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

0.1 Introduction

In this thesis, we investigate how adding frequently used axioms ("core axioms") to a subset of the Adimen SUMO grammar, selected based on a combination of syntactic and semantic criteria, enhances the success rate of automated theorem proving.

Problem Statement

- Given a formal logical grammar, such as Adimen SUMO, and a conjecture C, we aim to determine a subset of axioms that maximizes the probability of proving C.
- We use a combination of syntactic and semantic similarity measures to select an initial set of relevant axioms $\mathcal{A}_{\text{rel}}$.
- We hypothesize that augmenting this set with frequently used axioms $\mathcal{A}_{\text{core}}$ further improves proof success.

Entanglement Between Natural Language and Logical Grammar

- We do not assume a direct mapping $f : \mathcal{L}_{\text{NL}} \to \mathcal{L}_{\text{logic}}$ between natural language ($\mathcal{L}_{\text{NL}}$) and formal logic ($\mathcal{L}_{\text{logic}}$).
- Instead, we propose an **entanglement hypothesis**, where natural language and logical structures influence each other:

$$\mathcal{L}_{\text{NL}} \bowtie \mathcal{L}_{\text{logic}} \tag{1}$$

- Adimen SUMO, partially derived from WordNet, reflects structured knowledge from natural language.
- Natural language inherently follows logical structures, even if it remains ambiguous and context-dependent.
- Language models like SBERT capture semantic relationships, supporting the idea that formal logical expressions share deep structural similarities with natural language.

Necessity of Adding Core Axioms

- To find a proof, semantic similarity alone is insufficient because logical reasoning relies on structural dependencies and inferential steps that may not be directly encoded in semantic embeddings.
- Theorem provers require axioms that establish intermediate logical connections because of the nature of theorem proving:
- Operation of Theorem Prover E To better understand the importance of intermediate axioms, consider how a theorem prover like E operates:
  1. **Input and Preprocessing**:
     - Parsing of input axioms and conjectures into a logical syntax.
     - Conversion of formulas into conjunctive normal form (CNF) for uniformity.
  2. **Indexing**:

    - Clause indexing for efficient retrieval.
    - Use of structures like discrimination trees for quick unification and resolution.
  3. **Main Proof Search Loop**:
    - Selection of clauses using strategies based on heuristics.
    - Application of inference rules such as resolution and paramodulation.
    - Simplification and subsumption of inferred clauses.
  4. **Proof State Management**:
    - Balancing expansion and reduction of search space.
    - Employing search heuristics to efficiently explore the proof space.
  5. **Proof Search Termination**:
    - Success: Derivation of a contradiction clause indicating a proof.
    - Failure: Resource exhaustion without finding a proof.
  6. **Output Generation**:
    - Generation of a proof object showing applied inference rules.
    - Provision of a proof trace and performance statistics.

  - A proof is not simply a similarity-based retrieval task but requires chaining multiple inference steps to reach the desired conclusion.
  - Many proofs require bridging axioms, which provide necessary intermediate logical steps between known premises and the conjecture.
  - Without these bridging axioms, a theorem prover may fail to construct a proof, even if it has access to semantically relevant axioms.

- The absence of intermediate axioms results in gaps in the inference chain, preventing the prover from reaching the conclusion.
- Empirical evidence from theorem proving suggests that missing such axioms often leads to failed proof attempts.
- Core axioms provide frequently used logical rules and bridges that help complete reasoning chains.
- Without these core axioms, proofs may fail due to missing key inference steps, even if semantically similar axioms are present.

Formal Hypothesis

- Define:
  - $\mathcal{A}$: 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:

$$\mathcal{A}_{\text{rel}} = \{A \in \mathcal{A} \mid A \text{ is among the } k \text{ closest axioms to } C \text{ w.r.t. } d\} \tag{2}$$

  - $\mathcal{A}_{\text{core}}$: 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:

$$\forall C \in \mathcal{C}, \quad \Pr(P(T, \mathcal{A}_{\text{enh}}, C) = \text{True}) > \Pr(P(T, \mathcal{A}_{\text{rel}}, C) = \text{True}) \tag{3}$$

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

Justification and Implications

- Since Adimen SUMO's grammar is **entangled** with natural language structures, selecting axioms using semantic embeddings (e.g., SBERT) captures implicit logical dependencies.
- Frequently used axioms act as **structural anchors** within this entanglement, improving proof discovery.
- The theorem prover benefits from both the **nearest axioms** and the **core axioms**, leading to an overall higher proof success rate.

