UNIVERSITÄT ZU LÜBECK

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

IM FOCUS DAS LEBEN

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

Acknowledgements

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. The process involves:

- Selecting axioms based on syntactic and semantic similarity, forming a set denoted as $\mathcal{A}_{\text{rel}}$.
- Testing whether adding frequently used axioms, $\mathcal{A}_{\text{core}}$, improves proof success.

Even though there is no direct mapping between natural and logical language, we assume, that there is an entanglement between them:

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

Arguments for this assertion are that logical langugage is human made and therefore inevitably connected to human and consequently natural language. Also grammers like Adimen SUMO are partially derived by lexical-semantic networks that capture relationships between words of natural language. Making use of this entanglement the selection process of theorem proovers can be optimized by reducing the number of relevant axioms. Theorem prover E follows a refutational proof procedure using the superposition calculus. The proof search is based on a modified given-clause algorithm, consisting of several key steps:

1. Initialization: The input problem is converted into clausal normal form (CNF) and divided into two sets: unprocessed clauses U and processed clauses P.

2. Clause Selection: A clause g is selected from U based on heuristics, referred to as the given clause, and is then simplified against P.
3. Simplification: The given clause undergoes forward simplification using processed clauses. This step removes redundant literals, applies rewriting rules, and ensures that terms are arranged in a canonical order.
4. Inference: The prover applies the superposition calculus, combining g with clauses in P to generate new clauses. These new clauses are simplified and added to U if they are neither trivial nor redundant.
5. Saturation or Termination: The process stops when:
    - The empty clause $\bot$ is derived, confirming the proof.
    - The clause set is saturated, meaning no further inferences can be made.
    - A predefined resource limit, such as time or memory, is reached.

(**Schulz2019**)

Axioms play a crucial role in simplification. While the prover cannot arbitrarily discard axioms, as they form the foundation of logical reasoning, they are essential for clause rewriting and subsumption.

1.2 Hypothesis

To test whether adding frequently used axioms improves proof success, we define the key elements of the problem:

- $\mathcal{A}$ represents the full set of axioms available in Adimen SUMO.
- C is the conjecture we want to prove.
- d is a function that measures how similar an axiom is to the conjecture, based on both syntax and meaning.
- $\mathcal{A}_{\text{rel}}$ is the set of k axioms that are most similar to the conjecture according to d:

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

- $\mathcal{A}_{\text{core}}$ is a collection of axioms that have been used frequently in past proofs.
- The enhanced axiom set, $\mathcal{A}_{\text{enh}}$, is created by combining the most similar axioms with the frequently used ones:

$$\mathcal{A}_{\text{enh}} = \mathcal{A}_{\text{rel}} \cup \mathcal{A}_{\text{core}} \tag{1.3}$$

- $P(T, \mathcal{A}', C)$ is a function that checks if the theorem prover T is able to prove C using a given set of axioms $\mathcal{A}'$. If the proof is successful, the function returns true; otherwise, it returns false.

The hypothesis we test is:

$$\forall C \in \mathcal{C}, \quad \Pr(P(T, \mathcal{A}_{\text{enh}}, C) = \text{True}) > \Pr(P(T, \mathcal{A}_{\text{rel}}, C) = \text{True}) \tag{1.4}$$

In other words, the likelihood of proving a conjecture is higher when frequently used axioms are included along with the most similar ones.

1.3 Justification and Implications

Adimen SUMO's grammar is closely connected to natural language structures. This means that selecting axioms based on semantic embeddings, such as SBERT, helps capture underlying logical dependencies that might not be immediately apparent. Frequently used axioms act as structural anchors within this system, making it easier for the theorem prover to navigate the proof search space and discover relevant inference steps. By combining the most relevant axioms with those that have been frequently used in past proofs, the theorem prover benefits from both semantic alignment and structural consistency. This increases the overall proof success rate, as the prover has access to axioms that not only appear relevant but have also proven useful in previous reasoning tasks.

This thesis tests this hypothesis through empirical experiments, measuring proof success rates across different axiom selection strategies. The goal is to determine how much incorporating core axioms improves theorem proving performance.

1.4 Structure of this Work

This thesis is structured into several chapters, each addressing different aspects of the research problem and methodology. It begins with an overview of existing research, followed by the theoretical foundations, the proposed approach, experimental evaluations, and concluding discussions.

Chapter 2 provides a review of related work. It covers previous research on theorem proving and axiom selection techniques, as well as how semantic similarity has been applied in logical reasoning.

Chapter 3 introduces fundamental concepts necessary to understand the core of this research. It discusses topics such as common sense reasoning, word embeddings, and theorem proving. Additionally, it explains the role of formal grammars and axiom selection in automated reasoning.

Chapter 4 focuses on the experimental setup and evaluation. It outlines the benchmark datasets, describes the evaluation metrics, and presents the results of different axiom selection strategies. The impact of incorporating core axioms into theorem proving is analyzed based on empirical findings.

Chapter 5 summarizes the main contributions of this work and highlights key insights gained from the experiments. It also discusses potential future directions and remaining challenges in axiom selection and automated reasoning.

Finally, Chapter 6 explores ideas for future research, suggesting possible extensions and refinements to the proposed approach.

Each chapter builds upon the previous ones, providing a structured progression from theoretical background to practical implementation and evaluation.

