UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Can semantic similarities of words enhance common sense reasoning? A case study with prover E and SUMO.

Können semantische Ähnlichkeiten von Wörtern die Schlussfolgerungen des gesunden Menschenverstands verbessern? Eine Fallstudie mit Prover E und SUMO.

Bachelorarbeit

verfasst am
Institut für Software Engineering und Programming Languages

im Rahmen des Studiengangs
Informatik
der Universität zu Lübeck

vorgelegt von
Julian Britz

ausgegeben und betreut von
Prof. Dr. Diedrich Wolter

mit Unterstützung von
Moritz Bayerkuhnlein

Lübeck, den 06. Juli 2025

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

_____

Julian Britz

Zusammenfassung

Zusammenfassung

Abstract

Abstract

Acknowledgements

Ackbowledgements

Contents

1

Introduction

Automated theorem proving is a fundamental technique in formal logic and artificial intelligence. It enables the validation of logical statements based on a given set of axioms and inference rules. This thesis investigates whether incorporating frequently used axioms, referred to as core axioms, into a subset of the Adimen SUMO grammar improves proof success rates. The selected subset is determined by combining syntactic and semantic criteria to ensure both structural and conceptual relevance.

1.1 Problem Statement

- Given a formal logical grammar, such as Adimen SUMO, and a conjecture C, the objective is to identify a subset of axioms that maximizes the probability of proving C.
- Axioms are initially selected based on syntactic and semantic similarity measures, forming a set denoted as $\mathcal{A}_{\text{rel}}$.
- The hypothesis is that adding frequently used axioms, denoted as $\mathcal{A}_{\text{core}}$, improves proof success.

1.2 Entanglement Between Natural Language and Logical Grammar

- The relationship between natural language and formal logic is not a direct mapping but rather an entanglement, where both influence each other:

$$\mathcal{L}_{\text{NL}} \bowtie \mathcal{L}_{\text{logic}} \tag{1.1}$$

- Adimen SUMO, which is partially derived from WordNet, reflects structured knowledge originating from natural language.
- Logical structures often mirror patterns found in natural language, even though natural language remains ambiguous and context-dependent.
- Language models such as SBERT capture semantic relationships, supporting the idea that formal logical expressions exhibit deep structural similarities with natural language.

1.3 Necessity of Adding Core Axioms

- Semantic similarity alone is insufficient for finding proofs, as logical reasoning requires structured inference steps that cannot be captured by embeddings alone.
- Theorem provers rely on axioms that establish intermediate logical connections, which are crucial for the proof search process.

1.4 Operation of Theorem Prover E

The importance of intermediate axioms becomes evident when analyzing the inference mechanisms of the E theorem prover, which employs a given-clause algorithm:

1. Preprocessing: Converts input formulas into conjunctive normal form (CNF) and applies structural simplifications.
2. Clause Selection: Uses heuristics to prioritize clauses based on factors such as term complexity and length.
3. Inference Application: Performs superposition, paramodulation, and resolution to generate new clauses that contribute to the proof.
4. Simplification: Eliminates redundant clauses through rewriting, subsumption, and redundancy elimination.
5. Proof Search Termination: Stops either when an empty clause is derived, indicating proof completion, or when computational resources are exhausted.

Relying only on semantic closeness is inadequate because:

- Inference-based proofs require logical chaining. The prover does not simply retrieve related axioms but constructs sequential inference chains to derive contradictions or confirm validity.
- Clause selection is not solely based on semantic relevance. The prover applies heuristics that prioritize structural properties, meaning that semantically relevant axioms may not always contribute effectively to the proof.

Core axioms provide essential inference steps that enhance proof discovery and completion.

1.5 Formal Hypothesis

- Definitions:
  - $\mathcal{A}$: The set of all axioms in Adimen SUMO.
  - C: A conjecture to be proven.
  - $d : \mathcal{A} \times C \to \mathbb{R}^+$: A similarity function measuring syntactic and semantic closeness.
  - $\mathcal{A}_{\text{rel}}$: The subset of k nearest axioms to C:

$$\mathcal{A}_{\text{rel}} = \{A \in \mathcal{A} \mid A \text{ is among the } k \text{ closest axioms to } C \text{ w.r.t. } d\} \tag{1.2}$$

  - $\mathcal{A}_{\text{core}}$: A set of frequently used axioms in previous proofs.
  - $\mathcal{A}_{\text{enh}} = \mathcal{A}_{\text{rel}} \cup \mathcal{A}_{\text{core}}$: The enhanced axiom set.
  - $P(T, \mathcal{A}', C)$: A function that returns true if theorem prover T finds a proof of C using $\mathcal{A}'$, otherwise false.
- Hypothesis:

$$\forall C \in \mathcal{C}, \quad \Pr(P(T, \mathcal{A}_{\text{enh}}, C) = \text{True}) > \Pr(P(T, \mathcal{A}_{\text{rel}}, C) = \text{True}) \tag{1.3}$$

- This suggests that the probability of proving C increases when $\mathcal{A}_{\text{core}}$ is included.

1.6 Justification and Implications

- Since Adimen SUMO's grammar is intertwined with natural language structures, selecting axioms based on semantic embeddings, such as SBERT, captures implicit logical dependencies.
- Frequently used axioms function as structural anchors within this entanglement, improving proof discovery.
- The theorem prover benefits from both the nearest axioms and the core axioms, leading to a higher overall proof success rate.