This thesis aims to empirically validate this hypothesis by evaluating proof success rates with different axiom selection strategies and demonstrating the effectiveness of incorporating core axioms.

0.2 Related Work

(**Schon2024**) (**Àlvez2014**) (**Hoder2011**) (**Roederer2009**) (**Sutcliffe2007**)

0.3 Structure of this Thesis

Structure of this Thesis.

1

Preliminaries

1.1 Common Sense Reasoning

- **Explanation:** Common sense reasoning refers to the ability of a system to make presumptions about the everyday world, similar to human reasoning.
- **What is it?** It involves reasoning about typical situations, handling incomplete information, and making reasonable assumptions.
- **Applications and Use Cases:**
  - Natural language understanding.
  - Automated reasoning.
  - Robotics and AI decision-making.
- **Current State:** Modern AI systems, including deep learning models, struggle with common sense reasoning, but efforts such as knowledge graphs and large-scale pre-trained models aim to improve it.
- **Challenges:**
  - Ambiguity in natural language.
  - Lack of structured common-sense knowledge.
  - Computational limitations in reasoning.

1.2 Word Embeddings

- **Definition:** Word embeddings are dense vector representations of words that capture semantic meanings.
- **Creation:** Generated using neural network-based models like Word2Vec, GloVe, and FastText.
- **Properties:**
  - Captures semantic similarity.
  - Enables algebraic operations like $king - man + woman = queen$.
- **Applications:**
  - Natural language processing (NLP).

  - Machine translation.
  - Information retrieval.
- **Limitations and Challenges:**
  - Contextual ambiguity.
  - Bias in training data.
  - Scalability for large vocabularies.

1.3 Grammars

- **Explanation:** Grammars define the structure and rules governing a language, whether natural or formal.
- **Types of Grammars:** Context-free grammars, dependency grammars, formal grammars.
- **Structure and Organization:** Grammars consist of syntax rules, lexicons, and transformations.
- **Why Are They Needed?** They provide the foundation for parsing, generating, and interpreting language in NLP and logic systems.
- **SUMO:** Suggested Upper Merged Ontology, a formal ontology integrating linguistic and logical structures.

1.4 Theorem Provers

- **Explanation:** Theorem provers are automated reasoning tools designed to validate logical statements based on formal rules.
- **Types of Theorem Provers:** Resolution-based, tableau-based, and interactive provers.
- **Functionality:** They work by refutation—attempting to derive contradictions from the negation of a theorem.
- **Why Are They Needed?** Essential for verifying mathematical proofs, formal verification, and logic-based AI.
- **Limits and Challenges:**
  - Computational complexity.
  - Handling undecidable problems.
  - Scalability for large proof spaces.
- **Prover E:**
  - **How Does E Work?** A saturation-based theorem prover using superposition calculus.
  - **Satauto Mode:** An automatic mode that selects strategies dynamically for proof search.

1.5 Selection Strategies

- **Syntactic:** Selects axioms based on structural similarity and pattern matching.
- **Semantic:** Uses embeddings and meaning-based similarity to find relevant axioms.
- **Combination - Union:** Integrates syntactic and semantic methods to enhance selection efficiency.

2

Experiments

2.1 How WhiteboxTests are made

- Paper of Álvez about building WhiteBox Tests (**Álvez2017**)
  1. In the context of Adimen SUMO, **falsity tests** are formulated by taking specific statements or conditions and applying **negation** to them.
  2. These negated statements serve as a basis for falsity tests, which are essential in logical proof systems.
  3. A **theorem prover** such as Prover E then performs a process known as **refutation**.
  4. Through refutation, the theorem prover attempts to demonstrate the **inconsistency** or **falsehood** of the original statements.
  5. This is achieved by attempting to derive a **contradiction** from the negated statements, indicating that the original (non-negated) statements cannot all be true simultaneously.