2

Related Work

A significant part of this thesis builds upon the work of Schon (**Schon2024**), which explores context-specific axiom selection using large language models. Traditional axiom selection methods primarily rely on syntactic properties, often ignoring the meaning embedded in symbol names. Schon proposes an alternative approach by leveraging large language models to determine relevant axioms based on contextual similarity. This method aligns axiom selection with the goal's context, improving performance in commonsense reasoning tasks. Experiments show that this approach outperforms purely syntactic selection methods, demonstrating the potential of semantic similarity for guiding theorem provers.

The findings of Schon's work play a crucial role in this thesis, as a major aspect of the proposed approach involves using semantic embeddings for axiom selection. However, while Schon focuses on direct context-based selection, this thesis investigates the impact of incorporating frequently used axioms into the selection process. The goal is to determine whether combining core axioms with those chosen based on semantic similarity enhances proof success rates.

Beyond Schon's work, several other studies have contributed to axiom selection and theorem proving. Adimen-SUMO, developed by Álvarez et al. (**Alvez2014**), is a reengineered version of the SUMO ontology designed to improve compatibility with first-order theorem provers. The restructuring ensures that axioms are formulated in a way that facilitates inference, reducing ambiguities present in the original SUMO. This ontology serves as a key resource in evaluating theorem provers, including those used in this thesis.

Syntactic axiom selection has also been widely explored. Hoder and Voronkov (**Hoder2011**) introduced SInE, a system that selects axioms based on their syntactic relevance to the conjecture. By iteratively choosing axioms that introduce symbols appearing in the conjecture, SInE reduces the search space, making large knowledge bases more manageable. Similarly, Roederer et al.

Another relevant contribution comes from Álvarez et al. (**Alvez2017**), who introduced a framework for systematic white-box testing of first-order logic ontologies. Their work focuses on ensuring logical consistency and eliminating redundant or contradictory axioms. Automated testing techniques from this study provide insights into ontology refinement, which is relevant when evaluating the quality of selected axioms in theorem proving.

In summary, Schon's research serves as the primary foundation for this thesis, providing a framework for semantic axiom selection that is further extended through the integration of frequently used axioms. Other related studies on syntactic axiom selection, ontology structuring, and logical consistency contribute valuable insights, supporting the refinement of the proposed approach.

3

Preliminaries

This chapter introduces fundamental concepts relevant to this work. It covers common sense reasoning, word embeddings, formal grammars, automated theorem proving, and axiom selection strategies. These topics form the theoretical foundation for the experiments and analyses presented in later chapters.

3.1 Common Sense Reasoning

Common sense reasoning is an essential part of human cognition, allowing individuals to understand everyday situations, infer unstated knowledge, and make reasonable decisions even when information is incomplete. Automated reasoning systems attempt to replicate this ability to improve inference and decision-making.

Common sense reasoning involves recognizing unstated premises in logical arguments, understanding cause-effect relationships, and applying heuristics to solve problems under uncertainty. It plays a key role in artificial intelligence, particularly in natural language understanding, where context is crucial for interpreting meaning, and in automated reasoning, where structured logical frameworks help infer conclusions from available premises. Applications in law, medicine, and finance further highlight the importance of integrating knowledge-based reasoning for interpretability and reliability. Despite advancements in AI, deep learning models struggle with incorporating structured and generalizable knowledge, making common sense reasoning an ongoing challenge.

Interpreting a sentence often involves more than just processing its words—it requires incorporating background knowledge to make sense of implicit information. Take, for example, the Winograd Schema involving John and Billy viewing a stage (**Levesque2012**). While the sentence itself is simple, understanding it correctly depends on a series of assumptions that humans naturally make when constructing a mental model of the situation (**Bayerkuhnlein2023**).

A stage is something meant to be viewed. In a typical setting, viewers remain stationary, and obstructions can block their line of sight. Humans are physical entities, meaning they cannot see through solid objects. Both John and Billy are human names, so the sentence suggests a scene where one person is obstructing the other's view. The interpretation of who is blocking whom depends on the additional context provided by the sentence, but the reasoning process remains the same: we piece together knowledge about how the world works to arrive at a coherent understanding.

These intuitive assumptions form the foundation of common sense reasoning. The process aligns with the theory of mental models, which suggests that humans construct internal representations of a scenario to evaluate possible interpretations (**JohnsonLaird1989**). In the case of ambiguous pronouns, this internal model helps determine which assignment makes the most logical sense. This ability to build structured mental representations is what allows humans to effortlessly resolve ambiguities that challenge AI systems.

3.2 Word Embeddings

Word embeddings represent words as numerical vectors in a continuous space, capturing semantic relationships based on their contextual usage. Unlike traditional representations such as one-hot encoding, embeddings enable words with similar meanings to be positioned closer together in the vector space, allowing models to generalize across linguistic variations.

Different techniques have been developed to generate word embeddings, each with unique properties and advantages. Word2Vec (**Mikolov2013**) is one of the foundational models, introducing two architectures: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts a word based on its surrounding context, while Skip-gram does the inverse, predicting context words given a target word. This approach efficiently captures both syntactic and semantic relationships. FastText (**Bojanowski2017**) extends Word2Vec by incorporating subword information, representing words as bags of character n-grams. This allows it to handle rare words and out-of-vocabulary terms better than standard word-based models. GloVe (**Pennington2014**), in contrast, constructs embeddings by factorizing a word co-occurrence matrix, capturing both local and global semantic information.

The effectiveness of word embeddings depends on several factors, including the size of the context window, the type of training data, and the inclusion of subword information.

