Some cats are asleep"* contradicts *"No cats are asleep"* but is neutral to *"All cats are asleep"*.

These challenges highlight the need for NLI systems to not only perform syntactic analysis but also understand semantics, pragmatics, and possess real-world knowledge.

In real-world applications, the distinction between contradiction and neutrality can be particularly challenging, as it involves deep semantic understanding and contextual reasoning. If the task does not require distinguishing between contradiction and neutrality, NLI can be simplified to *recognizing textual entailment (RTE)*. In RTE, the focus shifts to determining whether the hypothesis is entailed by the premise, reducing the problem to a binary classification task. This simplification can make the task more manageable, allowing NLI systems to focus purely on the presence or absence of entailment.

In this thesis, we adopt the RTE approach to verify the presence of required information in student responses.

2.4.3 The Role of NLI in Information Extraction

While OpenIE can be employed to extract relational tuples from text, it does not verify whether those tuples are factually supported by the source. Extracting information is sometimes not enough; we want to ensure that the information is consistent and factually correct. This is crucial because OpenIE's strength lies in its flexibility to identify and extract diverse relational data. However, without a mechanism to validate this data, there is a risk of retrieving incorrect information. NLI bridges this gap by acting as a validation layer, allowing us to apply it on top of OpenIE to retrieve only the correct information.

In this thesis, we leverage NLI to verify whether answer passages extracted from student responses align with the semantic content of the sample solution. This approach combines the flexibility of OpenIE with the additional validation of NLI, ensuring that extracted answers are not only present but also contextually and logically consistent with the source material.

2.5 AI legal tutor case

To illustrate the application of our methods, we examine a practical business case originating from an Austrian publisher of legal educational materials. The scenario involves legal exercises where students are given cases to solve. In these exercises, students must analyze factual scenarios, apply relevant legal principles, and provide their reasoned judgments with supporting explanations. The ultimate objective for the publisher is to develop an AI-enhanced system capable of offering feedback to law students on their analyses, indicating completeness and identifying missing elements in a transparent manner.

Additionally to the challenges mentioned above for NLI such as lexical and structural variability of sentences, the complexity of legal language necessitates careful attention to word meanings, abbreviations, and synonyms, including Latin terms. To guide the analysis, the use case comes with detailed explanations for the sample solution, highlighting the ideal structure of the solution, which concepts with their synonyms are important and listing some variations of possible correct answers. Key concepts and relations from these guidelines are transformed into relational tuples, which are used to validate the student's attempt.

The core task involves comparing student-submitted attempts against relational tuples extracted from the detailed expert-authored sample solutions and guidelines. Specifically, the system needs to identify whether statements reflecting the required key information are present in a student's text, irrespective of the exact wording used.

For certain legal points, multiple valid applications or justifications might exist (e.g., different ways to demonstrate a legal transaction occurred for consideration). The guideline specifies that recognizing just one valid application in the student's text is sufficient. Furthermore, while some very domain-specific synonyms (like *"Gutgläubigkeit"*

[good faith] and *"Redlichkeit" [honesty]*) might be noted, providing comprehensive lists covering all variations of concepts, including Latin phrases or paragraph references, is unfeasible.

The primary focus of the work presented in this thesis is centered on the information extraction component required for such a feedback system. We aim to develop and evaluate a symbolic approach, specifically leveraging OpenIE and NLI as discussed previously, to detect the presence of the required key information from the sample solution within the student attempts. It is crucial to emphasize that this work does *not* encompass the development of the user-facing feedback interface, the assignment of points, automatic grading, or the generation of qualitative feedback messages for students. The scope is strictly limited to identifying whether the essential semantic content, as defined by the expert solution, is present in the student's answer.

CHAPTER 3

Related Work

This chapter provides a comprehensive overview of related work. Our primary focus is the intersection of open information extraction (OIE) and symbolic NLI, particularly approaches utilizing dependency parsing and relational tuples. Whenever relevant, we emphasize applications within the German legal domain. To provide a broader context, we present current state-of-the-art approaches, which predominantly rely on large language models.

3.1 Open Information Extraction

One of the earliest OIE systems, TextRunner [Yates et al., 2007] defined the task as an unsupervised extraction of relational tuples from large web corpora using a classifier trained on shallow linguistic features. [Wu and Weld, 2010] compared shallow features with dependency parse features for OIE, demonstrating that dependency parsing significantly improved extraction precision and recall, using Wikipedia infoboxes to create a high-quality training corpus.

REVERB [Fader et al., 2011] was developed as a successor to TextRunner, aiming to address incoherent extractions by introducing simple syntactic (POS-based regular expressions) and lexical constraints (relation phrases taking diverse arguments) on verb-based binary relations.

Building upon dependency parsing, Kraken [Akbik and Löser, 2012] employed hand-written rules over typed dependencies to extract complete, N-ary facts, aiming for higher completeness than previous binary-focused systems like ReVerb.

[Mausam et al., 2012] created OLLIE, which improved upon ReVerb by using high-precision ReVerb tuples to bootstrap a pattern learner, enabling the extraction of relations mediated by nouns and adjectives, and including contextual information like belief or conditionality.

[Stanovsky et al., 2016] argued that dependency trees alone might miss certain semantic information and proposed PROPS. This system converts dependency trees using rules into a more semantically oriented graph representation tailored to capture the propositional structure of sentences directly.

While rule-based systems offer explainability, neural approaches emerged, leveraging large datasets and complex architectures. RnnOIE [Stanovsky et al., 2018] framed OIE as a sequence tagging problem to handle challenges like multiple extractions per predicate. An encoder-decoder model (NeuralOIE) [Cui et al., 2018] was developed to generate relation triplets conditioned on the input sentence. These neural methods often lack the transparency required in domains like law.

LILLIE [Smith et al., 2022] presents a hybrid system combining linguistic rules with learning-based methods. This approach aims to leverage the strengths of both paradigms, using learning to refine and improve the quality of triples initially extracted or guided by linguistic principles.

3.2 Current State-of-the-Art in NLI and Information Extraction

Recent advancements in NLI and IE are largely dominated by pre-trained Large Language Models (LLMs). These models demonstrate impressive performance across various benchmarks, often employed in hybrid systems or adapted through prompting and fine-tuning.

For instance, [Sainz et al., 2022] explored few-shot information extraction using LLMs. Their approach involves prompting pre-trained models and fine-tuning them for textual entailment tasks, showcasing the potential of LLMs to perform IE with minimal task-specific annotations by leveraging their broad linguistic knowledge.

Similarly, hybrid approaches combine the strengths of structured knowledge and LLM reasoning. [Boer et al., 2024] proposed a method for question answering that uses knowledge graphs for initial triplet-based prefiltering, followed by LLM-based ranking and reranking to refine answers. This combination leverages structured data for efficiency and LLM capabilities for deeper understanding, achieving strong results.

However, while powerful, these LLM-based approaches often function as black boxes, limiting their interpretability and reliability in contexts demanding explicit reasoning and transparency, such as the legal domain central to this thesis.

3.3 Domain-Specific Challenges and Approaches

Applying NLI and IE techniques effectively can be complicated when dealing with specific languages and specialized domains, which often have unique challenges and requirements.

3.3.1 NLP for German

German presents particular challenges for NLP due to its flexible word order and rich morphology compared to English. Several works illustrate efforts to address these. For instance, [Neumann and Xu, 2003] focused on mining answers from German web pages using techniques adapted to German web content. PropsDE [Falke et al., 2016] involved adapting an English rule-based OIE system (PropS) to German, demonstrating feasibility but also the effort required for cross-lingual rule porting based on dependency parses. GerIE [Bassa et al., 2018] was designed specifically for German OIE, employing handcrafted rules over dependency parses tailored to German linguistic phenomena. More recently, [Engelbach et al., 2023] applied fine-tuned question-answering models to German legal documents, incorporating rule-based validation to check the neural model's output, highlighting work at the intersection of language and domain.

3.3.2 NLP for the Legal Domain

The legal domain introduces its own difficulties, including specialized terminology, complex sentence structures, and requirements for high precision and justification. Early work in the German legal domain, such as [Walter and Pinkal, 2006], used rule-based methods to extract definitions from court decisions. Datasets like GerDaLIR [Wrzalik and Krechel, 2021] provide resources, though often for specific tasks like citation-based retrieval rather than the semantic interpretation needed for NLI/OIE. These examples show the focus on developing methods, including rule-based, neural, and hybrid systems, to address the particular requirements of processing legal texts where interpretability and precision are often important considerations.

3.4 Symbolic NLI

Addressing requirements for explainability, particularly in specialized domains like law and for morphologically rich languages like German, symbolic NLI approaches model entailment through structured reasoning based on linguistic rules and logical frameworks.

[Angeli et al., 2015] utilized natural logic inference within an OIE framework. Their system focuses on selecting maximally specific candidate triples by reasoning about lexical relationships (like hyponymy and hypernymy) and clause structures, aiming to improve the precision of open-domain extraction through logical validation.

The Hy-NLI framework [Kalouli et al., 2020] includes GKR4NLI, a symbolic engine using natural logic that represents sentences as semantic graphs [Kalouli and Crouch, 2018]. GKR4NLI evaluates truth preservation under lexical substitutions based on specificity and monotonicity principles derived from natural logic. This provides explainable inferences, particularly effective for linguistically complex cases. While the full Hy-NLI framework combines this symbolic engine with LLMs for hybrid decision-making in English, the underlying GKR4NLI graph representation and natural logic reasoning provide a strong

foundation for explainable, linguistically-grounded NLI, which inspires the approach taken in this thesis for German legal text.

CHAPTER 4

Methodology

The goal of this chapter is to explain the components used for detecting entailment. First, we introduce our adapted graphical knowledge representation, which can be seen as a semantically rich graph with multiple sublayers. Then, we describe the multi-stage graph-based matching process that uses the graphical knowledge representations from the premise and hypothesis to detect entailment.

4.1 Graphical Knowledge Representation

As discussed earlier in Section 2.3.1, dependency parsing helps identify the grammatical structure of a sentence. However, relying only on dependency relations is often insufficient for recognizing textual entailment. Dependency trees show the direct links between words but may not capture enough information about word meanings, variations in phrasing, or context, which are important for deciding entailment.

To address these limitations and allow for a better understanding of sentence meaning, we propose employing a graphical knowledge representation. This approach uses the dependency parse tree as a base structure. We then enrich parts of this structure by adding extra layers of information.

Specifically, we enrich the graph by incorporating base forms identified using a morphological analyzer, lexical-semantic details such as synonyms, hypernyms, and hyponyms derived from external language resources, and contextual information such as negation. An overview of the graphical knowledge representation is shown in Figure 4.1.

The goal is to create a more detailed graph that helps our system compare the meaning of sentences more effectively, even when they are phrased differently. This allows the system to better check if one statement follows logically from another (entailment) or contradicts it.

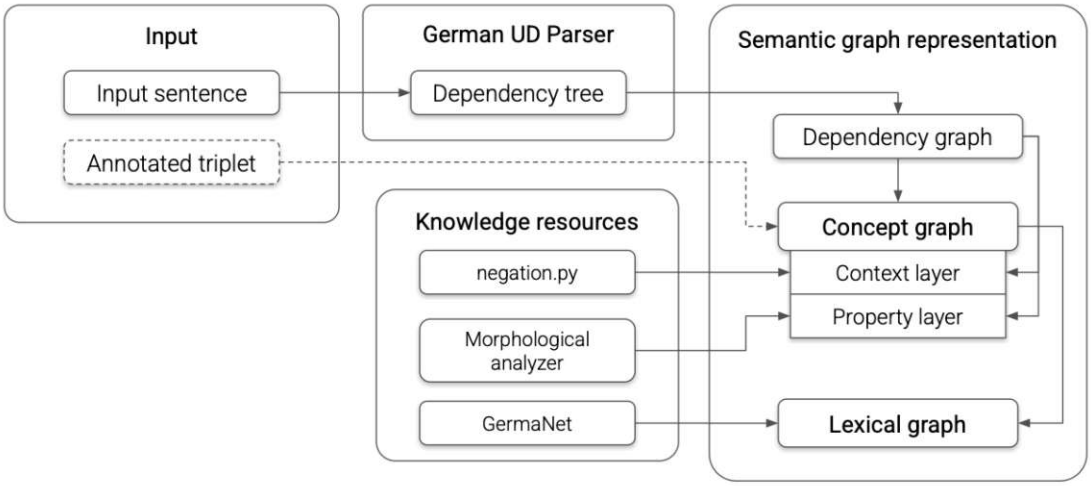


Figure 4.1: Overview of our adapted graphical knowledge representation for German inspired by GKR [Kalouli and Crouch, 2018].