2.2 Reengineering of paper (**Schon2024**)

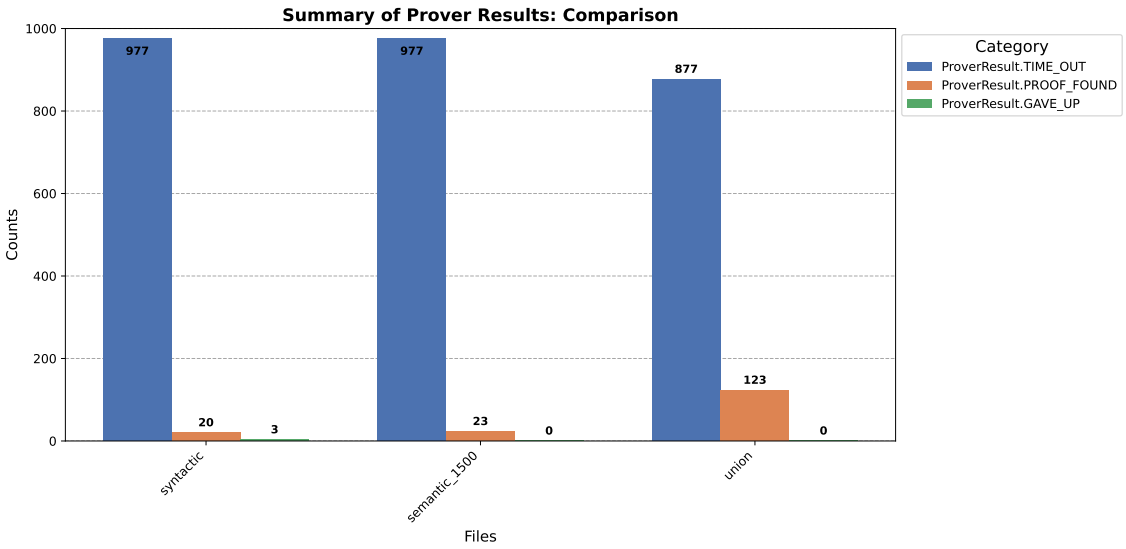

Figure 2.1: Reengineering of results of paper (**Schon2024**)