This thesis aims to validate this hypothesis through empirical experiments, evaluating proof success rates across different axiom selection strategies and assessing the effectiveness of incorporating core axioms.

1.7 Structure of this Work

This thesis is organized into several chapters, each addressing key aspects of the research question and methodology. Below is an overview of the structure:

- **Chapter 2: Related Work**
  - Reviews existing research on theorem proving and axiom selection techniques.
  - Discusses previous applications of semantic similarity in logical reasoning.
- **Chapter 3: Preliminaries**
  - Presents fundamental concepts such as common sense reasoning, word embeddings, and theorem proving.
  - Explains the role of formal grammars and axiom selection in automated reasoning.
- **Chapter 4: Methodology**
  - Details the proposed method for improving theorem proving success through core axiom integration.
  - Describes the dataset, theorem prover configurations, and selection strategies.
- **Chapter 5: Experiments**

  - Presents the experimental setup, including benchmark datasets and evaluation metrics.
  - Reports on empirical findings comparing different axiom selection strategies.
  - Analyzes the impact of core axioms on proof success rates.
- **Chapter 6: Conclusion**
  - Recaps the research contributions and key insights.
  - Discusses potential future directions and open challenges.
- **Chapter 7: Future Work**
  - Future Work

2

Related Work

The selection of relevant axioms for automated theorem proving has been extensively studied, particularly in the context of large knowledge bases. Previous research has explored both syntactic and semantic approaches to determining axiom relevance. This section provides an overview of key contributions in this area.

- Context-specific axiom selection using large language models (**Schon2024**)
  - Traditional axiom selection techniques rely on syntactic properties and often do not account for the meaning embedded in symbol names.
  - Large language models provide an alternative approach by leveraging contextual similarity to determine relevant axioms.
  - This method aligns axiom selection with the context of the goal, leading to improved performance in commonsense reasoning tasks.
  - Experimental results indicate that selection methods based on large language models outperform purely syntax-driven approaches.
- Adimen-SUMO: A reengineered ontology for first-order reasoning (**Alvez2014**)
  - Adimen-SUMO is a modified version of the SUMO ontology, adapted to enhance compatibility with first-order automated theorem provers.
  - The reengineering process ensures that axioms are structured in a way that facilitates inference, avoiding ambiguities present in the original SUMO.
  - This ontology is widely used in evaluating theorem provers within the domain of commonsense reasoning.
- SInE: A syntactic approach to large theory reasoning (**Hoder2011**)
  - The Sumo INference Engine (SInE) is a widely used system for axiom selection in large theories.
  - It operates by iteratively selecting axioms that introduce symbols appearing in the conjecture, prioritizing those that contribute most to the proof process.
  - The approach significantly reduces the search space, making large knowledge bases such as OpenCYC more manageable for theorem provers.
  - SInE has been successfully applied in theorem proving competitions and has influenced subsequent axiom selection methods.
- Divvy: A meta-system for axiom relevance ordering (**Roederer2009**)

  - The Divvy system introduces two syntactic relevance orderings for axioms in automated theorem proving.
  - These orderings are used to prioritize axioms, allowing for more efficient proof attempts.
  - Experimental evaluations demonstrate that this method effectively reduces proof complexity and enhances theorem prover performance.
- SRASS: A semantic relevance axiom selection system (**Sutcliffe2007**)
  - Unlike syntactic approaches, SRASS selects axioms based on their semantic relevance to the conjecture.
  - The system employs a heuristic ordering mechanism that combines both semantic and syntactic relevance.
  - This method has been shown to reduce the number of required axioms while maintaining proof completeness.
- Automatic white-box testing of first-order logic ontologies (**Alvez2017**)
  - This work introduces a framework for systematically testing first-order logic ontologies.
  - The approach ensures that ontologies maintain logical consistency and do not contain redundant or contradictory axioms.
  - By applying automated testing techniques, the system improves the robustness and reliability of formal ontologies used in theorem proving.

3

Preliminaries

This chapter presents the fundamental concepts and methodologies relevant to this thesis. The focus is on common sense reasoning, word embeddings, formal grammars, theorem provers, and axiom selection strategies, all of which are essential for understanding the following chapters.

3.1 Common Sense Reasoning

Common sense reasoning is a fundamental component of human cognition. It enables reasoning about everyday situations, allowing individuals to infer unstated knowledge, resolve ambiguities, and generalize from incomplete information. Automated reasoning systems attempt to replicate this ability to improve inference and decision-making.

Definition and Scope

Common sense reasoning refers to the ability to infer implicit knowledge and derive reasonable conclusions from real-world scenarios. It involves:

- Recognizing unstated premises in logical arguments.
- Understanding cause-effect relationships in structured and unstructured data.
- Applying heuristics to solve problems under uncertainty.

Applications in Artificial Intelligence and Theorem Proving

Common sense reasoning is applied across various domains of artificial intelligence:

- Natural language understanding requires contextual awareness to interpret implicit meaning and resolve ambiguity.
- Automated reasoning relies on structured logical frameworks to infer conclusions from available premises.
- Decision support systems in fields such as law, medicine, and finance integrate knowledge-based reasoning to enhance interpretability and reliability.

Despite significant progress in AI, particularly in deep learning, existing models struggle with integrating structured, generalizable knowledge representations, making common sense reasoning a persistent challenge.