Our method is inspired by similar graph-based approaches for NLI, such as the Graphical Knowledge Representation (GKR) developed for English [Kalouli and Crouch, 2018, Kalouli et al., 2020]. We adapt the core idea of GKR to create a related, but simpler, graphical knowledge representation specifically for German. Unlike GKR, which incorporates coreference resolution within the graph building process, we experiment with applying coreference resolution as a separate preprocessing step before parsing the sentences. This choice is explored further in Chapter 5.

4.1.1 Dependency Parsing

Dependency parsing constitutes the foundational step in our methodology. The resulting dependency graph for each sentence, derived from both the sample solution and the student attempt, serves as the structural backbone upon which the graphical knowledge representation is built.

4.1.2 Concept layer

The concept layer serves as an intermediate representation, designed to distill the complex syntactic structure of a sentence's dependency graph into its core semantic essence. Its primary purpose is to simplify the sentence representation down to the minimal information required for the entailment task, focusing on the key entities (concepts) and the relationships between them.

In the context of our AI legal tutor use case, the dataset provides sample solutions containing both full sentences and corresponding relational tuples (e.g., *prüfen(gutgläubig, Erwerb) [examine(bona_fide, acquisition)]* with the sentence *"Zu prüfen ist der gutgläubige Erwerb"[Bona fide acquisition is to be examined]*). These tuples represent the key legal

concepts or facts that must be present in a student's answer for it to be considered correct. They function as the target patterns for our entailment detection process.

Figure 4.2 illustrates this process, showing a concept graph derived from the annotated relational tuple *prüfen(gutgläubig, Erwerb)* and its corresponding source sentence "Zu prüfen ist der gutgläubige Erwerb" [*Bona fide acquisition is to be examined*]. The graph clearly shows the core concepts (*prüfen*, *Erwerb*, *gutgläubig*) and their dependency links as extracted from the sentence, representing the essential meaning defined by the annotation.

Figure 4.2: Example of a concept graph extracted from the annotation *prüfen(gutgläubig, Erwerb)* and the sentence *"Zu prüfen ist der gutgläubige Erwerb"* [*Bona fide acquisition is to be examined*]. Nodes represent concepts (words) and edges represent dependency relations.

This simplification significantly reduces the number of nodes and edges compared to the full dependency graph and aligns well with the structure of relational tuples used for matching.

A key advantage of this approach is its flexibility. The concept graph is not restricted to representing only actions and their participants. It can effectively represent simpler conceptual links, such as the modification of a noun by an adjective (e.g., representing *"gutgläubiger Erwerb" [bona fide acquisition]* directly) without the presence of a predicate. This allows us to capture a wider range of semantic patterns relevant to the legal domain.

For this work, we use the provided annotations from the sample solutions to define the scope of the concept graph. This ensures the graph precisely reflects the target pattern specified in the sample solution. This premise concept graph then serves as the foundation upon which subsequent layers (lexical, property, context) are built.

4.1.3 Lexical layer

Building upon the concept graph, the lexical graph enriches the representation by incorporating semantic information for each concept node. This layer leverages external lexical resources to expand the potential matches beyond exact word forms, enabling a more detailed understanding of semantic relatedness. The idea behind this additional layer is to allow the system to match more flexible entailment patterns, and as a consequence, increase recall. Similarly to the GKR [Kalouli and Crouch, 2018], where different sources for English are used to build the lexical layer, we adapt this idea for German.

Figure 4.3 provides a visualization of the lexical graph built upon the concept graph example from Figure 4.2.

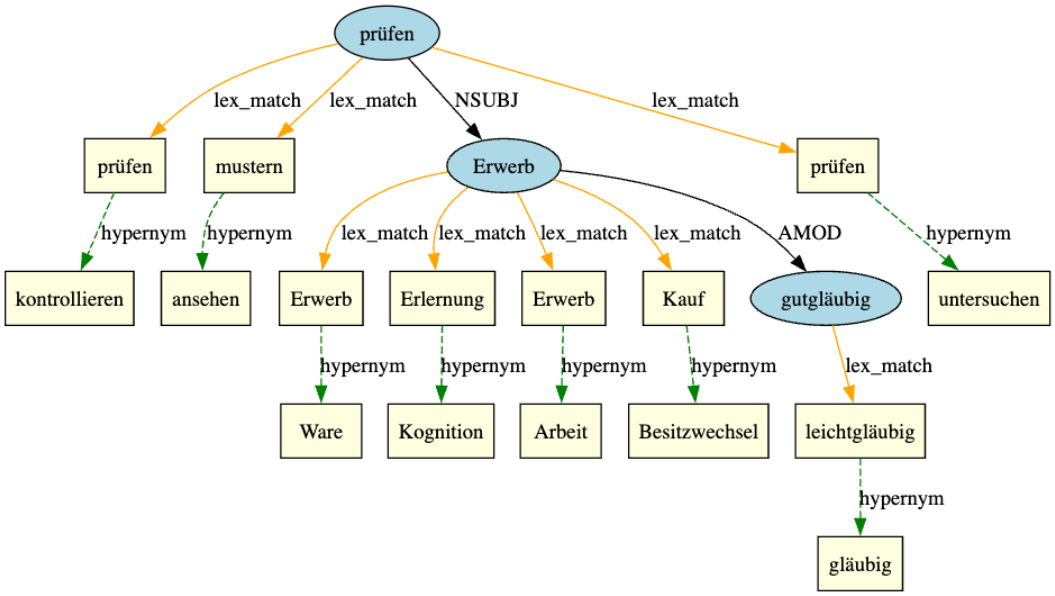


Figure 4.3: Lexical graph on top of the concept graph

In this figure, which corresponds to the sentence *"Zu prüfen ist der gutgläubige Erwerb"* [*Bona fide acquisition is to be examined*], new lexical nodes (yellow boxes) and edges representing lexical relations (*lex_match*, *hypernym*) are shown connected to the original concept nodes (blue ellipses). For clarity, the visualization limits the number of hypernyms displayed per node (here, a maximum of 2) and omits hyponyms entirely. The depth of hypernym/hyponym relations explored in GermaNet is a configurable parameter in our system, defaulting to a depth of 2.

The primary source for this lexical information is GermaNet [Hamp and Feldweg, 1997], a large lexical-semantic network for the German language developed at the University of Tübingen. Similar in structure and purpose to the English WordNet [Miller, 1992], GermaNet organizes German nouns, verbs, and adjectives into sets of synonyms called "synsets". It establishes various semantic relationships between these synsets, such as hypernymy (more general term) and hyponymy (more specific term), effectively functioning as both a comprehensive thesaurus and a lightweight ontology.

For each concept node derived from a token in the sentence, we query GermaNet to retrieve related lexical units. Specifically, we add nodes representing:

- **Synonyms:** Words with the same or very similar meaning (e.g., *"Auto"* and *"Wagen"*, both terms for a vehicle). These directly support entailment: if the premise uses *"Wagen"* and the hypothesis uses *"Auto"*, the synonym link allows the system to recognize them as equivalent concepts.
- **Hypernyms:** Words representing a broader category (e.g., *"Tier"* [*animal*] is a hypernym of *"Katze"* [*cat*]). These enable hierarchical inference: if the premise

states *"Die Katze schläft"* [*The cat sleeps*] and the hypothesis is *"Ein Tier schläft"* [*An animal sleeps*], the hypernym link (*Katze* → *Tier*) validates the entailment, as a cat is a type of animal.

- **Hyponyms:** Words representing a more specific instance of a category (e.g., *"Katze"* [*cat*] is a hyponym of *"Tier"* [*animal*]). These present a more complex case. Standard logical entailment does not typically flow from a general term to a specific one (e.g., *"Ein Tier schläft"* [*An animal sleeps*] does not strictly entail *"Die Katze schläft"* [*The cat sleeps*]). However, including hyponym relations allows for exploring more flexible matching scenarios. In certain contexts, particularly in information retrieval or question answering, identifying a specific instance (hyponym) mentioned in the hypothesis that falls under a general concept in the premise might be relevant. We include hyponyms experimentally to assess their potential benefit in capturing such looser forms of relatedness, acknowledging that they do not represent strict logical entailment.

These related terms are linked to the original concept node via specific edge types (e.g., *synonym*, *hypernym*, *hyponym*). Since a single word can have multiple meanings (senses), and thus multiple entries in GermaNet, we initially add lexical information corresponding to all possible senses associated with the concept node's lemma. The crucial step of Word Sense Disambiguation (WSD) is deferred to the matching stage (detailed in Section 4.2.1), where the hypothesis graph helps select the most appropriate sense.

4.1.4 Property layer

While the concept graph captures the core relational structure and the lexical layer adds semantic information, the property layer focuses on enriching each concept node with detailed linguistic features derived from the original sentence. This layer provides crucial morphological and syntactic information that refines the representation of each concept.

In our adaption of the GKR architecture [Kalouli and Crouch, 2018], we include properties for each concept using a combination of the spaCy NLP library [Honnibal et al., 2020] for general linguistic processing and, the DWDSmor component [Klein and Geyken, 2010] for in-depth German morphological analysis.

Figure 4.4 illustrates this layer by showing the concept nodes from the previous example (*"prüfen"*, *"gutgläubig"*, *"Erwerb"*) annotated with tables displaying their extracted linguistic properties. These properties are stored as attributes within the data associated with each node in the graph representation.

Each concept node is enriched with essential linguistic properties derived from the corresponding token, including its base form (lemma) and grammatical role (part-of-speech). Additionally, detailed morphological features crucial for German, such as case, number, and gender, are extracted to provide a richer grammatical description for each concept.