2.3 Analyze of reeingineering

– Mean variable count

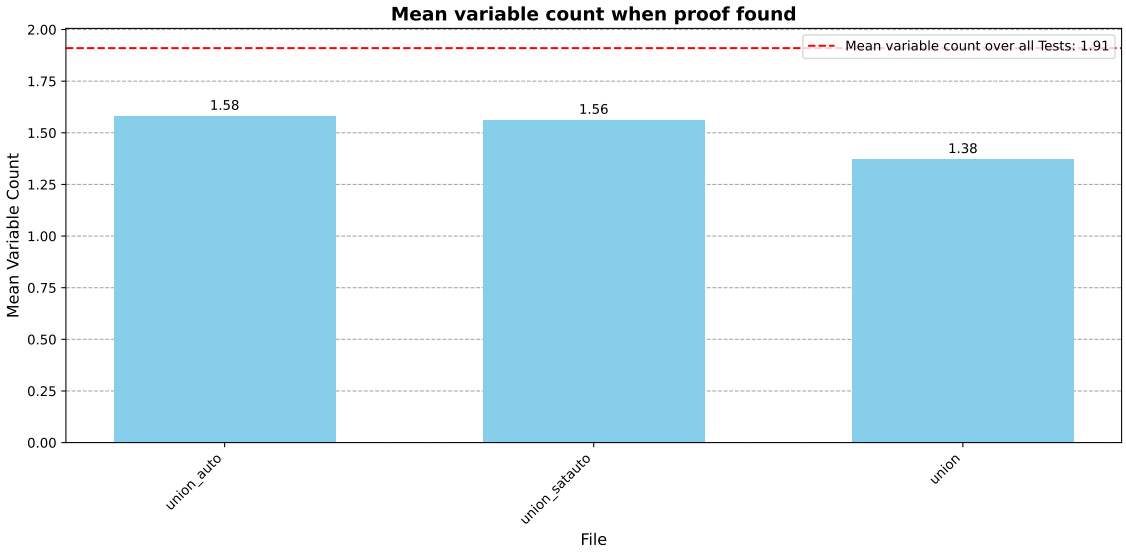

Figure 2.2: Variable Count

– Count signs

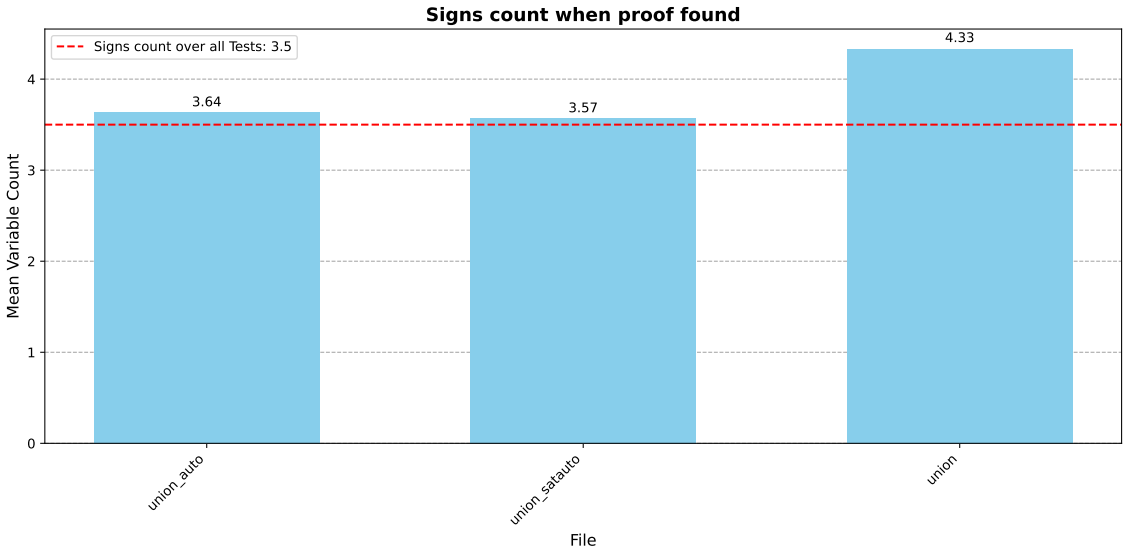

Figure 2.3: Count signs

– Character Count

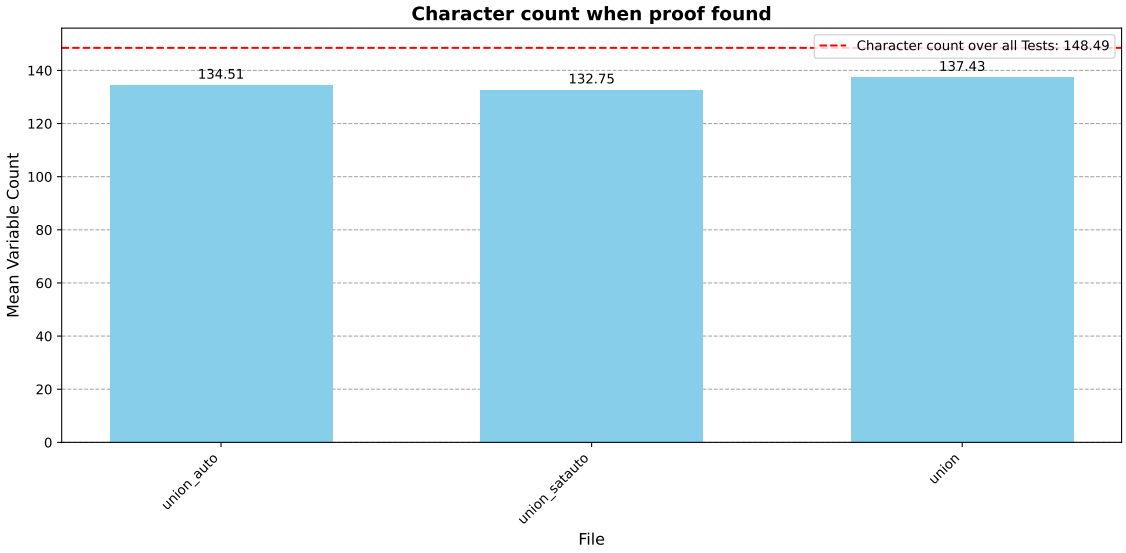

Figure 2.4: Character count

– Cosine similarity Why Would Provable Conjectures Have Lower Axiom Similarity?