3.2 Word Embeddings

Word embeddings are numerical representations of words that capture semantic relationships within a continuous vector space. They are widely used in natural language processing to model meaning and improve text-based inference.

Concept and Motivation

Traditional representations such as one-hot encoding are sparse and fail to capture relationships between words. Word embeddings mitigate this limitation by mapping words into dense vector spaces, where similar words appear closer together.

Types of Word Embeddings

Several models have been developed to learn word representations:

- Word2Vec (**Mikolov2013**) uses a shallow neural network to learn word relationships via the Continuous Bag of Words (CBOW) and Skip-gram models.
- GloVe (**Pennington2014**) constructs embeddings based on word co-occurrence statistics in a given corpus.
- FastText (**Bojanowski2017**) extends Word2Vec by incorporating subword information, improving handling of rare and out-of-vocabulary words.
- Contextual embeddings such as BERT and GPT (**Devlin2019**) adapt word representations based on surrounding context, improving semantic disambiguation.

Application in This Work

Word embeddings are used in this thesis to measure semantic similarity between axioms and conjectures. By leveraging pre-trained embeddings such as Sentence-BERT (SBERT), the selection of axioms can be optimized to improve theorem proving success rates.

3.3 Grammars in Natural and Formal Languages

A grammar defines the structural rules governing a language, whether natural or formal. In theorem proving, grammars serve as a foundation for representing logical statements in a structured and interpretable format.

Types of Grammars

- Context-free grammars (CFGs) define syntactic structures in both natural language parsing and programming languages.
- First-order logic (FOL) grammars formalize logical expressions using predicates, functions, and quantifiers.
- Ontology-based grammars, such as SUMO (Suggested Upper Merged Ontology) (**Niles2001**), integrate linguistic and logical structures to improve knowledge representation.

Relevance to Automated Reasoning

Formal grammars play a crucial role in automated theorem proving:

- They ensure logical consistency in axioms and conjectures.
- They enable automated systems to parse and manipulate logical expressions systematically.
- They facilitate knowledge representation in formal ontologies such as SUMO and Adimen-SUMO.

3.4 Automated Theorem Proving

Automated theorem proving is a key component of formal logic and artificial intelligence, aiming to determine whether a given conjecture follows from a predefined set of axioms.

Fundamental Concepts

- Logical inference consists of deriving new statements from existing ones using formal inference rules.
- Resolution is a proof technique based on deriving contradictions, commonly used in first-order logic theorem provers.
- Refutation aims to demonstrate that a conjecture holds by attempting to derive a contradiction from its negation.

Prover E: A Resolution-Based Theorem Prover

Prover E is a widely used automated theorem prover based on the superposition calculus (**Schulz2013**). The system follows a structured process:

1. Preprocessing converts input formulas into conjunctive normal form (CNF).
2. Clause selection applies heuristics to prioritize clauses based on complexity and term structure.
3. Inference rules such as resolution, paramodulation, and superposition generate new clauses that contribute to the proof.
4. Simplification techniques eliminate redundant clauses through rewriting, subsumption, and redundancy elimination.

5. Proof search continues until an empty clause is derived, indicating proof completion, or until computational resources are exhausted.

Understanding the internal mechanisms of Prover E is crucial for optimizing axiom selection strategies.

3.5 Axiom Selection Strategies

Axiom selection is a critical challenge in automated reasoning, as theorem provers often struggle with large search spaces. The selection process determines which axioms are included in the proof search, influencing both efficiency and success rates.

Syntactic Approaches

- SInE (Sumo INference Engine) (**Hoder2011**) selects axioms that introduce symbols relevant to the conjecture, reducing the search space.
- Divvy (**Roederer2009**) ranks axioms based on their syntactic relevance, improving proof efficiency.

Semantic Approaches

- SRASS (**Sutcliffe2007**) selects axioms based on their semantic relevance to the conjecture, integrating meaning-based selection criteria.
- Axiom selection using large language models (**Schon2024**) aligns selected axioms with the context of the conjecture, improving commonsense reasoning in automated theorem proving.

Hybrid Approaches

- Combining syntactic and semantic techniques balances structural relevance with meaning-based similarity.
- The inclusion of frequently used axioms improves proof success by providing additional structural support to the inference process.

3.6 Summary

This chapter introduced key concepts including common sense reasoning, word embeddings, formal grammars, theorem proving, and axiom selection. These foundations provide the theoretical basis for the experimental work presented in later chapters, where different axiom selection strategies are evaluated in the context of automated theorem proving.

4

Methodology

Methodology.

5

Experiments

This chapter presents the experimental setup and evaluation of different axiom selection strategies in the context of automated theorem proving. The experiments focus on white-box testing, reengineering of prior work, comparative analysis of selection methods, and performance assessments across different theorem provers.

5.1 Construction of White-Box Tests

The methodology for constructing white-box tests follows the approach described by Álvez (**Álvez2017**). The tests are designed to assess the logical consistency and inference capabilities of theorem provers by introducing controlled variations in axiom selection.

- In the context of Adimen SUMO, falsity tests are created by taking specific statements or conditions and applying negation.
- The negated statements serve as the foundation for falsity tests, which play a crucial role in evaluating logical proof systems.
- A theorem prover such as Prover E executes a process known as refutation.
- Through refutation, the prover attempts to demonstrate the inconsistency or falsehood of the original statements.
- This is achieved by attempting to derive a contradiction from the negated statements, thereby indicating that the original (non-negated) statements cannot all be true simultaneously.