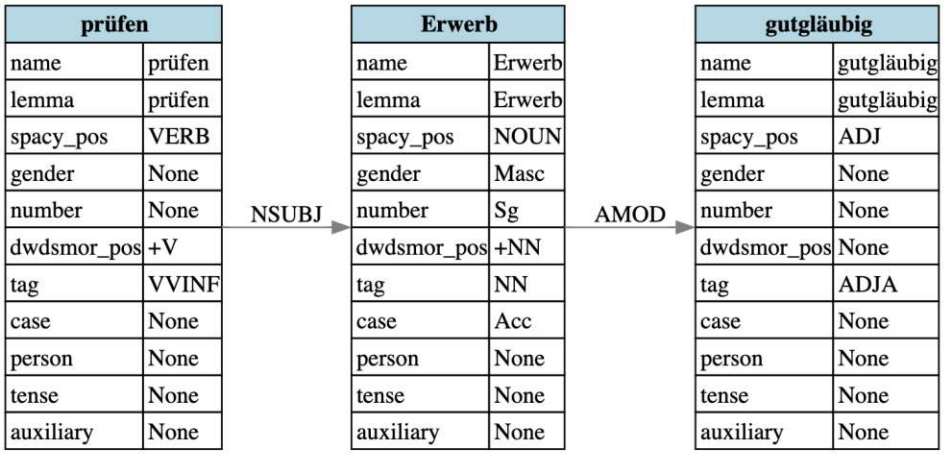


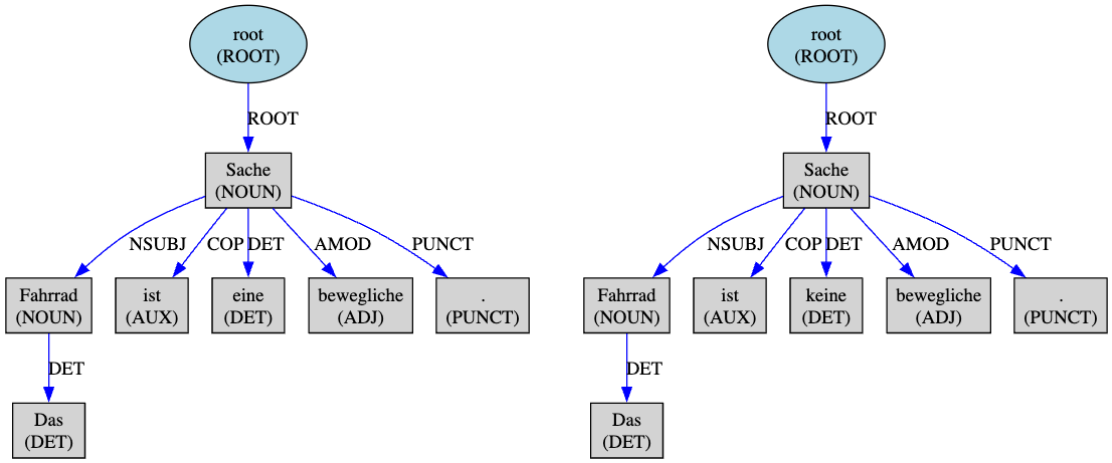
Figure 4.4: Example of the property layer. Concept nodes from Figure 4.2 are shown with their associated linguistic properties (lemma, POS, gender, number, case, etc.) extracted using spaCy and DWDSmor.

These properties add layers of specificity and constraint to the concept nodes. While not every mentioned property will be explicitly applied in the symbolic NLI system presented in this thesis, we still include this rich information to provide a robust graph representation. For future work and use case task specific different properties can be selected and applied to further enhance the system's accuracy.

4.1.5 Context layer

Until now, there is no information about the existence of the "concept" of the sentence. The dependency graphs for negated and non-negated sentences are structurally identical, as visible in Figure 4.5. The context layer should introduce the contexts, indicating whether the concepts and their relations have been instantiated or not, specifically focusing on negation.

Contrary to the GKR representation where different contexts such as implicatures, negation, disjunctions are detected and a separate graph is built, we focus solely on negation detection. However, simply identifying negation words is insufficient. Accurate NLI requires understanding precisely *what* is being negated (the scope of negation). Failing to determine the correct scope can lead to critical errors, such as incorrectly concluding entailment when a key concept is negated in the hypothesis but not the premise, or vice-versa.



(a) Dependency graph for the positive sentence "Das Fahrrad ist eine bewegliche Sache" (The bicycle is a movable object).

(b) Dependency graph for the negated sentence "Das Fahrrad ist keine bewegliche Sache" (The bicycle is not a movable object).

Figure 4.5: Comparison of concept graphs for a positive sentence (left) and its negated counterpart (right). The underlying structure is identical before employing the negation detection.

The necessity for differentiating negation types becomes clear in complex sentences. Consider this sentence from our use case:

(1) Jedoch liegt kein gültiger Titel vor, da Paula nur Sachbesitzerin, nicht
However lies no valid title forth, as Paula only physical.owner, not
Rechtbesitzerin ist und daher nicht über das Fahrrad verfügt dürfte
legal.owner is and therefore not over the bicycle disposes should
[However, there is no valid title, as Paula is only the physical owner, not the legal owner, and therefore should not be allowed to dispose of the bicycle.]

In this single sentence, multiple concepts and relations exist, some negated and some affirmed:

- "gültiger Titel" [valid title] is negated by "kein".
- "Rechtbesitzerin" [legal owner] is negated by "nicht".
- "über das Fahrrad verfügt dürfte" [should dispose of the bicycle] is negated by "nicht".
- "Sachbesitzerin" [physical owner] is **not** negated.

This example highlights the importance of accurately determining the scope. A naive approach might incorrectly negate "Sachbesitzerin" or fail to negate "gültiger Titel".

To address this challenge, we developed a rule-based negation detection system that analyzes the dependency parse tree. Our approach is inspired by [Carrillo de Albornoz et al., 2012], where the scope of negation is solved using dependency parsed trees and WordNet. We adapted some ideas for the German language. Our system operates in two passes:

1. Cue Identification: The first pass identifies potential negation cues within the sentence. It uses a pre-defined expanded lexicon (*NEGATION_LEXICON*) containing various German negation words categorized by their function (particles like *nicht*, noun phrase negators like *kein*, prepositions like *ohne*, negating verbs like *verneinen*, and conjunctions like *weder*). Crucially, this pass also employs a list of *FALSE_NEGATION_PATTERNS* (e.g., *"nicht nur"* - not only, *"kein anderer"* - no other) to filter out phrases where negation words appear but do not actually negate the surrounding context. Only tokens identified as true negation cues proceed to the next step.

2. Scope Determination: The second pass determines the scope for each identified true negation cue based on its type and its position in the dependency tree. Different rules apply depending on the cue type:

- **Particle Negation (e.g., *nicht*, *nie*):** These typically modify a specific word (their syntactic head).
  - The primary target of negation is the head word itself (often a verb or adjective). Example: In *"Der Erwerb ist nicht gutgläubig"*, *nicht* negates *gutgläubig*.
  - *Subject Expansion:* If the negated head is a verb, its subject is also included in the scope. Example: In *"Paula hat das Fahrrad nicht gekauft"*, both *gekauft* and *Paula* are marked as negated.
  - *Boundary Limitation:* This right-side expansion is stopped if it encounters a subordinate clause introduced by conjunctions listed in *SUBORDINATE_BOUNDARIES* (e.g., "weil", "obwohl"). This prevents negation from incorrectly extending into clauses expressing cause, condition, etc.
- **Noun Phrase Negation (e.g., *kein*, *keine*):** These negate an entire noun phrase.
  - The algorithm identifies the head noun governed by the negator (e.g., "Titel" in "kein gültiger Titel").
  - It then traverses the dependency subtree starting from this head noun (including adjectives like "gültiger" or prepositional phrases attached to the noun).
  - *Boundary Limitation:* Similar to particle negation, this traversal stops if it encounters a subordinate clause boundary.
- **Prepositional Negation (e.g., *ohne*, *außer*):** These negate the noun phrase(s) that function as the object of the preposition. Example: In *"Er kam ohne einen Mantel" [He came without a coat]*, *ohne* negates *einen Mantel*.

- **Verb Negation:** Certain verbs inherently carry negative meaning, such as *verneinen, ablehnen, bestreiten, leugnen, verweigern, ausschließen,* or *vermeiden [deny, reject, dispute, deny, refuse, exclude, avoid].* Our system identifies these verbs using a predefined lexicon (*NEGATION_LEXICON*). When such a verb is detected, the system identifies the main grammatical roles connected to the negating verb, specifically its subject (the entity performing the action, or being described in passive sentences) and its objects or complements (entities receiving the action or completing the verb's meaning). The system then marks these core participants as being negated by the verb. This enables us to distinguish whether a student correctly affirms a concept or incorrectly negates it through their choice of verb. For example, in a sentence like *'Der gutgläubige Erwerb ist zu verweigern' [Acquisition in good faith is to be denied]*, the presence of the negating verb *verweigern* influences how the related concepts are interpreted in terms of affirmation or negation.

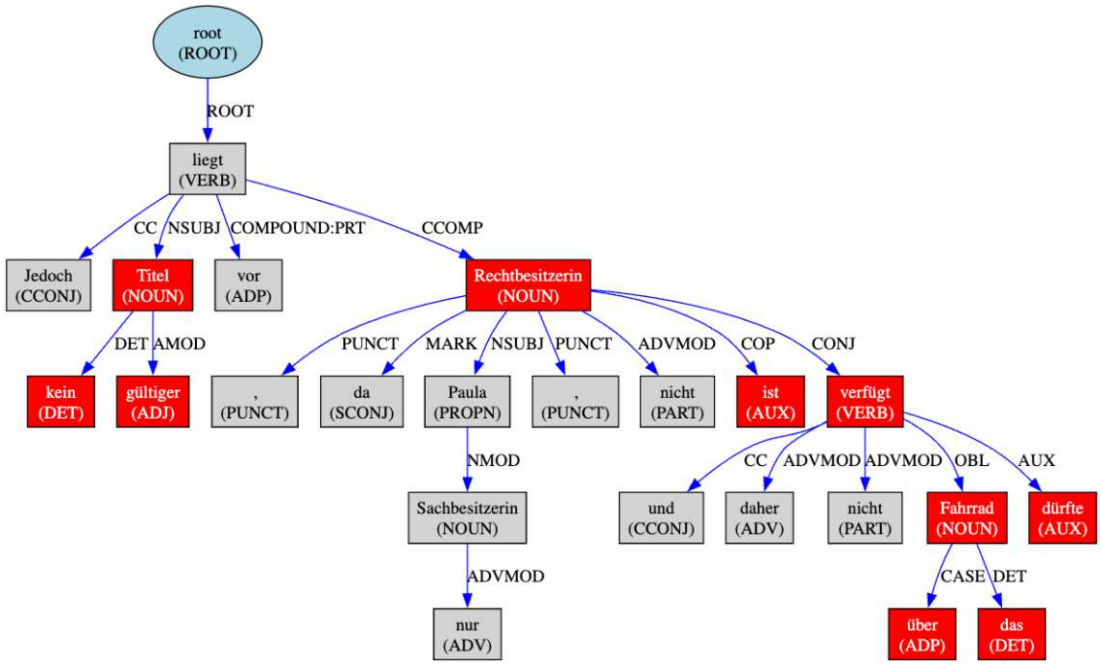


Figure 4.6: Negation scopes detected for the complex sentence. Negated concepts are marked with a red background. Note how different rules apply to 'kein' and the two instances of 'nicht'.

Figure 4.6 visually demonstrates the outcome of our scope determination rules applied to the complex example sentence *"Jedoch liegt kein gültiger Titel vor, da Paula nur Sachbesitzerin, nicht Rechtbesitzerin ist und daher nicht über das Fahrrad verfügt dürfte"*. In the following, we explain how the system arrives at this specific negation marking.

- **kein gültiger Titel:** The system first identifies *kein* as a noun phrase negator.

Following the rules for this type, it finds the head noun governed by *kein*, which is *Titel*. It then marks the subtree rooted at *Titel*, including the modifier *gültiger*, as negated. This corresponds to the red highlighting of *kein*, *gültiger*, and *Titel* in the figure.

- **nicht Rechtbesitzerin:** The first instance of *nicht* is identified as a particle negator. Its syntactic head in the dependency tree is *Rechtbesitzerin*. According to the particle negation rule, the head itself is the primary target, so *Rechtbesitzerin* is marked as negated (shown in red). Notably, *Sachbesitzerin* is correctly left unaffirmed, as it is not the head of this specific *nicht*.

- **nicht über das Fahrrad verfügt dürfte:** The second *nicht* is also a particle negator, and its head is the main verb *verfügt*. The rule dictates negating the head verb (*verfügt*) and its associated auxiliary verb (*dürfte*). Furthermore, the subject expansion rule identifies *Paula* (the subject of *verfügt*) and includes it in the negation scope. The rule also extends the scope to other dependents connected to the verb, such as the prepositional phrase *über das Fahrrad*. The figure highlights *nicht*, *verfügt*, *dürfte*, *über*, *das*, and *Fahrrad* in red, reflecting this determined scope.

This detailed, context-aware approach allows the system to accurately represent which parts of the sentence meaning are affirmed and which are negated, forming a critical input for the subsequent entailment detection process.