Beyond standard word embeddings, contextual embeddings such as BERT and GPT (**Devlin2019**) further refine representation by considering the full sentence structure, allowing word meanings to dynamically change based on context. This advancement is particularly relevant in cases where words have multiple meanings depending on their use in different sentences.

In this work, word embeddings are leveraged to determine the semantic similarity between axioms and conjectures. By using pre-trained models like Sentence-BERT (SBERT), it becomes possible to enhance axiom selection in theorem proving, ensuring that the selected axioms are not only structurally relevant but also semantically meaningful.

3.3 Grammars in Natural and Formal Languages

Grammars define the structural rules of a language, whether it is a natural language like English or a formal system used in logic. In theorem proving, grammars are essential for structuring logical statements in a way that machines can interpret and manipulate.

Different types of grammars serve various purposes. Context-free grammars (CFGs) define syntactic structures in natural language processing and programming languages. First-order logic (FOL) grammars represent logical expressions using predicates, functions, and quantifiers. Ontology-based grammars, such as SUMO (Suggested Upper Merged Ontology) (**Niles2001**), combine linguistic and logical structures to improve knowledge representation.

Grammars are crucial in theorem proving because they enforce logical consistency, allow automated systems to process logical expressions systematically, and provide a structured representation of knowledge in formal systems like SUMO and Adimen-SUMO.

3.4 Automated Theorem Proving

Automated theorem proving aims to determine whether a given conjecture follows logically from a set of axioms. It is a key component of formal logic and artificial intelligence.

Theorem provers rely on different proof strategies. Logical inference derives new statements from existing ones using formal rules. Resolution is a proof technique that derives contradictions, commonly used in first-order logic provers. Refutation works by attempting to show that a conjecture holds by deriving a contradiction from its negation.

Prover E is a widely used theorem prover based on the superposition calculus (**Schulz2013**). It follows a structured process where input formulas are first converted into conjunctive normal form (CNF). The prover then selects clauses heuristically, applies inference rules such as resolution and superposition to generate new clauses, simplifies redundant ones, and continues searching until either an empty clause is derived, proving the conjecture, or computational resources are exhausted. Understanding these mechanisms is important for optimizing axiom selection.

3.5 Axiom Selection Strategies

Theorem provers often struggle with large search spaces, making axiom selection a critical challenge. The process of selecting axioms influences proof efficiency and success rates.

3.6 Syntax-Based Selection: SInE

The SInE selection strategy (**Hoder2011**) is a trigger-based approach designed to efficiently select axioms in automated theorem proving. Instead of selecting axioms based on fixed rules, it dynamically determines relevance using a trigger relation. This relation ensures that an axiom is only selected if at least one of its symbols appears less frequently than or as frequently as all other symbols in the axiom. This prevents highly common symbols from triggering every axiom they appear in, reducing unnecessary selections.

The selection process starts by identifying symbols that appear in the conjecture to be proven. These symbols are considered directly triggered. If a triggered symbol appears in an axiom and satisfies the trigger relation, the axiom is selected, and its other symbols become triggered in the next selection step. This process continues recursively up to a defined depth k, ensuring that only the most relevant axioms are included.

One limitation of this approach arises when an axiom contains symbols with nearly equal occurrence counts. In such cases, only the least frequent symbol can trigger the axiom, potentially excluding useful axioms. To address this, a benevolence parameter b is introduced, allowing symbols that occur up to b times more frequently than the least common symbol to also act as triggers. This adjustment provides more flexibility in axiom selection.

3.7 Vector-Based Selection

Syntax-based selection methods like SInE do not consider the meaning of symbol names, which can lead to the loss of useful information. This is particularly problematic in commonsense reasoning, where symbol names often carry semantic meaning that could improve selection. To address this, vector-based selection methods leverage word embeddings to quantify the semantic similarity between axioms and the goal.

As seen above, word embeddings are a common technique in natural language processing, based on the distributional hypothesis (**Miller1991**). This hypothesis states that words appearing in similar contexts tend to have similar meanings. By training neural networks on large text corpora, words are mapped into a continuous vector space, where their geometric relationships capture semantic similarity. Cosine similarity is commonly used to measure the closeness of two word vectors, with values ranging from -1 (completely dissimilar) to 1 (identical).

Definition 5.1 (Cosine similarity of two vectors (Schon2023)). Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ be non-zero vectors. The cosine similarity of $\mathbf{u}$ and $\mathbf{v}$ is defined as:

$$\cos\,\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||||\mathbf{v}||}$$

Schon (**Schon2023**) introduces a statistical approach for axiom selection that relies on word embeddings to represent axioms as vectors. The idea is to encode axioms into a continuous vector space where semantically similar axioms are mapped closer together. This is done in a preprocessing step, where each axiom in the knowledge base is transformed into a vector representation using a word embedding model.

When selecting axioms for a given goal G, the goal is vectorized in the same way as the axioms in the knowledge base. The selection process then identifies the n axioms whose vector representations are most similar to that of G. The similarity between vectors is measured using cosine similarity, which quantifies how closely two vectors align regardless of their magnitude.

Formally, given a knowledge base KB, a goal G, and a vectorization function f that maps axioms to $\mathbb{R}^n$, the vector-based selection process selects the n axioms in KB that have the highest cosine similarity to G. The selected axioms form a subset of KB where no other axiom outside this subset has a higher similarity to G than the least similar axiom within the selected set.

This approach ensures that axioms most relevant to the goal, based on their semantic similarity, are prioritized during the selection process.