5.2 Reengineering of Prior Work (**Schon2024**)

The evaluation of axiom selection techniques includes a reengineering of the results presented in (**Schon2024**). The goal is to validate the prior findings and assess the impact of alternative selection methods.

- The prover results are categorized into:
  - Timeout: Cases where the prover exhausted computational resources without finding a proof.

- Proof found: Cases where the prover successfully derived a proof.
- Gave up: Cases where the prover terminated without reaching a conclusion.

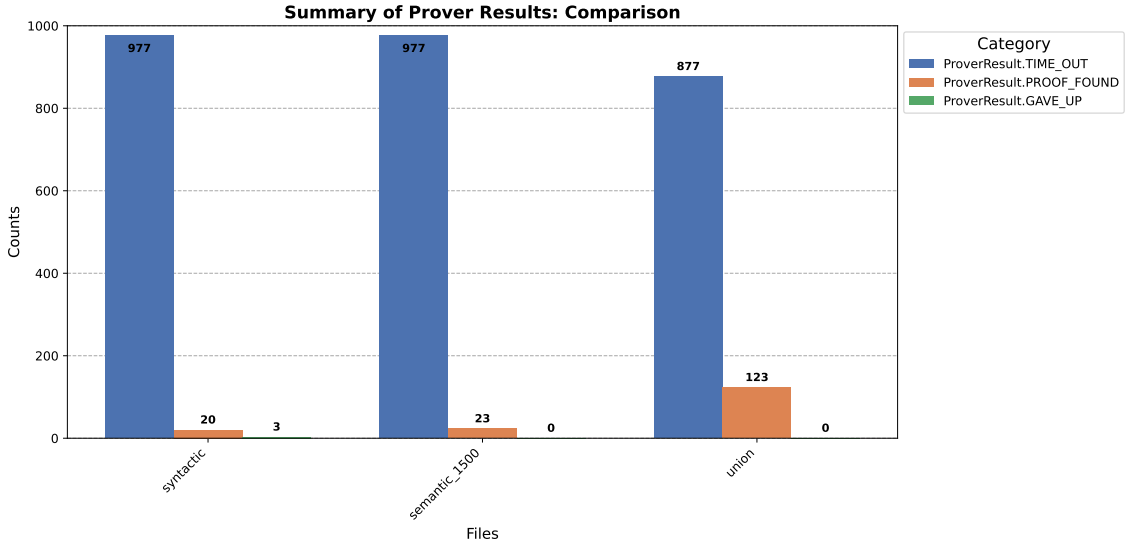

Figure 5.1: Reengineering of results from (**Schon2024**)

- Theorem prover performance is evaluated based on:
  - A high percentage of test cases resulted in timeouts or unsuccessful proof attempts, illustrating the difficulty of proving conjectures without additional axioms.
  - The success rate for proof discovery remained low, suggesting that the standard mode lacks sufficient knowledge for effective inference.
  - The dataset incorporates multiple axiom selection techniques, including syntactic, semantic-based, and union-based selection.
- The distribution of results shows:
  - The prover executed 977 test cases, applying each axiom selection approach independently.
  - The number of successfully proven conjectures was relatively low, reinforcing the importance of improved axiom selection techniques.
  - Some conjectures had no successful proofs, highlighting weaknesses in current selection strategies.
- Implications for further research:
  - The findings suggest that semantic similarity alone is insufficient for effective axiom selection.
  - Further experiments should explore whether the integration of core axioms leads to improved proof success rates.
  - Analyzing the relationship between axiom complexity and proof success may provide additional insights into theorem proving behavior.

5.3 Analysis of Reengineering

The reengineered results are analyzed to better understand proof complexity and selection effectiveness.

– The mean variable count in proofs is examined.

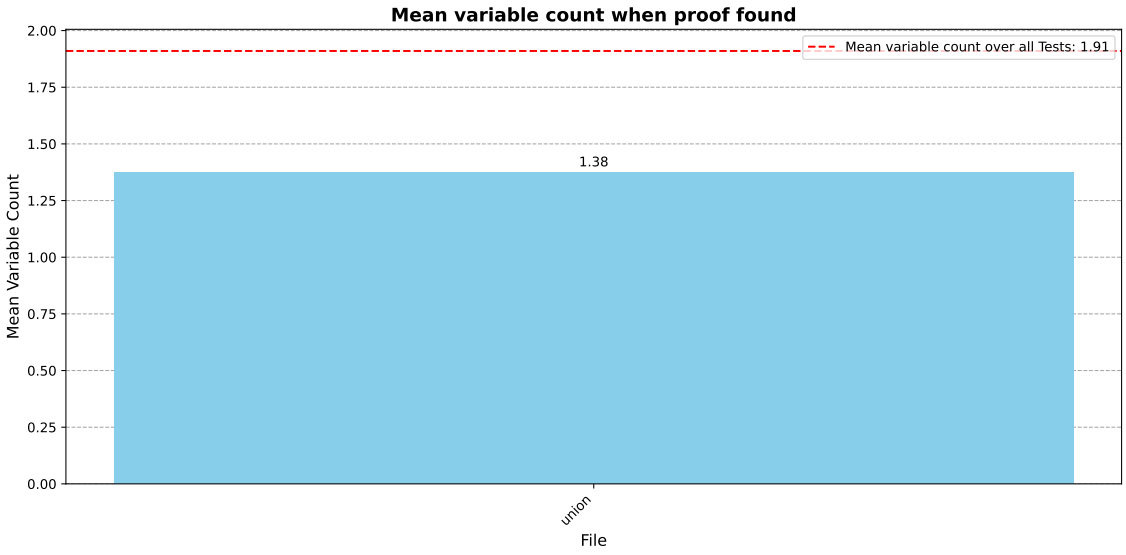

Figure 5.2: Variable count in proofs

– The count of special symbols in proofs is analyzed.