4.2 Entailment Detection

To illustrate the entailment detection process described in the following subsections, we will use a running example. Let the premise pattern be derived from the sentence *"Zu prüfen ist der gutgläubige Erwerb" [Bona fide acquisition is to be examined]*, represented by the relational tuple *prüfen(gutgläubig,Erwerb)*. We will primarily consider the hypothesis sentence *"Der gutgläubige Kauf ist zu überprüfen" [The bona fide purchase is to be reviewed/checked]* to demonstrate the matching steps. Additional hypothesis variations will be introduced to clarify specific concepts like structural validation modes.

Following the construction of the multi-layered graphical knowledge representations for both the premise (derived from the gold-standard relational tuple and its source sentence) and the hypothesis (each sentence from the student's attempt), the core task is to determine if the meaning expressed in the hypothesis entails the meaning required by the premise. Our system approaches this as a graph matching problem, specifically checking if the conceptual structure defined in the premise pattern graph can be found within the hypothesis sentence graph, considering semantic variations and logical consistency.

The process is designed to identify entailment, focusing on whether the student's answer contains the necessary information as defined by the premise pattern. It does not attempt to classify the relationship into the full NLI classes (entailment, contradiction, neutral).

Instead, it performs a targeted search for positive evidence of entailment, incorporating validation steps to avoid simple contradictions, particularly those involving negation.

The detection process is split into several stages: initial predicate matching, subsequent argument matching to ensure all required arguments are present, and finally, validation checks focusing on structural consistency and negation agreement. Throughout this process, detailed information about the nature of the matches is recorded to provide explainability for the final entailment decision.

4.2.1 Initial matching

The first step aims to establish potential alignment points between the premise pattern graph and the hypothesis sentence graph.

The system takes the graphical knowledge representation representing the premise pattern and compares its core concept node (identified as the predicate during annotation) against all nodes in the dependency graph of a hypothesis sentence. The matching can be performed based on different criteria, allowing for experimental flexibility:

- **Lexical Meaning (Lemma):** The system can match based on the fundamental dictionary form (lemma) of the words, accessed via the property layer of the graphs. This allows matching *"kaufen" [to buy]* with *"kauft" [buys]*.
- **Lexical Relations:** If lemma matching is used for arguments, GermaNet relations (synonyms, hypernyms, hyponyms) can be optionally considered, as configured. Instead of relying only on strict lexical units provided within GermaNet *synsets*, we employ a path-based similarity measure to capture closely related terms. This approach addresses challenges like words having multiple senses (synsets) and the need to identify near-synonyms or co-hyponyms not listed in the same synset. The measure calculates the path distance between synsets via the hypernymy relation. This distance is then normalized to produce a similarity score between 0 and 1. For instance, consider the verbs *"prüfen"* [examine] and *"überprüfen"* [review/check]. They do not appear in the same synset, and neither is a direct hypernym of the other. Relying only on direct links would thus disregard their clear semantic relatedness, even though GermaNet's own definition for *"überprüfen"* acknowledges it is often identical to *"prüfen"* (*"v. häufiger: völlig identisch mit prüfen..."*). Our path-based measure bridges this gap: these two verbs share the common hypernym *"kontrollieren"* [control/check], resulting in a short path (length 2: prüfen → kontrollieren ← überprüfen) and thus a high similarity score (e.g., >0.9). This similarity threshold is configurable; we use a default of 0.9 to prioritize high-confidence matches, assuming terms scoring this high are likely synonyms or very closely related concepts, thereby maintaining explainability. Lowering the threshold could increase recall but potentially match less related terms. When multiple senses exist for a lemma, the system compares all potential sense pairings within the same semantic field and selects the one yielding the highest similarity

score. In our running example, the premise predicate *prüfen* would be matched with the hypothesis node *überprüfen* using this lexical relation approach, yielding a high similarity score and establishing a potential alignment.

The choice between purely lemma-based or a combination of lemma and the specific utilization of lexical relations, represents a configurable aspect of our methodology.

As a final step in the initial matching stage, the negation status from the context layer is considered if the corresponding experimental setting is enabled (*use_negation=True*, see Section 5.5). If enabled, a potential match is only forwarded to the argument matching stage if the premise concept and the candidate hypothesis concept share the same negation status (both affirmed or both negated). For example, if our hypothesis sentence were *"Der gutgläubige Kauf ist **nicht** zu überprüfen"*, the node *überprüfen* would be marked as negated by the context layer. Since the premise predicate *prüfen* is affirmed, this potential match would be discarded at this stage due to the negation mismatch.

The output of this stage is a set of potential matches, where each match links the premise predicate node to a specific node in the hypothesis graph, along with metadata indicating how the match was achieved (e.g., *equals_lemma*, *equals_synonyms*, *subclass*, *hyponym*).

4.2.2 Argument matching

Once a potential predicate match is established (like *prüfen* → *überprüfen* in our example), the system proceeds to the argument matching stage. The goal here is to verify that all the arguments associated with the predicate in the premise pattern graph (*Erwerb* and *gutgläubig*) also have corresponding, compatible matches in the hypothesis sentence graph (*"Der gutgläubige Kauf ist zu überprüfen"*), relative to the initial predicate match.

For a given predicate match (linking a premise predicate to a hypothesis node), the system identifies all argument nodes connected to the predicate in the premise graph's concept layer. Then, for each premise argument node, it searches the hypothesis graph for a suitable matching node. In our running example, the system needs to find matches for *Erwerb* and *gutgläubig* in the hypothesis sentence. *gutgläubig* can be matched directly to *gutgläubig* based on lemma equality. Matching *Erwerb* to *Kauf* would require using lexical relations, assuming they are considered synonyms or closely related by the path-based similarity measure exceeding the configured threshold.

Similar to the initial matching, the criteria for matching arguments can be based on lemma or POS, depending on the experimental configuration (Section 5.5). In addition, further constraints from the property layer can be optionally enforced during argument matching:

- **Lexical Relations:** If lemma matching is used for arguments, GermaNet relations (synonyms, etc.) can be optionally considered, as configured.

- **Morphological Features:** Checks for agreement in grammatical gender or number can be applied if enabled in the configuration, particularly relevant for German nouns and adjectives.

- **Negation Status:** If negation handling is enabled (*use_negation*), the negation status of a premise argument and its potential hypothesis match must be identical. In our example, both *Erwerb* and *gutgläubig* in the premise are affirmed. Therefore, their matches in the hypothesis (*Kauf* and *gutgläubig*) must also be affirmed.

A critical requirement of this stage is that *every single* argument node present in the premise pattern graph must find a valid and compatible match in the hypothesis graph according to the active configuration. If even one premise argument cannot be successfully aligned (e.g., if *Kauf* was not considered similar enough to *Erwerb*, or if *gutgläubig* was missing from the hypothesis), this entire potential entailment path (stemming from the initial predicate match) is considered invalid, and the system may proceed to check other potential predicate matches if available.

Successfully matching an argument involves finding the candidate node in the hypothesis sentence graph that satisfies the configured criteria and constraints (negation, gender, number if active), while also ensuring that a single hypothesis node is not matched to multiple distinct premise nodes (arguments or predicate) within the same match attempt.

4.2.3 Validation

The final stage involves validating the complete match (predicate and all arguments) found through the preceding steps. This validation focuses on structural consistency between the matched nodes in the premise and hypothesis graphs and reinforces the check against logical contradictions involving negation.

Structural Consistency Check: After successfully aligning the predicate and all its arguments, the system can *optionally* perform a structural check to ensure the relationships *between* these aligned concepts are preserved. This check examines the dependency edges connecting the concept nodes within the premise graph and verifies their correspondence in the hypothesis graph. The methodology supports three different levels of structural strictness, configured via the *edge_check_mode* parameter:

- **Exact dependency label:** This is the strictest mode. For every dependency edge connecting two matched concept nodes in the premise graph (e.g., an object relation edge from *prüfen* to *Erwerb* and a modifier edge from *Erwerb* to *gutgläubig*), it checks if an edge with the *exact same dependency label* exists between the corresponding hypothesis nodes (*überprüfen*, *Kauf*, and *gutgläubig*). In our first hypothesis sentence, *"Der gutgläubige Kauf ist zu überprüfen"*, this check would likely pass, assuming a similar grammatical structure.

- **Path exists:** This is a more relaxed mode. For every dependency edge between two matched concept nodes in the premise graph, it only checks if *any* directed path exists between the corresponding hypothesis nodes in the hypothesis graph. Consider the hypothesis *"Beim Verkauf des Fahrrades von Paula an Fanny, ist zu überprüfen, ob ein gutgläubiger Erwerb durch Fanny in Frage kommt" ["When selling the bicycle from Paula to Fanny, it must be checked whether a bona fide acquisition by Fanny is possible"]*. The system might match *prüfen* → *überprüfen*, *Erwerb* → *Erwerb*, and *gutgläubig* → *gutgläubiger*. Although the direct dependency links might differ (e.g., *Erwerb* might be part of a subordinate clause governed by *überprüfen*), this mode verifies that some path exists between *überprüfen* and *Erwerb*, and between *Erwerb* and *gutgläubiger*, confirming connectivity without enforcing the specific original grammatical relation.

- **Open:** This mode skips the structural validation entirely, relying only on the successful matching of the individual predicate and argument nodes within the sentence. For example, given the hypothesis *"Der derivative Erwerb sowie der gutgläubige Verkauf sind zu überprüfen" ["The derivative acquisition and bona fide sale must be reviewed"]*, this mode might match *prüfen* → *überprüfen*, *Erwerb* → *Erwerb* (the first instance), and *gutgläubig* → *gutgläubige* (modifying *Verkauf*). It ignores the fact that *Erwerb* and *gutgläubige* are not directly related in the hypothesis structure as they were in the premise. While maximizing recall, this can lead to logically incorrect entailments.

If a structural check is enabled (i.e., mode is not *Open*) and fails according to the selected mode, the match is invalidated.

Negation Consistency: While potentially checked during individual node matching (if *use_negation* is enabled), the validation stage implicitly relies on the consistent application of negation status. The requirement that matched predicate and argument nodes must share the same negation status (both affirmed or both negated according to the context layer) is fundamental to avoiding simple contradictions and correctly identifying entailment, leading to more robust and precise extracted entailments.

Final Entailment Decision: The system concludes that the hypothesis sentence entails the premise pattern if and only if:

1. A valid initial predicate match is found between the premise pattern and the hypothesis sentence (using the configured method: primarily lemma in our experiments).

2. All arguments defined in the premise pattern find corresponding valid matches in the hypothesis sentence (using the configured method: lemma or POS, and satisfying active constraints like negation, gender, number).

3. The configured structural consistency check (if not *none*) passes according to the selected *edge_check_mode*.

If these conditions are met for at least one premise pattern within a given gold-standard section, and for at least one sentence in the student's attempt, the system predicts entailment for that section.

Explainability Output: For each successful entailment found, the system stores a detailed record of the match. Listing 4.1 shows an example of such an output for the match found between our premise pattern *prüfen(gutgläubig,Erwerb)* and the hypothesis sentence *"Beim Verkauf ... in Frage kommt."* (used previously to illustrate the *Path exists* mode).

This record includes:

- The specific premise pattern (*pattern*) and the hypothesis sentence involved.
- The mapping between premise concept nodes (*premise_node_id*) and hypothesis nodes (*hypothesis_node_id*), including the path-based similarity score if GermaNet relations are used (*similarity_score*). This is shown in *predicate_matches* and *argument_matches*.
- The type of match achieved for each node (e.g., *equals_lemma*, *equals_synonyms*, *subclass*) under *match_type*.
- Confirmation of whether structural checks were performed and passed (*dependency_check_passed* - null here indicates it wasn't performed) and which mode was used.
- Additional details like the overall configuration used (*approach*), the final node mapping (*node_map*), internal flags (*detail_flags*), and specificity counts (*argument_specificities*).