3.8 SeVEn: Sentence-Based Vector Encoding

Sentence embeddings function similarly to word embeddings, as they are also generated by training neural networks on large text corpora. However, while word embeddings focus on individual words, sentence embeddings capture the meaning of entire sentences or even full documents (**Kiros2015SkipThought**). This ability makes them particularly useful for encoding more complex relationships between axioms.

Following this idea, the SeVEn approach builds upon vector-based selection by modifying how axioms are represented. Instead of encoding individual symbol names, each axiom is first translated into a sentence that captures its semantic meaning. This sentence is then encoded into a vector using a sentence embedding model. The overall process remains similar to standard vector-based selection, but the vectorization method is adapted to work at the sentence level. Given a knowledge base KB, where each axiom A is transformed into a sentence S using a function t, the sentence embedding function f then maps S into a vector representation $v_S(A)$. The full sentence-based vector representation of the knowledge base is denoted as $V_S(KB)$, where each axiom is now represented as a sentence embedding.

This adaptation allows SeVEn to leverage the richer semantic information captured in sentence embeddings while maintaining the core structure of vector-based selection.

3.9 Combining Vector-Based and Syntactic Selection

.

To address this issue, hybrid selection strategies combine statistical and syntactic approaches. An example of this is Similarity SInE (**Furbach2019WordEmbeddings**), which extends the original SInE selection by allowing not just the least frequent symbol to trigger an axiom but also symbols with a high enough cosine similarity to the trigger. This modification ensures that semantically related axioms can still be included in the selection process.

Building on this idea, (**Schon2023**) introduced a hybrid approach that integrates SeVEn with the syntactic selection method of SInE. The process starts by expanding the goal with axioms selected through SeVEn. Then, SInE is applied to this extended set of conjectures, selecting additional axioms based on its trigger-based mechanism.

Using formal notation, let $\mathrm{sine}(KB, \{A_1, \ldots, A_n\})$ represent the axioms selected by SInE for a set of conjectures, and let $\mathrm{seven}(KB, G)$ denote the axioms selected by SeVEn for a goal G. The combined selection strategy is then defined as:

$$\mathrm{union\_select}(KB, G) = \mathrm{sine}(KB, G \cup \mathrm{seven}(KB, G)) \tag{3.1}$$

This formulation allows for flexibility in axiom selection, as different selection strategies can be incorporated depending on the specific needs of the reasoning task.

4

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.

4.1 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 constructed by taking specific statements or conditions and applying negation. These negated statements form the basis for falsity tests, which are essential for evaluating logical proof systems. A theorem prover such as Prover E processes these tests through a method called refutation. The goal of refutation is to show that the original statements are inconsistent or false. This is done by attempting to derive a contradiction from the negated statements. If a contradiction is found, it confirms that the original, non-negated statements cannot all be true at the same time.

4.2 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.

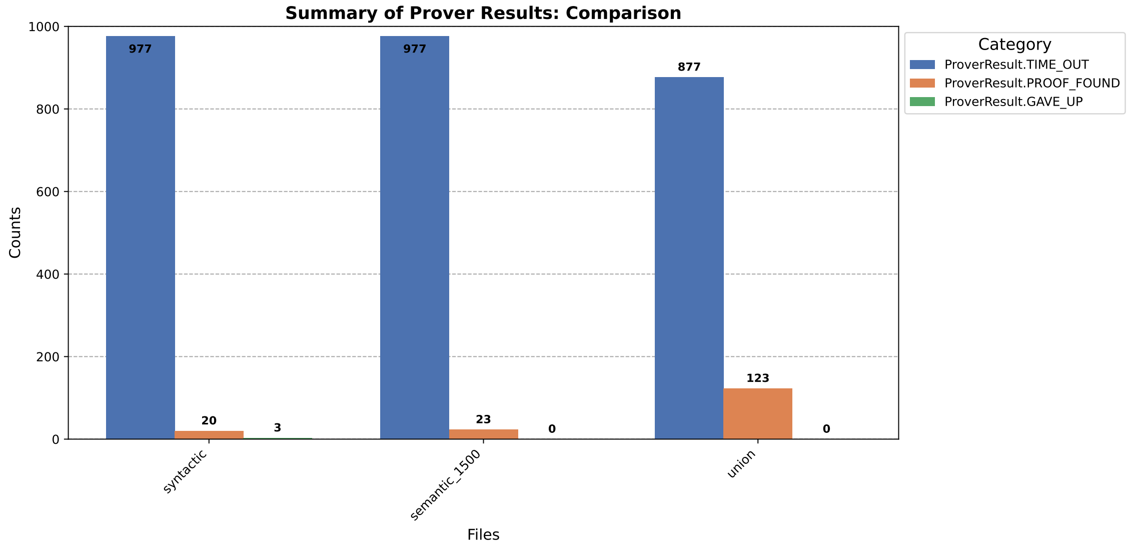

Figure 4.1: Reengineering of results from (**Schon2024**)

- 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.

4.3 Analysis of Reengineering

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

- The mean variable count in proofs is examined.