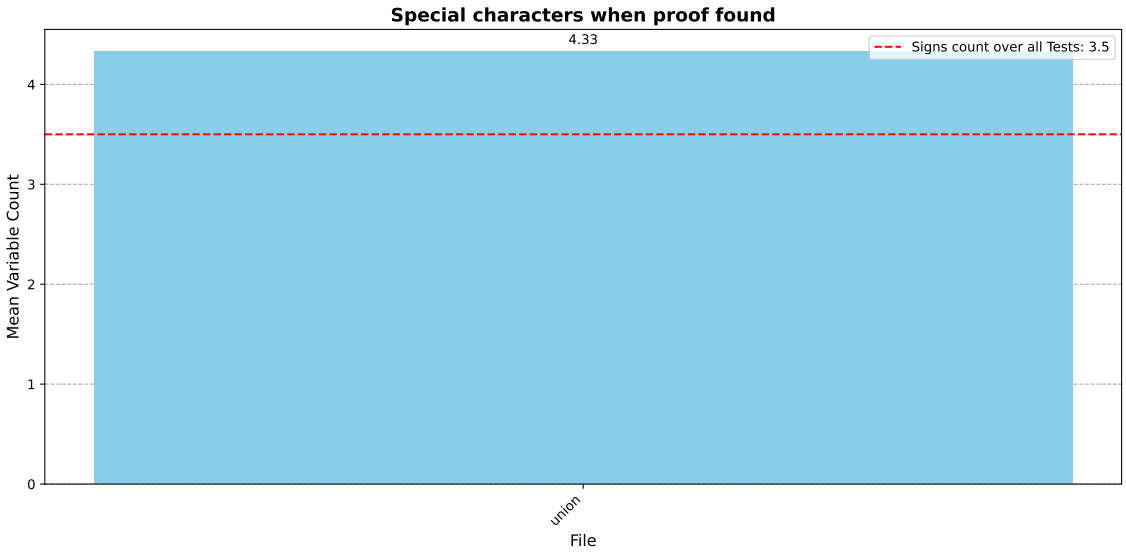

Figure 5.3: Special character count

– The character count across all test cases is examined.

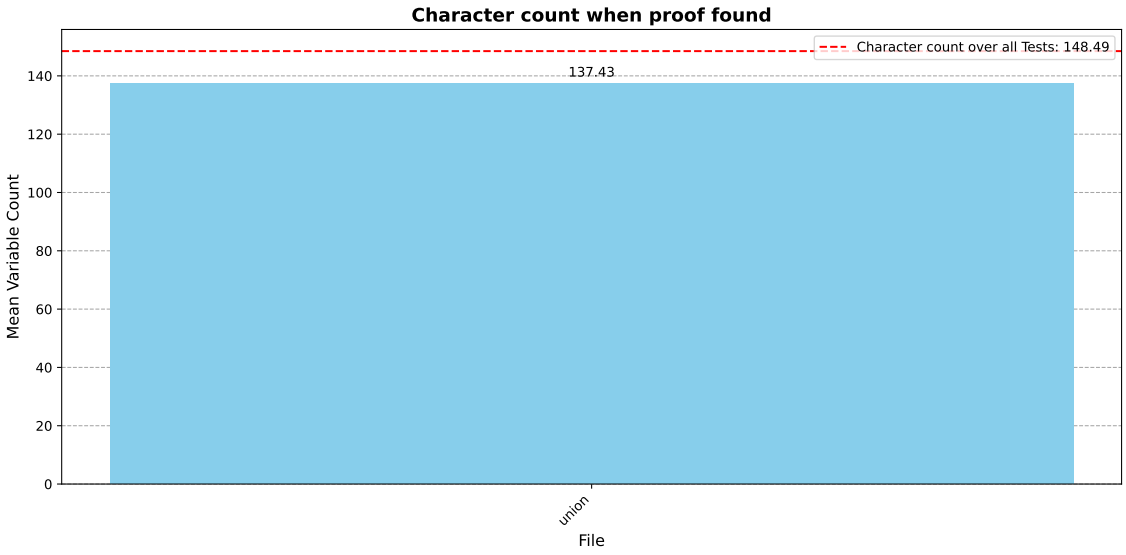

Figure 5.4: Character count distribution

– The cosine similarity between axioms and conjectures is analyzed to determine whether similarity influences proof success.