This detailed trace provides transparency and allows for error analysis, making the system's reasoning process explainable.

```
{
  "attempt_id": 4,
  "section": 1,
  "pattern": "prüfen(gutgläubig, Erwerb)",
  "approach": "lemma_lemma_synonyms",
  "matched_sentences": [
    "Beim Verkauf des Fahrrades von Paula an Fanny, ist zu überprüfen, ob
 ein gutgläubiger Erwerb durch Fanny in Frage kommt."
  ],
  "predicate_matches": [
    {
      "premise_node_id": "c_1",
      "premise_node_label": "prüfen",
      "hypothesis_node_id": 11,
      "hypothesis_node_label": "überprüfen",
      "match_type": "equals_synonyms",
      "similarity_score": 0.92857
    }
  ],
  "argument_matches": [
    {
      "premise_node_id": "c_4",
      "premise_node_label": "gutgläubig",
      "hypothesis_node_id": 15,
      "hypothesis_node_label": "gutgläubig",
      "match_type": "equals_lemma",
      "similarity_score": 1.0
    },
    {
      "premise_node_id": "c_5",
      "premise_node_label": "Erwerb",
      "hypothesis_node_id": 16,
      "hypothesis_node_label": "Erwerb",
      "match_type": "equals_lemma",
      "similarity_score": 1.0
    }
  ],
  "detail_flags": {
    "initial_match_found": true,
    "argument_match_passed": true,
    "dependency_check_passed": null
  },
  "evaluation": "TP"
}
```


Listing 4.1: Example of a record of a successful entailment

CHAPTER 5

Experimental Setup

This chapter details the experimental framework used to evaluate our graph-based entailment detection system. We begin by exploring the dataset specific to our AI legal tutor use case, providing insights into its structure and characteristics. Subsequently, we describe the experimental data preprocessing steps investigated, namely coreference resolution and compound word splitting. We then introduce the baseline systems against which our approach is compared and define the evaluation metrics used to assess performance. Finally, we outline the various matching configurations of our system that are systematically evaluated to understand the impact of different features and constraints.

5.1 Data Exploration

The dataset for our AI legal tutor use case, previously introduced in Chapter 2, comprises three core components for each scenario:

1. The legal case description.
2. A detailed sample solution with key passages for the case.
3. Student attempts to solve the case.

These materials originate from the domain of Austrian jurisprudence.

For this study, we were provided with data for two distinct legal cases, including sample solutions and several annotated student attempts for each. Our experiments focus specifically on one case (hereafter referred to as Case 1) for which detailed annotations are available. For Case 1, the sample solution is segmented into "key passages", each representing a crucial point or concept required for a correct answer. These key passages

correspond to the relational tuples (e.g., *"prüfen(gutgläubig, Erwerb)" [check(bona_fide, acquisition)]*) that serve as the premise patterns for our entailment detection system.

The evaluation for Case 1 is structured into 8 distinct sections, each potentially containing one or more key passages (premise patterns). A student's attempt is evaluated against these 8 sections, and the score for each section contributes to the overall grade. Students are unaware of this internal sectional structure, its weighting, or the specific points allocated per section; they only receive the case description to solve. Some sections might be satisfied by matching a single premise pattern, while others may offer alternative correct answers, requiring a match with any one of several patterns. In total, Case 1 involves 18 distinct premise patterns distributed across the 8 sections.

We have access to 5 student attempts for Case 1. The gold-standard annotation classifies 24 student answer segments as correctly entailing the required premise pattern(s), while 16 segments are marked as incorrect (either contradicting the sample solution or failing to cover the necessary aspect). The combined student attempts consist of 53 sentences and 707 tokens.

5.2 Dependency Parser

For this task, we employ the *de_hdt_lg* model, a German language model available within the spaCy framework [Honnibal et al., 2020]. The dependency parser component of this model is trained primarily on the Hamburg Dependency Treebank (HDT) [Foth et al., 2014], which has been converted to the Universal Dependencies (UD) format (UD/de-hdt). The HDT was created at the University of Hamburg through manual annotation, guided by specific annotation standards and aided by a constraint-based parser. It is a large corpus consisting of 261,821 sentences (approximately 4.8 million tokens), sourced entirely from the German news website heise.de, covering articles published between 1996 and 2001. In addition to the HDT data, the *de_hdt_lg* model's training also incorporates data from the WikiNER corpus [Nothman et al., 2013].

5.3 Data Preprocessing Experiments

Beyond the core graph construction described in Chapter 4, we investigate the impact of two optional preprocessing steps applied to the student attempts. These steps are treated as experimental variations, evaluated for their potential benefits. Standard preprocessing techniques like stopword removal or extensive lemmatization were not applied to preserve the original linguistic structure as much as possible.

5.3.1 Coreference Resolution

Coreference occurs when multiple expressions in a text refer to the same entity. A common example involves pronouns referencing previously mentioned nouns, such as in "The cat is hungry. It wants food," where "It" refers to "The cat." Resolving these

references (mapping "It" back to "The cat") can be important for understanding the text's meaning.

This phenomenon poses challenges, particularly in German. Consider this example from a student attempt: *"Wenn Fanny gutgläubig ist, d.h., sie wusste nicht und konnte nicht wissen, dass Paula nicht die Eigentümerin war, dann hat sie das Eigentum an dem Fahrrad durch den Kauf und die Übergabe gutgläubig erworben." [If Fanny is in good faith, i.e., she did not know and could not have known that Paula was not the owner, then she acquired ownership of the bicycle in good faith through the purchase and handover.]*

Here, the pronoun *"sie" [she]* appears twice. The first *"sie"* likely refers to Fanny, while the second *"sie"* also refers to Fanny. However, resolving the reference requires understanding the sentence structure and context, especially given that both *"Fanny"* and *"Paula"* are female names, and *"Paula"* is mentioned later in the sentence. While humans often resolve such ambiguities intuitively, it presents a significant challenge for automated systems.

The GKR framework [Kalouli and Crouch, 2018], which inspired our graphical knowledge representation, incorporates coreference resolution directly into the graph building process. Given the complexity, we opted for a simpler approach: applying coreference resolution as an experimental preprocessing step before generating the graphs. We utilize the coreference resolution component available within the spaCy framework [Honnibal et al., 2020] (specifically, its German models) to process the student attempts. The hypothesis motivating this experiment is that the annotated premise patterns likely use the full names of entities, not pronouns. Resolving pronouns in the student attempts back to these full names could potentially increase the likelihood of finding matches.

5.3.2 Compound Word Splitting

German frequently uses compound words, where two or more words are combined to form a single, valid new word (e.g., *"Fußballspiel" [football game] from "Fußball" [football] + "Spiel" [game]*). Sometimes, connecting letters like 's' are used, as in *"Eigentumserwerb" [property acquisition] ("Eigentum" (property) + "s" + "Erwerb" (acquisition))*.

While these are standard words, their composite nature might obscure semantic relationships if the individual components are relevant for entailment. GermaNet [Hamp and Feldweg, 1997] provides information about the structure for the most common compound words in German (identifying the *head* and *modifier* components).

As another experimental preprocessing step, we leverage this information. For tokens identified as compounds in GermaNet, we replace the compound word with a phrase explicitly stating the relationship between its components (e.g., replacing *"Eigentumserwerb" [property acquisition]* with *"Erwerb von Eigentum" [acquisition of property]*). This normalization aims to expose the underlying concepts within compounds, potentially facilitating matches with premise patterns that might refer to these components separately. We evaluate the impact of applying this splitting process to the student attempts.

5.4 Baselines and Evaluation Metrics

To contextualize the performance of our proposed graph-based system, we compare it against several baselines and employ standard evaluation metrics.

5.4.1 Baselines

We define two primary baselines:

1. **Sentence Embedding Baseline:** This baseline addresses the overall task of identifying required information within student attempts, framing it as a sentence similarity problem rather than a strict NLI task. We compare simplified source sentences from the sample solution (these are the same sentences used to derive the premise patterns for our graph-based method) against individual sentences extracted from the student's attempt. The core idea is to transform these sentences into vector representations (embeddings) and calculate their similarity. A student's answer regarding specific required information is considered correct if the similarity score between the corresponding source sentence and any sentence in their attempt exceeds a predefined threshold. The process involves the following steps:
   - Student attempts are split into individual sentences using spaCy's German language model.
   - Embeddings are generated for both the simplified source sentences and each student sentence using a multilingual sentence transformer model ('distiluse-base-multilingual-cased-v1') [Reimers and Gurevych, 2019].
   - The cosine similarity between each source sentence embedding and each student sentence embedding is calculated.
   - If any similarity score exceeds a predetermined threshold, the student attempt is considered to entail the required information for that section.
2. **Triplet Matching Baseline (No External Resources):** This uses our developed graph-matching system but in its most basic configuration. It performs matching based solely on exact lemma matches between the premise pattern nodes and hypothesis sentence nodes, without leveraging any lexical-semantic information from GermaNet (synonyms, hypernyms, hyponyms) or morphological features from DWDSmor. Negation and structural checks are also disabled. This baseline isolates the contribution of the core graph alignment algorithm itself.

Additionally, we attempted to establish a baseline using the HOLMES system [Hudson, 2023], an Open Information Extraction tool based on predicate logic that has been adapted for German as well. We provided HOLMES with the simplified premise sentences (as used in the LLM baseline) and the original student attempt sentences. However, HOLMES

was unable to extract matching structures and consequently failed to detect any entailments in our dataset, resulting in zero recall. Therefore, it was not included in the final comparative evaluation.

5.4.2 Evaluation Metrics

We frame the entailment detection task as binary classification. For each pairing of a premise pattern (from a gold-standard section) and a hypothesis sentence (from a student attempt), the system predicts either positive (entailment) or negative (no entailment). The primary goal is to correctly identify the positive instances, where the student's text successfully entails the meaning required by the sample solution's premise pattern. It is important to emphasize that this evaluation focuses solely on extracting positive evidence of entailment to provide feedback on correctly mentioned points; the system is not designed to explicitly identify contradictions or report missing information.

Performance is evaluated using standard metrics from information retrieval and classification: Precision, Recall, and F1-score. These metrics are calculated based on the comparison between the system's predictions and the gold-standard annotations, categorized as follows:

- **True Positives (TP):** The system correctly predicts entailment when the gold standard indicates entailment. (The student's answer contains the required information, and the system detects it).
- **False Positives (FP):** The system incorrectly predicts entailment when the gold standard indicates no entailment. (The system claims the student provided the required information, but they did not, according to the annotation).
- **False Negatives (FN):** The system incorrectly predicts no entailment when the gold standard indicates entailment. (The student did provide the required information, but the system failed to detect it).
- **True Negatives (TN):** The system correctly predicts no entailment when the gold standard indicates no entailment. (The student did not provide the required information, and the system correctly identifies its absence).

It is important to note that our system, by design, searches for positive evidence of entailment and does not explicitly count True Negatives. Furthermore,False Negatives are inferred by identifying the gold-standard entailments that the system failed to predict.

The evaluation metrics are calculated as:

Recall (Sensitivity, True Positive Rate): Measures the proportion of actual positive instances (true entailments in the gold standard) that the system correctly identified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.1}$$

Precision: Measures the proportion of instances predicted as positive by the system that are actually positive according to the gold standard.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.2}$$

F1-Score: The harmonic mean of precision and recall, providing a single score that balances both metrics. It gives more weight to lower values, making it suitable when minimizing both false positives and false negatives is important.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{5.3}$$

These three metrics form the basis for evaluating and comparing the different system configurations and baselines presented in the Chapter 6.

5.5 Matching Settings Experiments

After constructing the graphs for premises and hypotheses (potentially incorporating the preprocessing steps described in Section 5.3), the entailment detection module performs the matching process. This process is highly configurable, allowing us to systematically evaluate the contribution of different information layers and matching strategies. The only parameter fixed during graph construction is the depth for retrieving hypernyms and hyponyms from GermaNet, which defaults to 2 levels.