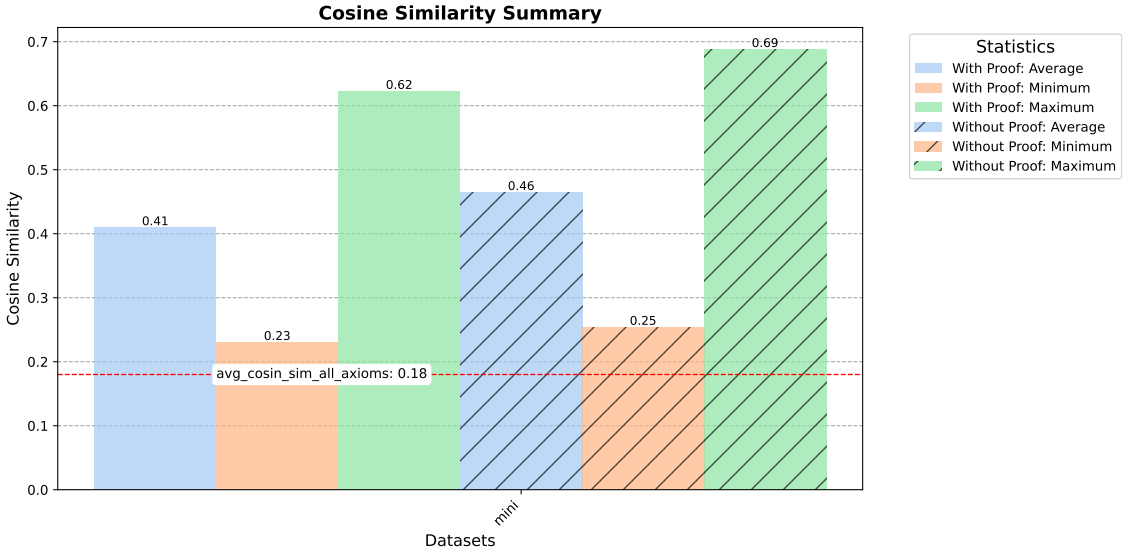

Figure 2.5: Cosine similarity

This suggests that proving the conjecture involves bridging different concepts, rather than just refining a single closely related idea.