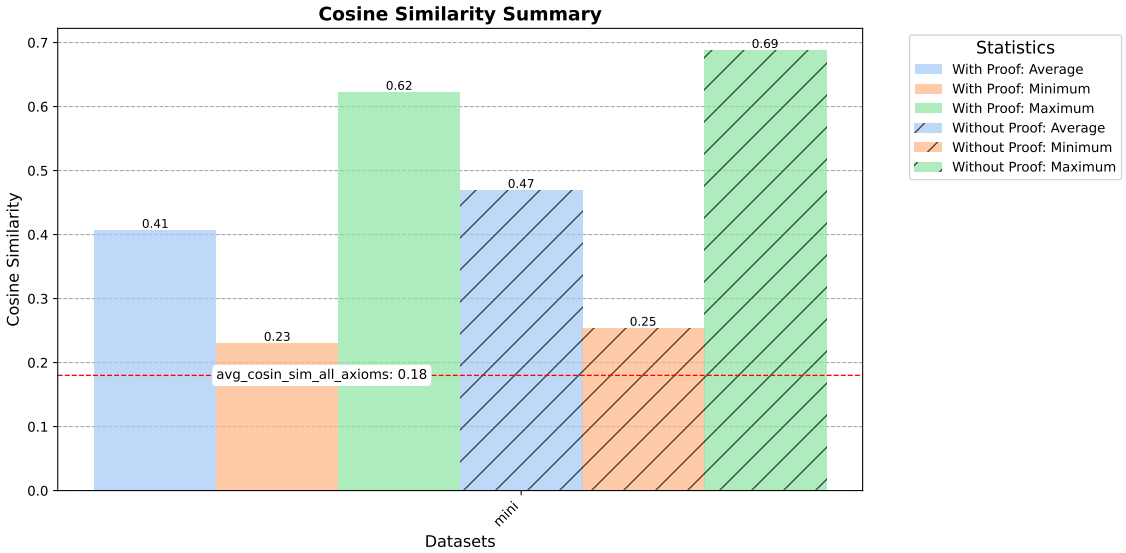

Figure 5.5: Cosine similarity distribution

The results suggest that provable conjectures often have lower axiom similarity. This indicates that proving a conjecture may require bridging different concepts rather than simply refining closely related axioms.

5.4 Integration of Core Axioms

The integration of core axioms into the selection process is evaluated.

- The results of theorem proving with core axioms are examined.

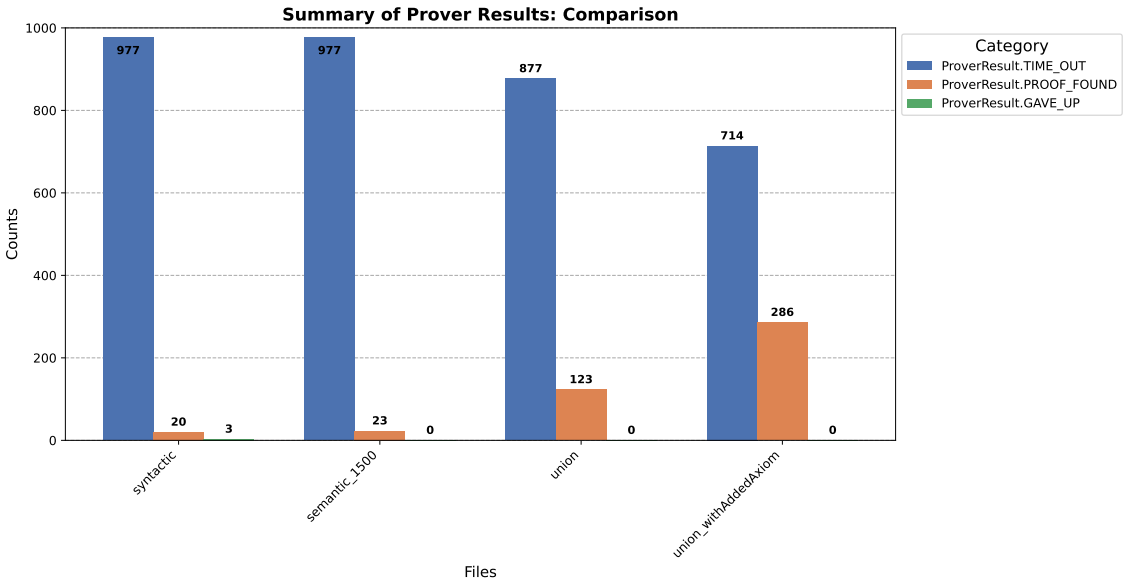

Figure 5.6: Summary of prover results with core axioms

- Cosine similarity is compared across different selection methods.