We experiment with various settings controlled by boolean flags and configuration parameters:

- **Matching Type Flags (*init*, *arg*):** These internal flags determine whether lemma (*False*) or POS (*True*) is used for initial predicate matching (*init*) and argument matching (*arg*). Our experiments focus on *init=False, arg=False* (Lemma-Lemma) and *init=False, arg=True* (Lemma-POS).
- **POS Matching Granularity (*pos_matching_type*):** When POS matching is enabled (*arg=True*), this parameter selects which POS representation to use: spaCy's *pos* tags (e.g., *NOUN*, *VERB*) or its more fine-grained *tag* tags (providing more specific distinctions).
- **Use GermaNet Relations (*use_synonyms*, *use_hypernyms*, *use_hyponyms*):** These boolean flags control whether the lexical layer is utilized during lemma matching. Enabling these allows matching based on synonyms, hypernyms (generalizations), or hyponyms (specifications) retrieved from GermaNet, in addition to exact lemma matches. We test configurations using none, only synonyms, or all three relation types.

- **Use Negation (*use_negation*):** A boolean flag determining whether the negation status from the context layer (Section 4.1.5) is enforced. If *True*, a premise node and a hypothesis node can only match if they have the same negation status (both negated or both affirmed). If *False*, negation status is ignored during matching. This allows evaluating the impact of negation handling.
- **Use Morphological Constraints (*use_gender*, *use_number*):** Boolean flags to enable constraints based on morphological features from the property layer (Section 4.1.4). If *use_gender* is *True*, matching nodes (typically nouns/adjectives when using lemma matching) must have compatible grammatical gender. If *use_number* is *True*, they must match in grammatical number (singular/plural). These are primarily explored in combination with Lemma-POS matching to potentially constrain the broader matches produced by POS tags.
- **Structural Validation (*edge_check_mode*):** This parameter controls the level of structural consistency checking performed during the validation stage (Section 4.2.3) after initial node matches are found. It determines whether and how the dependency relations between matched nodes are compared.

By systematically varying these settings, we aim to identify which combination of semantic information (lemmas, lexical relations), structural information (POS tags, dependency relations), morphological features (gender, number), and context (negation) yields the best performance for entailment detection in our specific use case, balancing precision and recall.

In the next chapter, we will present the results obtained from applying these different data preprocessing steps and matching configurations, including an error analysis facilitated by the system's explainability features.

CHAPTER 6

Results & Discussion

This chapter presents and discusses the results of the experiments detailed in Chapter 5. We evaluate our proposed symbolic entailment detection method within the context of an AI legal tutor use case. Given that entailments make up 60% of our evaluation dataset, our analysis places particular emphasis on precision metrics, though we also consider recall and F1 scores to provide a comprehensive evaluation.

First, we compare the performance of our symbolic baseline approach against two preprocessing techniques: coreference resolution and compound word splitting. Second, we investigate the impact of incorporating external lexical resources from GermaNet. Third, we analyze how different strategies for matching lemmas and POS tags affect performance. Fourth, we explore the influence of different structural validation modes on the dependency graph matching. Finally, we assess the effectiveness of an explicit negation detection step in improving precision.

6.1 Baseline Performance and Default Settings

Before evaluating specific components like preprocessing or lexical resources, we establish the performance of our baseline systems. We compare a standard BERT Sentence Embedding approach, as explained in Section 5.4.1, against our basic symbolic method (lemma-lemma matching) using different structural constraints.

6.1.1 BERT Sentence Embedding Baseline

The BERT baseline calculates the cosine similarity between sentence embeddings of student answers and target solutions. A match is predicted if the similarity exceeds a predefined threshold. Table 6.1 shows the performance for various thresholds.

While a low threshold (0.1) yields the highest F1 score due to perfect recall, it suffers from low precision. The achieved precision of 0.667 is considered low, offering only a

Cosine similarity threshold	Precision	Recall	F1 Score
0.1	0.667	1.000	0.800
0.3	0.630	0.708	0.667
0.5	0.615	0.333	0.432
0.7	0.667	0.083	0.148

Table 6.1: BERT Sentence Embedding baseline performance by the cosine similarity threshold

marginal improvement over simply guessing the positive class for every instance (which would achieve 60% precision).

6.1.2 Symbolic Baseline and Structural Validation

Our basic symbolic baseline (*lemma_lemma*) matches lemmas directly without external resources. We evaluated its performance under the three structural validation modes described in Section 6.5: (*open*), (*path exists*), and (*dependency exact*). Results are shown in Table 6.2.

Structural Mode	Precision	Recall	F1 Score
open (*lemma_lemma*)	0.875	0.292	**0.438**
path exists (*lemma_lemma*)	0.833	0.208	0.333
dependency exact (*lemma_lemma*)	1.000	0.167	0.286

Table 6.2: Symbolic Baseline (`lemma_lemma`) Performance by Structural Mode

As expected, stricter structural requirements increase precision (reaching 1.000 for "dependency exact") but significantly decrease recall and the overall F1 score. The "open" mode, requiring only the presence of matching terms within the same sentence, achieves the highest F1 score (0.438) among the symbolic baselines. Notably, all symbolic baselines demonstrate substantially higher precision than the BERT approach, with the "dependency exact" mode achieving perfect precision, which is particularly valuable for our use case. However, our symbolic baseline suffers from low recall. A primary goal of this work was to increase recall while maintaining the high precision advantages of the symbolic approach. We will examine the results of our efforts to address this recall challenge in the next sections.

6.2 Impact of Preprocessing Techniques

We conducted experiments to evaluate the effect of text preprocessing on the system's performance. We compared three scenarios: no preprocessing (baseline), coreference resolution applied, and compound word splitting applied (as described in Section 5.3).

Table 6.3 summarizes the precision, recall, and F1 scores for each preprocessing step compared to the baseline (*lemma_lemma_baseline*) which uses simple lemma matching without any external resources or specific structural constraints.

Preprocessing	Precision	Recall	F1 Score
None (Symbolic Baseline)	0.875	0.292	**0.438**
Coreference Resolution	0.875	0.292	**0.438**
Compound Word Splitting	1.000	0.250	0.400

Table 6.3: Performance comparison of different preprocessing techniques.

As shown in Table 6.3, applying coreference resolution yielded identical results to the baseline. This indicates that, for our dataset and annotation scheme, resolving pronouns did not lead to the extraction of additional or different relevant relational tuples compared to the baseline. A closer look at the annotations revealed that relational tuples often omitted explicit subjects or objects when they were clearly inferable from the case description context.

Compound word splitting resulted in perfect precision (1.000) but lower recall (0.250) and F1 score (0.400) compared to the baseline. The improvement in precision is noteworthy, as it eliminates false positives, though at the cost of some recall. While splitting compounds might seem beneficial for precision, it sometimes hindered matching, as discussed below.

Overall, neither preprocessing technique improved the F1 score over the baseline in this specific setup, though compound splitting did achieve perfect precision.

6.2.1 Qualitative Analysis

The explainability of our symbolic approach allows for a qualitative analysis of the differences observed.

Coreference Resolution: Consider the example: *"Wenn Fanny gutgläubig ist, d.h., sie wusste nicht und konnte nicht wissen, dass Paula nicht die Eigentümerin war, dann hat sie das Eigentum an dem Fahrrad durch den Kauf und die Übergabe gutgläubig erworben"* ["If Fanny is in good faith, i.e., she did not know and could not have known that Paula was not the owner, then she acquired ownership of the bicycle in good faith through the purchase and handover"]. The system needed to resolve "sie" [she] in the final clause. It incorrectly resolved the coreference to "Paula" instead of "Fanny". However, the annotated target tuple for this segment was simply "erwerben(Eigentum)" [acquire(ownership)], omitting the agent because only Fanny could acquire ownership in this context. Thus, despite the incorrect resolution, the system still matched the tuple, and precision was unaffected.

In a simpler case, *"Es gehört Fanny, da sie es gutgläubig erworben hat"* ["It belongs to Fanny, as she acquired it in good faith"]. Coreference resolution correctly changed the sentence to *"Es gehört Fanny, da Fanny es gutgläubig erworben hat"* ["It belongs to Fanny, as Fanny acquired it in good faith"]. The target tuple was "erwerben(gutgläubig)"

[acquire(good_faith)], again omitting the agent. If the annotation had been more specific, e.g., "erwerben(Fanny, gutgläubig)", the coreference resolution step would have been crucial for finding the match, whereas the baseline might have missed it. This suggests coreference resolution could be beneficial for tasks with more detailed relational tuple annotations that include subjects and objects more consistently.

Compound Word Splitting: Splitting compound words sometimes prevented matches. For example, the system aimed to match the tuple *"beweglich(Fahrrad, Sache)" [movable(bicycle, thing)]* indicating that the bicycle is a movable object. When processing an answer containing *"Fahrrad" [bicycle]*, the splitting algorithm segmented it into *"Fahr" [drive/ride]* and *"Rad" [wheel/bike]*. This prevented a direct lemma match with the target tuple's argument *"Fahrrad" [bicycle]*. While lexical resources like GermaNet (discussed next) could potentially bridge this gap by relating *"Rad" [wheel/bike]* to *"Fahrrad" [bicycle]* (e.g., as a synonym or hypernym), the splitting process itself introduced this intermediate hurdle. Given that incorporating lexical resources offers a more robust way to handle such variations (Section 6.3), we chose to omit the compound splitting step in subsequent experiments. We hypothesize that relationships between compounds and their components are better captured through semantic relations like synonymy or hyponymy.

6.3 Impact of Lexical Resources

We investigated how incorporating lexical-semantic knowledge from GermaNet affects performance. We focused on integrating synonyms, hyponyms (more specific terms), and hypernyms (more general terms) into the matching process, using the default "open" structural validation mode.

Table 6.4 presents the results compared to the symbolic baseline and the BERT sentence embedding baseline.

Approach	Relation Type	Precision	Recall	F1 Score
Symbolic Baseline	-	0.875	0.292	0.438
Lemma+Lemma+Synonyms	Synonyms	0.857	0.500	**0.632**
Lemma+Lemma+Hyponyms	Hyponyms	<u>0.889</u>	0.333	0.485
Lemma+Lemma+Hypernyms	Hypernyms	0.875	0.292	0.438

Table 6.4: Performance comparison using lexical resources (open Structure)

The inclusion of synonyms provided the most substantial improvement, boosting the F1 score to 0.632 primarily through increased recall (0.500), while maintaining a strong precision of 0.857. Incorporating hyponyms also slightly improved the F1 score to 0.485 and achieved the highest precision (0.889) among the lexical resource approaches. However, adding hypernyms did not lead to any improvement over the symbolic baseline in this configuration. A possible explanation for this lack of improvement with hypernyms could be that we limited the depth of both hypernym and hyponym relations to 2 in our GermaNet integration. While this depth appears sufficient for hyponyms to capture more

specific terms, it may be too restrictive for hypernyms, where a greater depth might be needed to reach more general terms that could facilitate additional matches.

6.3.1 Analysis of Synonym Integration

While allowing synonym matches significantly increased recall, it occasionally introduced false positives, slightly reducing precision compared to the baseline. For instance, the student's answer *"beim Fahrrad handelt es sich um eine bewegliche Sache"* [*"the bicycle is a movable object"*] was incorrectly matched with the target pattern "verkauft(Fahrrad)" [sold(bicycle)]. This occurred because *"handeln" [to trade]* was matched as a synonym for *"verkaufen" [to sell]* in GermaNet for a specific sense. The system's inability to perform accurate word sense disambiguation led to this incorrect match based on an inappropriate synonym sense.