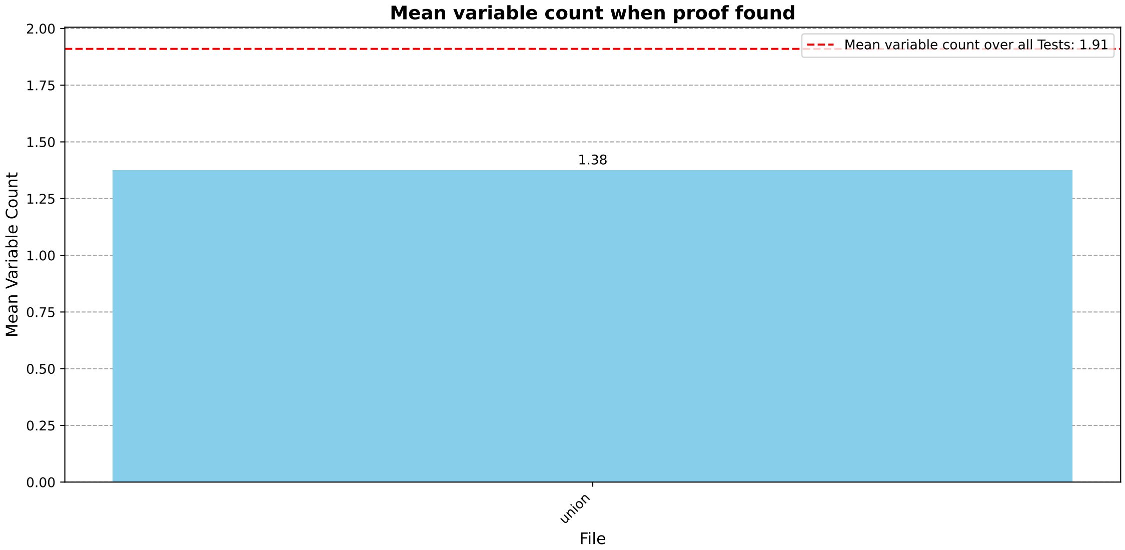

Figure 4.2: Variable count in proofs

- The count of special symbols in proofs is analyzed.

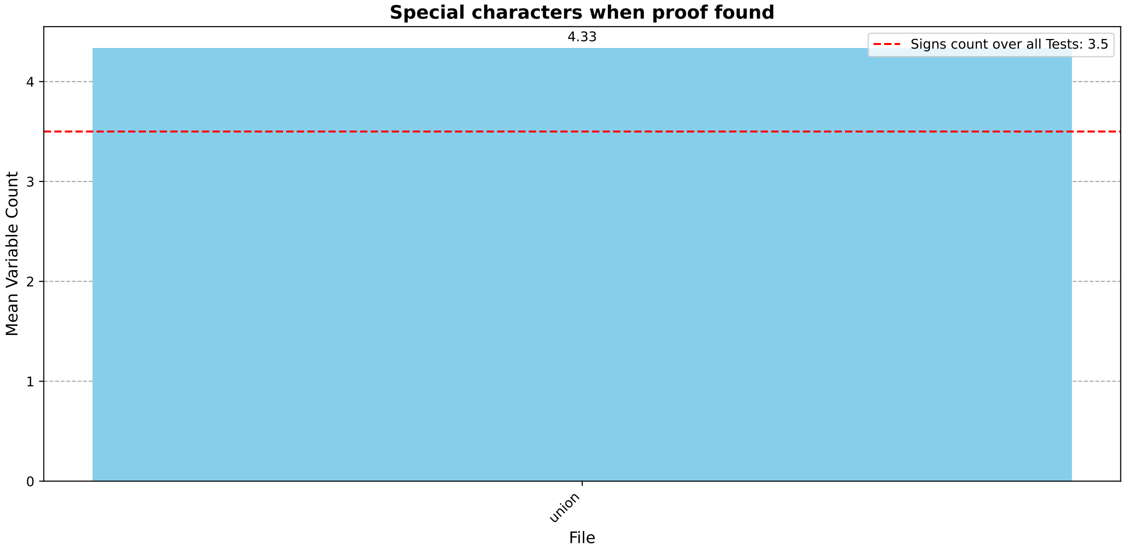

Figure 4.3: Special character count

- The character count across all test cases is examined.