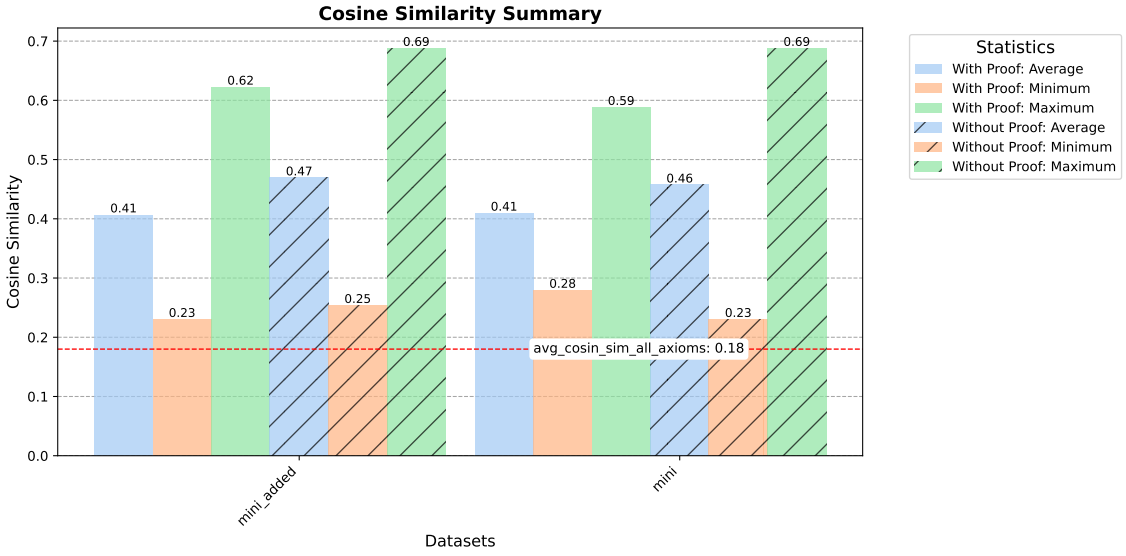

Figure 5.7: Cosine similarity comparison

5.5 Comparison of Large Language Models

The performance of different large language models in axiom selection is assessed.

- The results of theorem proving with different models are analyzed.

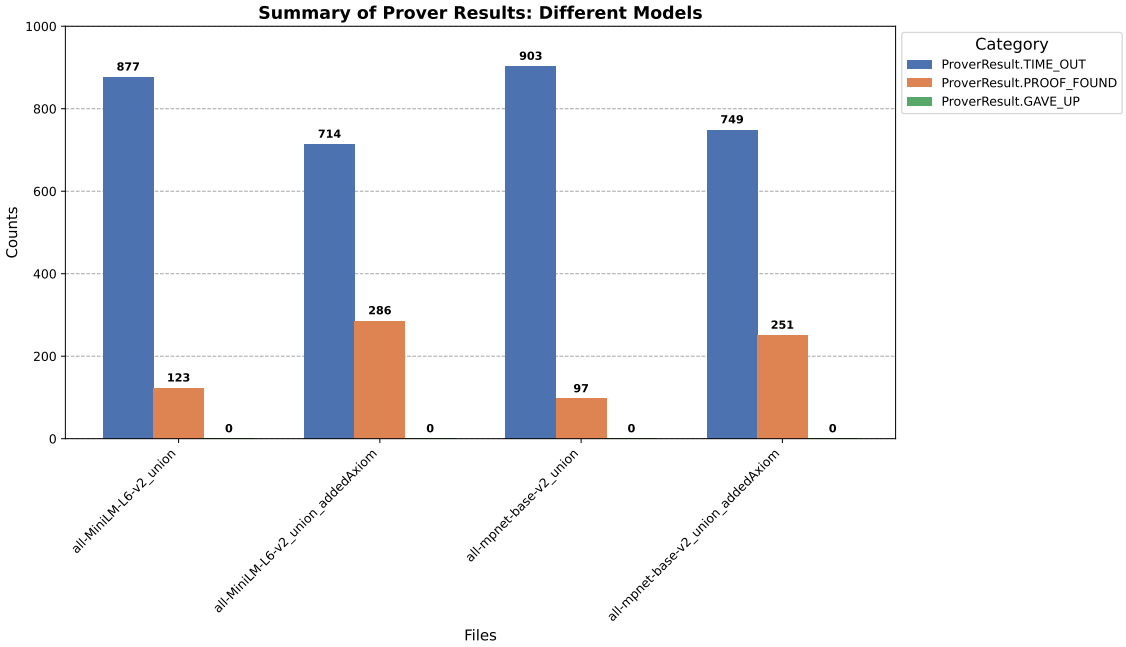

Figure 5.8: Summary of prover results with different models

- The cosine similarity of selected axioms is compared.

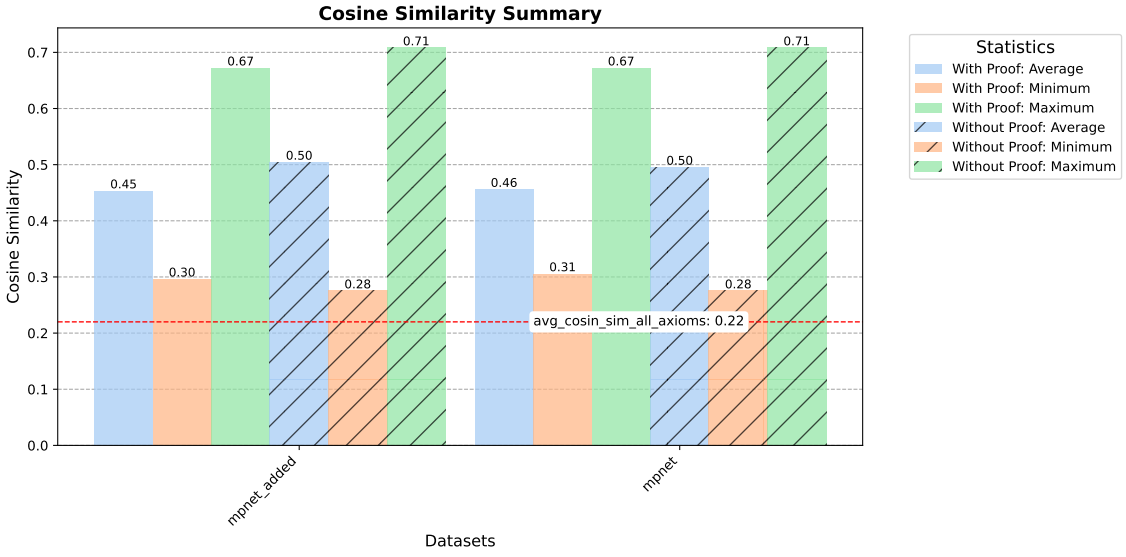

Figure 5.9: Cosine similarity in different models

5.6 Evaluation of Theorem Prover Configurations

The effect of different theorem prover configurations is analyzed.