Another false positive arose from the target pattern "verkauft(Fahrrad)" derived from the solution text *"Das Fahrrad wurde verkauft"* [*"The bicycle was sold"*]. The system matched the student attempt *"Da Paula nicht Eigentümerin des Fahrrades ist kann Sie dieses nicht verkaufen"* [*"Since Paula is not the owner of the bicycle, she cannot sell it"*]. This match is incorrect for two reasons:

- Tense/Aspect Mismatch: The target pattern refers to a past, completed action, while the student's answer discusses a potential future action. While we include the verb form in our property sublayer, we did not explore it as a constraint in this experiment.
- Negation: The student's answer explicitly negates the action (*"nicht verkaufen" [cannot sell]*). This particular false positive is successfully eliminated when negation detection is enabled (Section 6.6), demonstrating how negation handling can improve precision.

It was also observed that using GermaNet's path-based relatedness measure (instead of strict synonym sets), as explained in Section 4.2.1, implicitly captured some hyponymic relations. For example, terms like *"Fahrrad" [bicycle]* and *"Rad" [bike]*, *"Kaufvertrag" [purchase contract]* and *"Vertrag" [contract]*, or *"Preis" [price]* and *"Kaufpreis" [purchase price]* might be considered related via short paths in GermaNet, even if not strict synonyms.

6.4 Lemma and POS Matching Strategies

This section explores different strategies for matching nodes in the graphs, combining lemma matching with POS information. As defined in Chapter 4, relational tuples follow the format "predicate(argument1, argument2, ...)". Our baseline approach ("lemma_lemma") requires exact lemma matches for both the predicate and its arguments (or their synonyms/hyponyms when lexical resources are enabled).

We experimented with relaxing the matching criteria for arguments, requiring only the predicate to match by lemma (or related term) while allowing arguments to match based on POS tags instead. We tested two types of POS tags: coarse-grained universal POS tags ("lemma_pos") and fine-grained STTS tags ("lemma_tag").

Table 6.5 shows the performance of these different matching strategies, combined with GermaNet synonyms and using the default "Open" structural validation mode. It also includes a variant ("lemma_tag_syno_gender") that additionally checks for gender agreement, as well as variants with negation detection enabled.

Approach	**Precision**	**Recall**	**F1 Score**
Symbolic Baseline	0.875	0.292	0.438
lemma_pos	0.846	0.458	0.595
lemma_tag	0.846	0.458	0.595
lemma_tag_syno	0.818	0.750	0.783
lemma_tag_syno_gender	0.800	0.667	0.727
lemma_pos_syno_negation	0.900	0.750	0.818
lemma_tag_syno_negation	<u>0.947</u>	0.750	**0.837**
lemma_tag_syno_negation_gender	0.941	0.667	0.780

Table 6.5: Performance comparison of different lemma and POS matching modes (Open Structure, with Synonyms where indicated)

Relaxing the argument matching to use POS or TAG tags ("lemma_pos", "lemma_tag") improved the F1 score compared to the symbolic baseline. The combination of fine-grained TAG matching for arguments with lemma matching for predicates and incorporating synonyms ("lemma_tag_syno") achieved a strong F1 score of 0.783. While this approach shows a slight decrease in precision (0.818) compared to the symbolic baseline (0.875), the substantial gain in recall (0.750 vs. 0.292) justifies this trade-off.

When negation detection is enabled, we observe further improvements in performance. The "lemma_tag_syno_negation" approach achieves the highest overall F1 score (0.837) with excellent precision (0.947) while maintaining the high recall (0.750). This demonstrates that fine-grained TAG attributes are more effective than simple POS tags when negation is enabled, as seen by comparing with "lemma_pos_syno_negation" (F1 = 0.818).

Adding gender agreement checks consistently lowers recall in both settings, though it shows potential for reducing false positives when combined with negation detection. This suggests that enforcing gender agreement is too restrictive for this task, potentially filtering out valid matches where gender information is not critical to the semantic meaning.

Overall, these results indicate that the combination of lemma matching for predicates, TAG-based matching for arguments, synonym integration, and negation detection provides the most balanced and effective approach for matching relational tuples in German legal texts.

6.5 Impact of Structural Validation Modes

We investigated the effect of varying the strictness of structural validation when matching dependency paths between the student's answer and the target solution pattern. As detailed in Chapter 4, we compared three modes:

- Exact Dependency (dep): Requires the dependency relation label path between matched nodes to be identical.
- Path exists (path): Requires only that a directed path exists between the matched nodes in the student's answer graph, corresponding to the path in the target pattern, regardless of the specific edge labels.
- Open: The least restrictive mode, requiring only that the predicate and argument nodes are present in the same sentence, without checking the dependency path between them.

Table 6.6 compares these modes using the "lemma_lemma_syno" approach (lemma matching for predicate and arguments, plus synonyms).

Approach	Structural Mode	Precision	Recall	F1 Score
Symbolic Baseline	Open	0.875	0.292	0.438
lemma_lemma_syno	Open	0.857	0.500	**0.632**
lemma_lemma_syno	Path Existence	0.909	0.417	0.571
lemma_lemma_syno	Exact Match	1.000	0.292	0.452

Table 6.6: Performance comparison for different structural validation modes (using Lemma+Lemma+Synonyms)

As expected, the "Dependency exact" mode yielded the highest precision (1.000) but the lowest recall (0.292), resulting in an F1 score of 0.452. The perfect precision achieved with this mode is particularly valuable in applications where false positives must be minimized. The "Path exists" mode offered a compromise with higher precision (0.909) than the "open" mode and better recall than the "Dependency exact" mode, achieving an F1 score of 0.571. This mode confirms the intuition that the directed nature of dependency graphs implies meaningful relationships, even if the specific labels vary.

However, the highest recall (0.500) and overall F1 score (0.632) were achieved with the "open" mode, while still maintaining strong precision (0.857). This might be because the student attempts and the sample solution operate within a constrained domain with limited vocabulary. Consequently, simply mentioning the correct entities (predicate and arguments) in the same sentence is often sufficient to indicate the correct meaning, even if the grammatical structure or dependency relations differ slightly from the target pattern. Furthermore, potential inaccuracies in dependency parsing for complex sentences could also favour the less restrictive "open" mode.

6.6 Impact of Negation Detection

Finally, we analyze whether incorporating an explicit negation detection step improves system performance, specifically by reducing false positives and thus increasing precision. We applied the negation check to several of the best-performing configurations from the previous experiments. The negation component verifies whether the matched predicate or tuple is negated in the student's answer compared to the target pattern.

Table 6.7 compares the performance of selected approaches with and without the negation detection step enabled.

Approach	Negation	Precision	Recall	F1 Score
Symbolic Baseline	No	0.875	0.292	0.438
lemma_lemma_synonyms	No	0.857	0.500	0.632
lemma_lemma_synonyms_negation	Yes	1.000	0.500	0.667
lemma_tag_syno_open	No	0.818	0.750	0.783
lemma_tag_syno_negation_open	Yes	0.947	0.750	**0.837**
lemma_tag_syno_path	No	0.812	0.542	0.650
lemma_tag_syno_negation_path	Yes	0.929	0.542	0.684

Table 6.7: Performance comparison with and without negation detection

The results consistently show that applying the negation detection step increases precision, often significantly. For instance, adding negation to the "lemma_lemma_synonyms" approach raised precision from 0.857 to 1.000, improving the F1 score from 0.632 to 0.667. Similarly, for the best performing "lemma_tag_syno_open" approach, negation detection increased precision from 0.818 to 0.947, boosting the F1 score from 0.783 to 0.837. The recall remains unaffected as negation detection only filters out existing matches, it does not find new ones. These results strongly support the inclusion of an explicit negation handling mechanism for improving the accuracy of the symbolic entailment system, particularly for applications where precision is required.

6.7 Analysis of Missed Entailments

This section examines some examples of correct student answers that the system failed to identify (false negatives) and discusses potential reasons.

1. Student answer: *"Entgeltlichkeit liegt aufgrund des Kaufpreises iHv 100€ vor" ["Consideration exists due to the purchase price of €100"]*. The target patterns for this point were *"entgeltliches(Rechtsgeschäft)" ["onerous(legal_transaction)"]* and *"Preis(angemessener)" ["price(reasonable)"]*. Matching either should suffice.

The system failed to entail the noun *"Entgeltlichkeit" ["consideration/onerousness"]* from the adjectival predicate *"entgeltlich" ["onerous"]* in the first pattern. This highlights a limitation of strict lemma matching; morphological variants might require explicit

handling or more sophisticated lexical resources. Furthermore, the pattern required the argument *"Rechtsgeschäft" ["legal transaction"]* (or a synonym) to be present, which was absent in the student's answer, although contextually implied.

For the second pattern, *"Preis(angemessener)" ["reasonable(price)"]*, the system (and indeed, a human reader without further context) cannot determine if the mentioned price of €100 is *"angemessen" ["reasonable"]*. This points to the need for domain-specific knowledge, which are beyond the scope of the current symbolic matching approach.

2. Student answer: *"Da Paula nicht Eigentümerin des Fahrrades ist kann Sie diese nicht verkaufen" ["Since Paula is not the owner of the bicycle, she cannot sell it"]*. This was annotated as correctly conveying the meaning of the target pattern *"scheidet(derivativ, Erwerb)" ["ruled_out(derivative, acquisition)"]*, meaning derivative acquisition is not possible. Capturing this entailment requires significant legal background knowledge to understand that Paula's inability to sell due to lack of ownership implies the impossibility of derivative acquisition by Fanny. This represents a complex entailment requiring domain-specific reasoning beyond lexical and structural matching. It is worth considering whether such an answer, which relies on implied legal reasoning (Paula's lack of ownership prevents derivative acquisition) rather than stating it explicitly, should be considered fully correct for assessment purposes.

3. Student answer: *"es handelt sich um eine individuell bestimmbare Sache, die beweglich ist." ["it is an individually determinable thing that is movable."]*. The corresponding target pattern was *"Fahrrad(bewegliche, Sache)" ["bicycle(movable, thing)"]*. Although the student's answer correctly identifies the bicycle as movable, the system missed the match. The pronoun *"es" ["it"]* refers to the bicycle, but this coreference was not resolved or linkable because *"Fahrrad" ["bicycle"]* was not mentioned in the preceding context of this specific student's answer. Even the POS/TAG matching strategies (Section 6.4) failed here, as the pronoun *"es" ["it"]* and the noun *"Fahrrad" ["bicycle"]* do not share the same POS or TAG attributes.

CHAPTER 7

Conclusion

This thesis presented the development and evaluation of a symbolic system for recognizing textual entailment in German legal texts, motivated by the requirements of an AI legal tutor application.

7.1 Contributions

The primary contributions of this work are the adaptation of the graphical knowledge representation that includes external German lexical resources and the negation detection component for the symbolic entailment detection system.

First, we employed a multi-layered knowledge graph representation for German text based on the GKR architecture [Kalouli and Crouch, 2018], building upon a dependency parsing. This representation then integrates several layers: a concept layer that simplifies the syntactic structure to core entities and relations guided by annotated target patterns; a lexical layer that enriches concept nodes with synonyms, hypernyms, and hyponyms from GermaNet, enabling matching beyond literal terms and increasing semantic flexibility; a property layer that adds detailed morphological and POS information (from spaCy and DWDSmor); and finally, a context layer focusing on negation, implemented via a rule-based system that determines the scope of negation cues within the dependency structure. This rich representation provides a structured base for semantic comparison.

Second, we developed and evaluated a fully symbolic, rule-based entailment detection system that uses configurable graph-matching methods to determine if the graph representation of a sentence entails the graph representation of another sentence.

7.2 Addressing Research Questions

Our experimental evaluation provided insights into the research questions posed at the beginning:

Regarding RQ1, whether symbolic rules grounded in German morphology and semantics can increase recall and precision compared to two different baseline approaches, our findings show significant enhancements. Incorporating German-specific semantic and morphological information improved entailment detection compared to both a standard sentence embedding baseline and a basic symbolic lemma-matching baseline. The integration of lexical resources (particularly synonyms from GermaNet) substantially improved recall while maintaining high precision, as shown in Chapter 6. Using detailed morphological features and POS tags allowed for more refined matching strategies. Several configurations of our symbolic system achieved perfect or near-perfect precision, reducing false positives.