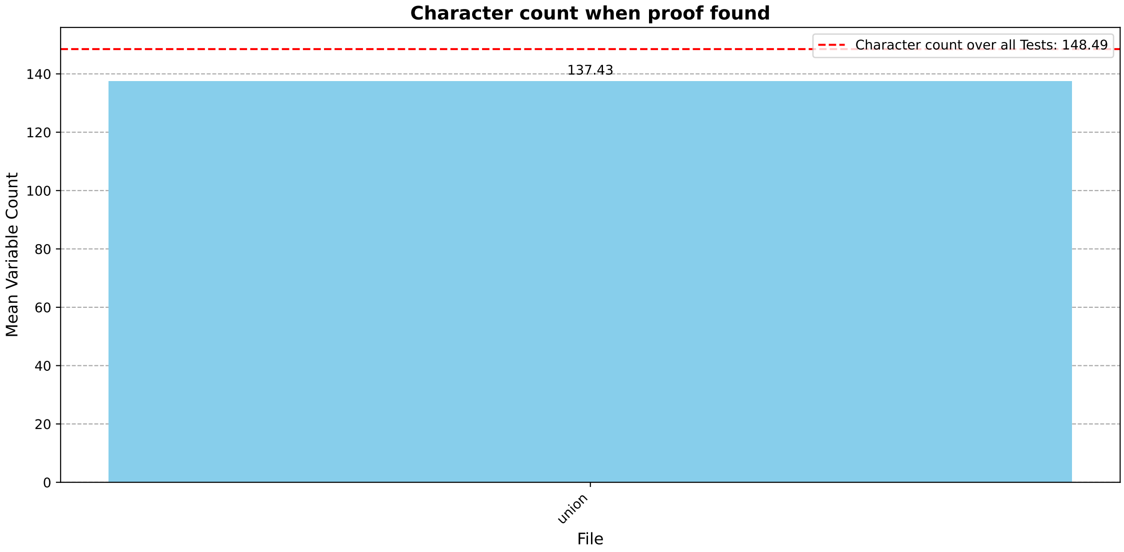

Figure 4.4: Character count distribution

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

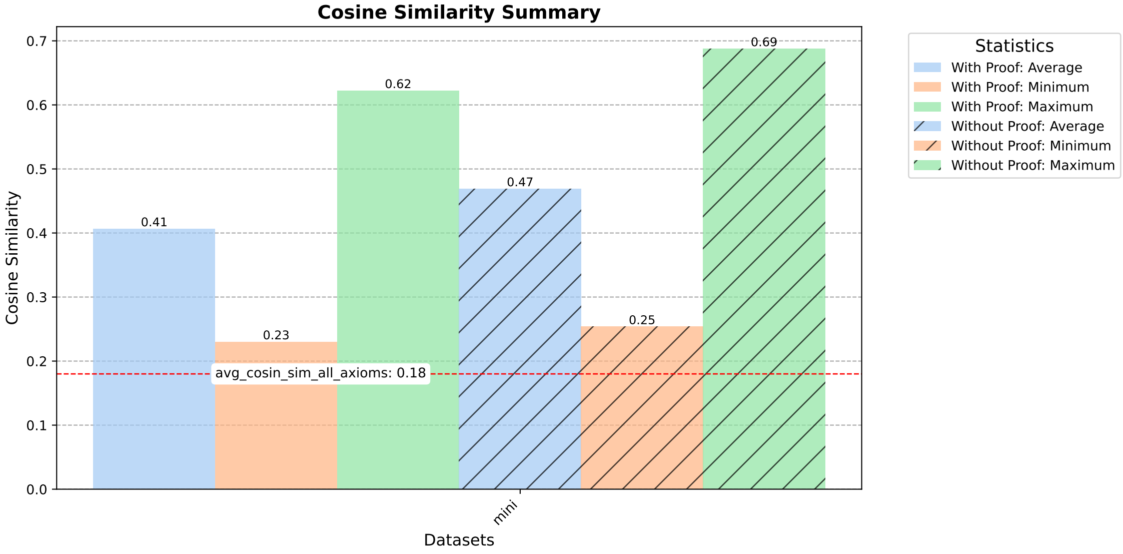

Figure 4.5: Cosine similarity distribution

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.

4.4 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.

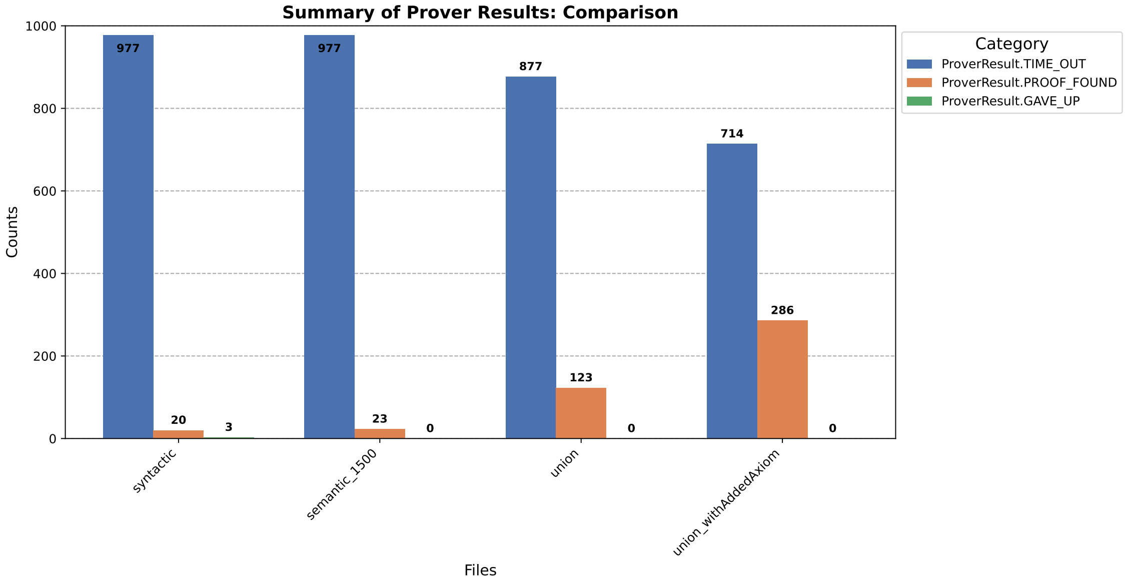

Figure 4.6: Summary of prover results with core axioms

- Cosine similarity is compared across different selection methods.