- The performance of Satauto mode is examined.

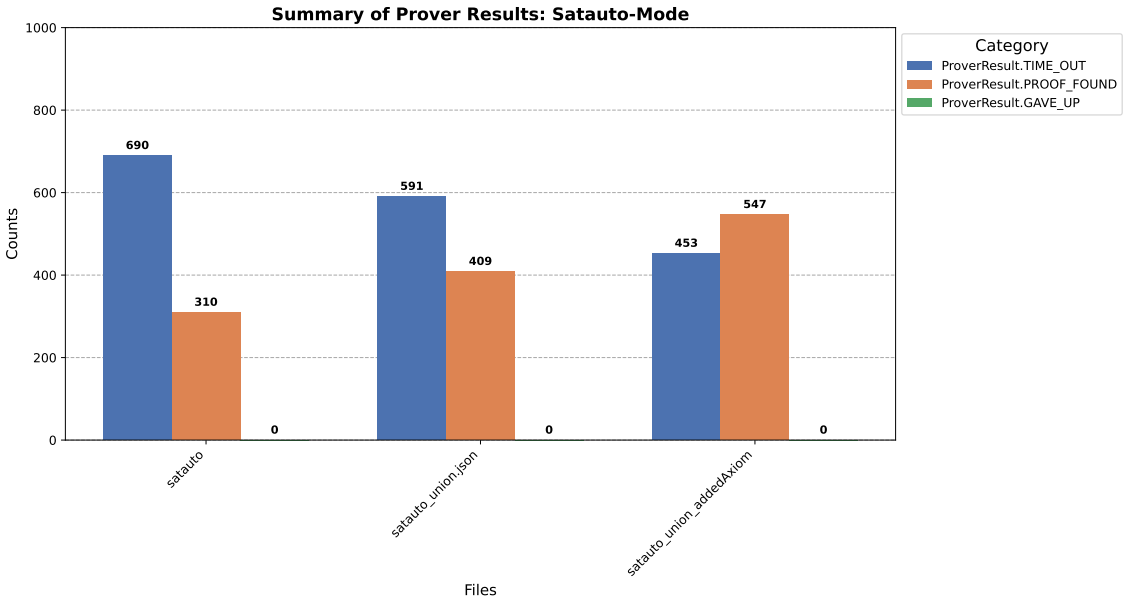

Figure 5.10: Summary of prover results in Satauto mode

- The effect of auto mode, which uses SInE and search heuristics, is assessed.

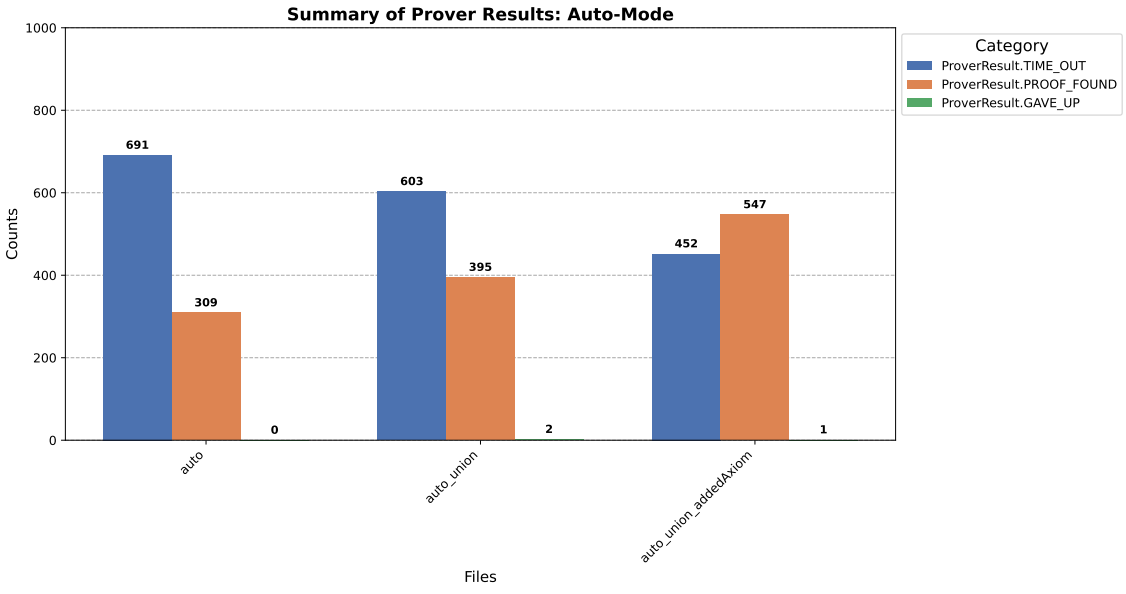

Figure 5.11: Summary of prover results in Auto mode

6

Conclusion

Conclusion.

7

Future work

Future work.

$$