Concerning RQ2, how unique German linguistic features like compound nouns, case-based syntax, and gender affect rule-based inference, our experiments revealed varied impacts. Explicitly splitting compound nouns did not improve overall performance and slightly reduced recall, though it increased precision in one setting. Problems came up because splitting sometimes prevented direct matches. Accessing components of compound nouns seemed better handled through hyponymy and synonym relations. Specific German language characteristics such as free-word order and case-bases syntax were implicitly captured by the dependency parser, requiring no explicit handling. While adding explicit case matching wasn't needed for structural validation in our setup, case information could be valuable for future tasks like word sense disambiguation. Regarding gender, incorporating grammatical gender agreement as a constraint consistently lowered recall, although it slightly improved precision sometimes (especially with negation detection). Enforcing strict gender agreement might be too restrictive for this task, filtering out valid matches where gender isn't critical to the meaning. The trade-off needs careful consideration.

Finally, for RQ3, whether explicit negation detection and morphological consistency checks can reduce false positives, our results strongly support this. The rule-based negation detection component consistently improved precision, often significantly, by correctly handling negated statements. The false positive rate was always reduced when negation detection was active, showing its effectiveness. Morphological checks like gender agreement also showed potential but had less impact and a larger recall penalty compared to negation detection.

7.3 Limitations

Despite promising results, our approach has several limitations.

The entire graphical knowledge representation depends heavily on the initial dependency parser accuracy. Errors in parsing, performed by the parser model trained primarily

on news text (HDT Treebank), directly affect the system's performance. The parser's effectiveness on domain-specific legal text was not explicitly evaluated and is a potential weakness. The HDT Treebank's origin (news articles in the tech domain from 1996-2001) might also represent a domain and time mismatch for current legal texts, potentially affecting parsing accuracy. Evaluating the parser's performance specifically on the target legal domain is crucial.

The context representation is limited, currently modeling only negation. Other factors like modality, certainty, or tense are not represented, limiting the ability to handle more complex semantic relationship, which are crucial in our use case.

A key limitation is the simple relational structure, which maps each predicate or argument to a single token from the sentence. This proves problematic for German compound verbs, where multi-token verbs like *"sich handeln um" [to be about]* have a distinct meaning from their base verb *"handeln" [to act/to trade]*. The inability to represent these multi-token verbs as single semantic units led to false negatives in certain entailment scenarios. Furthermore, the system was designed for a specific use case and doesn't support scenarios needing multiple tuple matches for a specific pattern. This limitation was evident in some missed entailments where the required meaning was expressed across multiple clauses or involved concepts not easily reducible to single tokens. More flexible graph structures or semantic representations might be needed.

The system targets only the entailment relation, unable to capture contradiction or neutrality, limiting its use in general NLI tasks. Extending it for more comprehensive feedback requires significant additions.

The system struggles with handling implicit knowledge and reasoning. The analysis of missed entailments showed it cannot handle cases needing significant domain knowledge, complex inference. Symbolic matching struggles with these knowledge-intensive inferences as it relies on explicit node and edge matching, limiting its ability to capture paraphrases or inferences not directly supported by resources.

Morphological variation sometimes caused failures when entailment depended on recognizing relationships between different word forms (like noun vs. adjective), indicating limitations in lemmatization or lexical matching.

Finally, word sense disambiguation is basic; the heuristic used sometimes led to incorrect sense matches and false positives, suggesting a need for a more advanced WSD mechanism.

7.4 Future Work

Based on these findings and limitations, several directions for future research appear useful. We divide into two parts: improvements to the symbolic entailment detection system and suggestions for future work on the AI legal tutor application.

7.4.1 Symbolic textual entailment

Future enhancements to the symbolic system could focus on several areas. Firstly, the context layer should be expanded beyond negation to model other relevant contexts, such as modality, certainty, tense, and aspect, which is crucial for improving entailment accuracy.

Secondly, investigating more flexible relational representations beyond simple token-based tuples could better handle complex meaning structures inherent in legal language.

Thirdly, implementing an adaptive matching strategy could improve robustness. Such a strategy could start with strict settings for high precision and progressively relax constraints for initially unmatched patterns. This approach would provide confidence information based on the relaxation level, allowing low-confidence matches to be flagged for review by a domain expert or a more sophisticated large language model. Exploring hybrid approaches that integrate large language models could combine the symbolic system's precision and interpretability with data-driven robustness, particularly for complex cases. For instance, vector representations could be adapted for heuristic matching in tasks like Word Sense Disambiguation (WSD) and semantic similarity assessment.

Furthermore, the negation detection component could be enhanced to recognize morphological negation cues in adjectives and adverbs, such as prefixes (e.g., "a-", "un-", "ir-", "des-") and suffixes (e.g., "-los"). Detecting such implicit negations would further enhance the system's reasoning capabilities.

Finally, the system's utility could be enhanced by extending it to detect contradictions, not just entailment. This would involve defining specific conflict patterns or integrating domain-specific rules.

7.4.2 AI legal tutor application

Specific improvements related to the AI legal tutor application should address parser reliability and domain knowledge integration. Given the system's reliance on syntactic structure, the dependency parser's reliability is critical. Therefore, a thorough evaluation of its performance on German legal texts is necessary, along with exploring alternatives like domain-specific fine-tuning or different parser models to quantify and mitigate parsing errors.

Integrating domain-specific knowledge is also essential. Incorporating resources such as legal ontologies, specialized dictionaries, and references to relevant legal paragraphs would significantly improve the handling of specific terminology and enable more complex reasoning patterns relevant to the legal domain.

Beyond entailment and contradiction, the tutor application could benefit from explicitly detecting affirmation. Similar to negation detection, identifying when a student's answer clearly affirms a concept could provide valuable feedback regarding their understanding and the direction of their reasoning. To further improve negation handling itself, we advise

domain experts to incorporate more domain-relevant negation cues into the system's knowledge base.

In conclusion, this thesis demonstrated the potential of a linguistically-informed, symbolic graph-based approach for precise textual entailment detection in the challenging domain of German legal text. While achieving high precision and offering explainability, the system faces limitations related to parser dependency, knowledge representation, and complex inferences. The outlined future work, particularly exploring hybrid models and integrating domain knowledge, offers promising paths towards building more robust and comprehensive NLI systems for specialized domains like law.

Overview of Generative AI Tools Used

In this thesis following supportive tools were used:

- **ChatGPT-4o** was used for proofreading and improving the clarity of the content of some paragraphs in the thesis. Overall, the following prompt was used: *"Can you double-check the grammar? Rephrase something if unclear, but without changing the key content or the writing style: [input paragraph]"* Each change was then manually checked to ensure that the original meaning and style were preserved.
- **DeepL** was used to provide translations for legal terms and phrases from German to English.
- **Google Translate** was used to provide an initial translation for the sections *Abstract* and *Acknowledgements*. Both translations were then manually refined.
- **Grammarly** was employed once as a final proofreading tool to ensure that the final version of the thesis was free of grammatical errors.

List of Figures

List of Tables

Bibliography

[Akbik and Löser, 2012] Akbik, A. and Löser, A. (2012). Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, AKBC-WEKEX '12, page 52–56, USA. Association for Computational Linguistics.

[Angeli et al., 2015] Angeli, G., Johnson Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In Zong, C. and Strube, M., editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

[Bassa et al., 2018] 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.

[Bhutani et al., 2016] Bhutani, N., Jagadish, H. V., and Radev, D. (2016). Nested propositions in open information extraction. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Austin, Texas. Association for Computational Linguistics.

[Boer et al., 2024] Boer, D., Koch, F., and Kramer, S. (2024). Harnessing the power of semi-structured knowledge and llms with triplet-based prefiltering for question answering.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In Màrquez, L., Callison-Burch, C., and Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

[Brown et al., 1992] Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

[Carrillo de Albornoz et al., 2012] Carrillo de Albornoz, J., Plaza, L., Díaz, A., and Ballesteros, M. (2012). UCM-I: A rule-based syntactic approach for resolving the scope of negation. In Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., and Yuret, D., editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 282–287, Montréal, Canada. Association for Computational Linguistics.

[Cui et al., 2018] 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.

[Dagan et al., 2005] Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

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

[Engelbach et al., 2023] Engelbach, M., Klau, D., Scheerer, F., Drawehn, J., and Kintz, M. (2023). Fine-tuning and aligning question answering models for complex information extraction tasks.

[Etzioni et al., 2008] Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

[Fader et al., 2011] Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 1535–1545, USA. Association for Computational Linguistics.

[Falke et al., 2016] Falke, T., Stanovsky, G., Gurevych, I., and Dagan, I. (2016). Porting an open information extraction system from English to German. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 892–898, Austin, Texas. Association for Computational Linguistics.

[Foth et al., 2014] Foth, K. A., Köhn, A., Beuck, N., and Menzel, W. (2014). Because size does matter: The Hamburg dependency treebank. 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 2326–2333, Reykjavik, Iceland. European Language Resources Association (ELRA).

[Hamp and Feldweg, 1997] Hamp, B. and Feldweg, H. (1997). GermaNet - a lexical-semantic net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

[Honnibal et al., 2020] Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

[Hudson, 2023] Hudson, R. P. (2023). Holmes extractor: Information extraction from english and german texts based on predicate logic. `https://github.com/richardpaulhudson/holmes-extractor`. Version 4.2.1, MIT License.

[Kalouli and Crouch, 2018] Kalouli, A.-L. and Crouch, R. (2018). GKR: the graphical knowledge representation for semantic parsing. In Blanco, E. and Morante, R., editors, *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 27–37, New Orleans, Louisiana. Association for Computational Linguistics.

[Kalouli et al., 2020] Kalouli, A.-L., Crouch, R., and de Paiva, V. (2020). Hy-NLI: a hybrid system for natural language inference. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5235–5249, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Klein and Geyken, 2010] Klein, W. and Geyken, A. (2010). Das Digitale Wörterbuch der Deutschen Sprache (DWDS). Volume 26:79–96.

[Levy and Dagan, 2016] Levy, O. and Dagan, I. (2016). Annotating relation inference in context via question answering. In Erk, K. and Smith, N. A., editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

[Mausam et al., 2012] Mausam, Schmitz, M., Soderland, S., Bart, R., and Etzioni, O. (2012). Open language learning for information extraction. In Tsujii, J., Henderson, J., and Paşca, M., editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

[McCoy et al., 2019] McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Miller, 1992] Miller, G. A. (1992). WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

[Neumann and Xu, 2003] Neumann, G. and Xu, F. (2003). Mining answers in german web pages. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, page 125, USA. IEEE Computer Society.

[Nie et al., 2020] Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial nli: A new benchmark for natural language understanding.

[Niklaus et al., 2018] 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 et al., 2016] Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

[Nothman et al., 2013] Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Sainz et al., 2022] Sainz, O., Gonzalez-Dios, I., Lopez de Lacalle, O., Min, B., and Agirre, E. (2022). Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2439–2455, Seattle, United States. Association for Computational Linguistics.

[Smith et al., 2022] Smith, E., Papadopoulos, D., Braschler, M., and Stockinger, K. (2022). Lillie: Information extraction and database integration using linguistics and learning-based algorithms. *Information Systems*, 105:101938.

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

[Stanovsky et al., 2018] 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.

[Vaswani et al., 2023] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

[Walter and Pinkal, 2006] Walter, S. and Pinkal, M. (2006). Automatic extraction of definitions from German court decisions. In Califf, M. E., Greenwood, M. A., Stevenson, M., and Yangarber, R., editors, *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 20–28, Sydney, Australia. Association for Computational Linguistics.

[Wrzalik and Krechel, 2021] 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.

[Wu and Weld, 2010] Wu, F. and Weld, D. S. (2010). Open information extraction using Wikipedia. In Hajič, J., Carberry, S., Clark, S., and Nivre, J., editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

[Yates et al., 2007] 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.

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