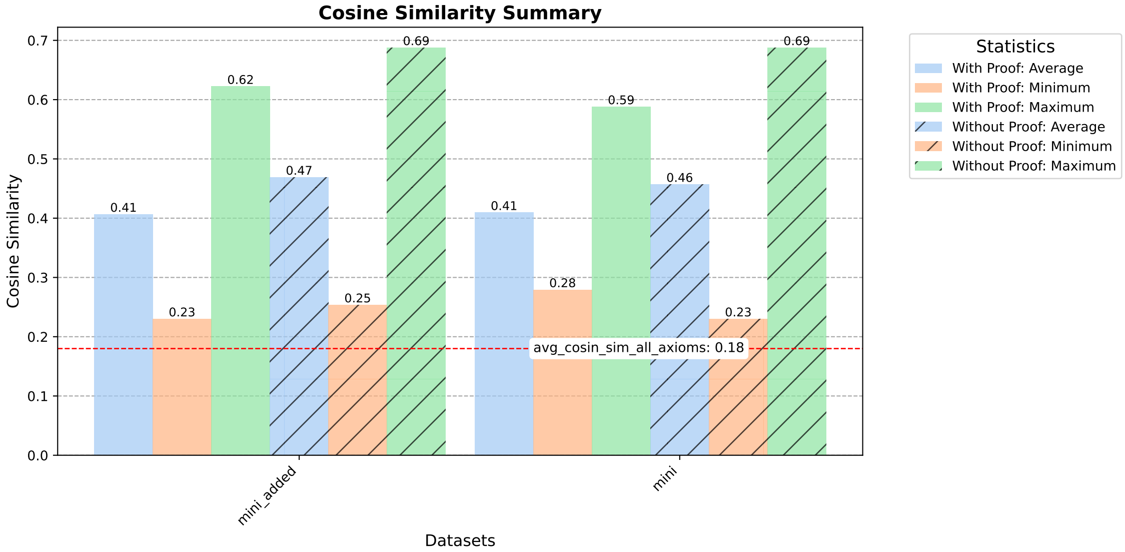

Figure 4.7: Cosine similarity comparison

4.5 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.

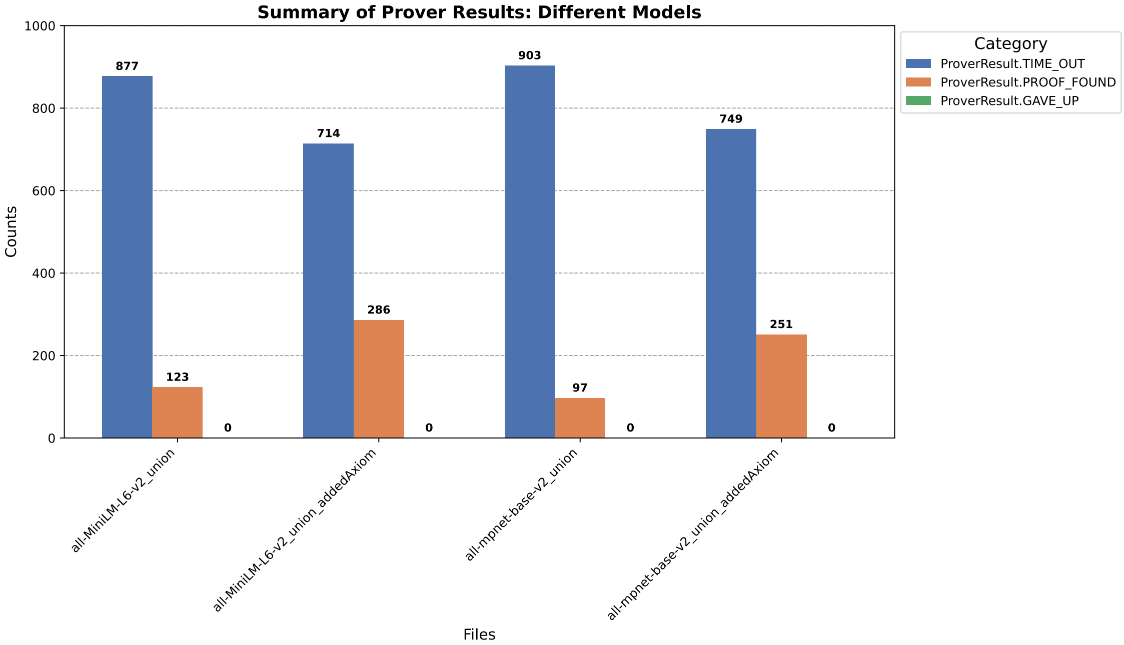

Figure 4.8: Summary of prover results with different models

- The cosine similarity of selected axioms is compared.

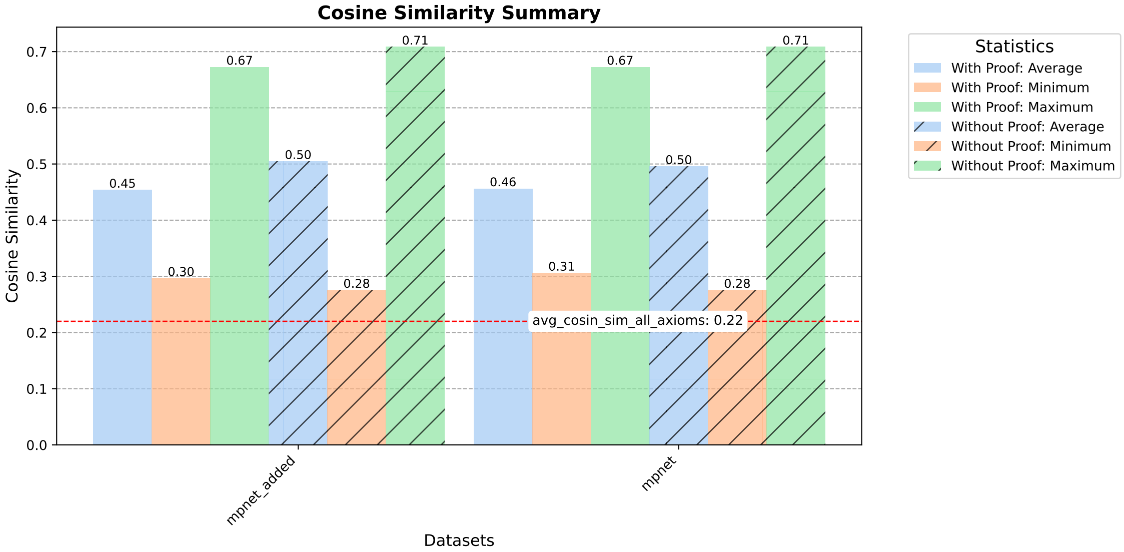

Figure 4.9: Cosine similarity in different models

4.6 Evaluation of Theorem Prover Configurations

The effect of different theorem prover configurations is analyzed.

- The performance of Satauto mode is examined.

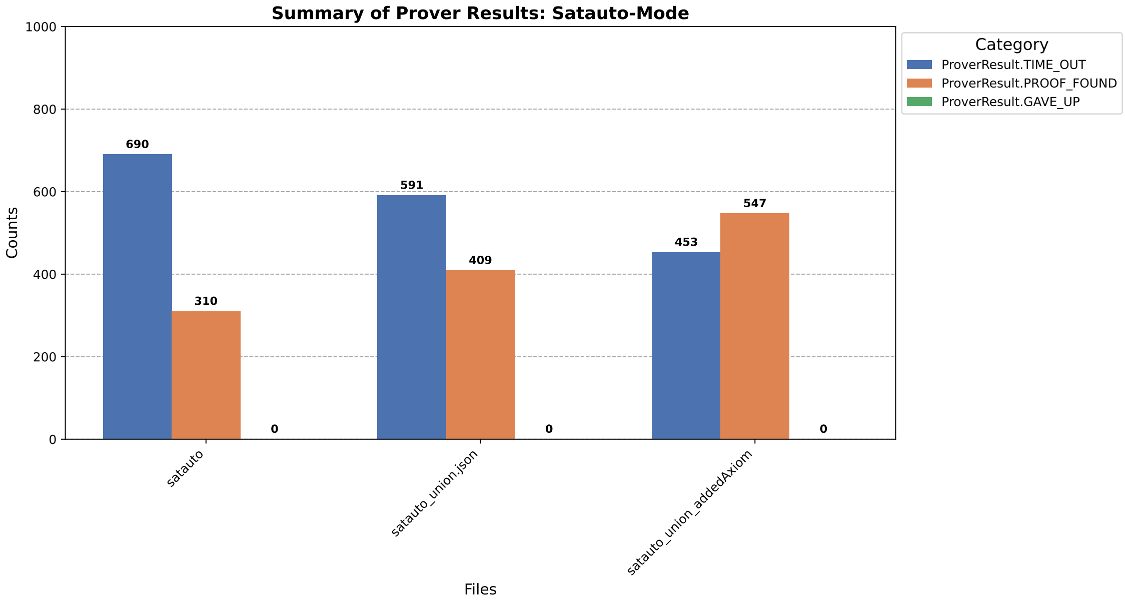

Figure 4.10: Summary of prover results in Satauto mode

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

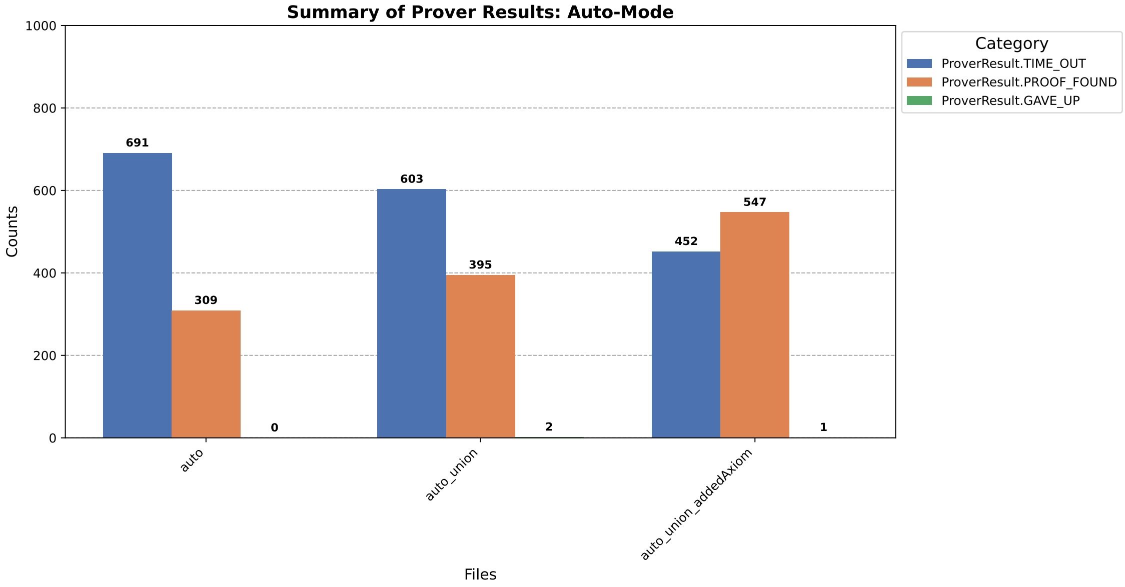

Figure 4.11: Summary of prover results in Auto mode

5

Conclusion

Conclusion.

6

Future work

Future work.