2.4 Add core axioms

– Results of prover

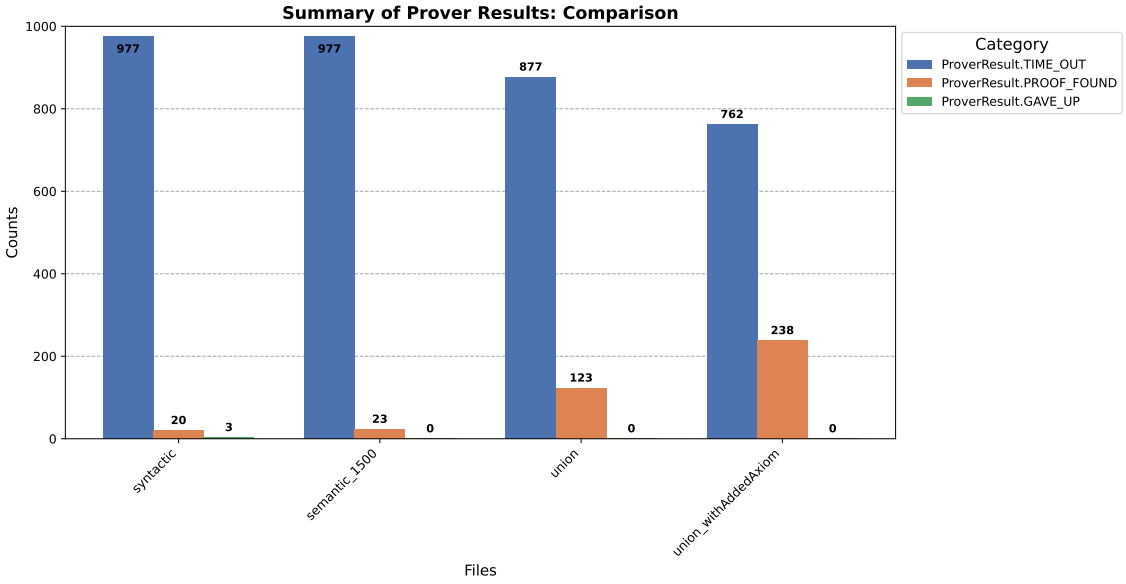

Figure 2.6: Summary of Prover Results with core axioms

– Cosine similarity

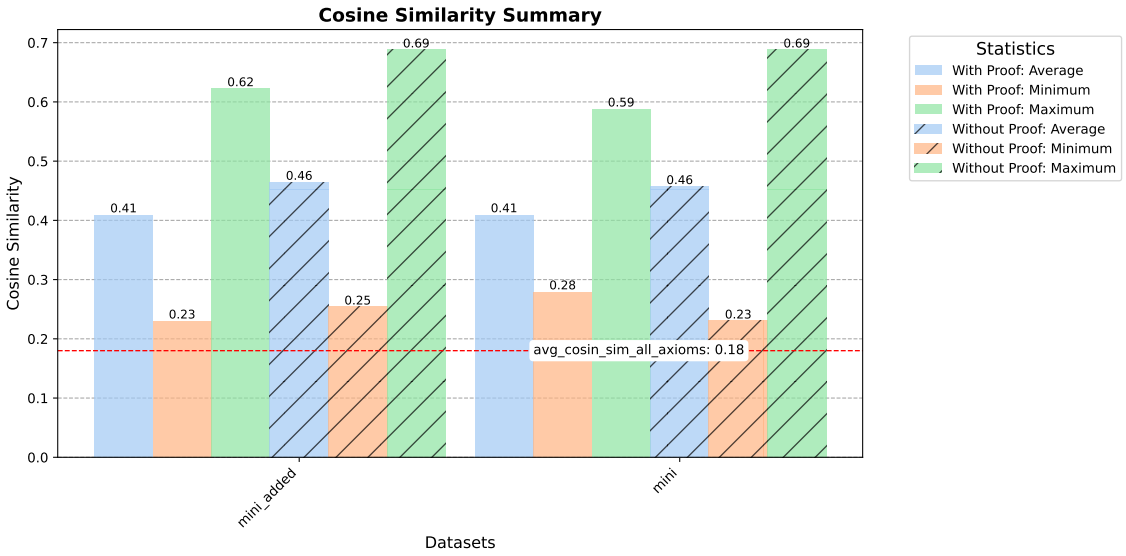

Figure 2.7: Summary of Prover Results: Standard Mode

2.5 Comparing LLMs

– Results of prover

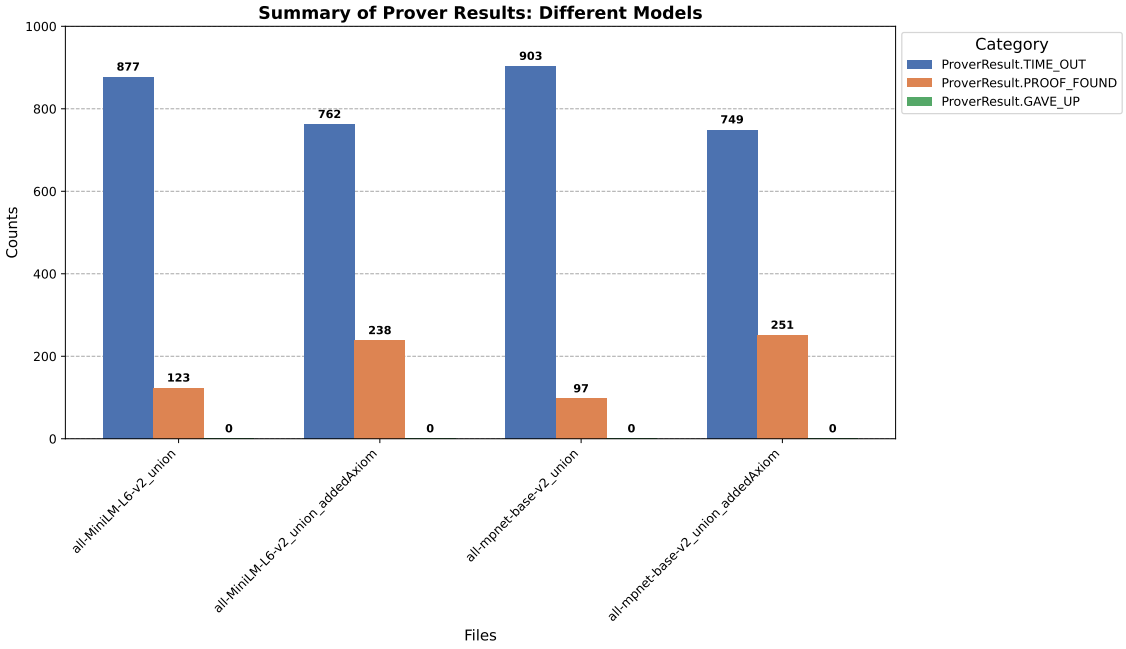

Figure 2.8: Summary of Prover Results: Different models

– Cosine similarity

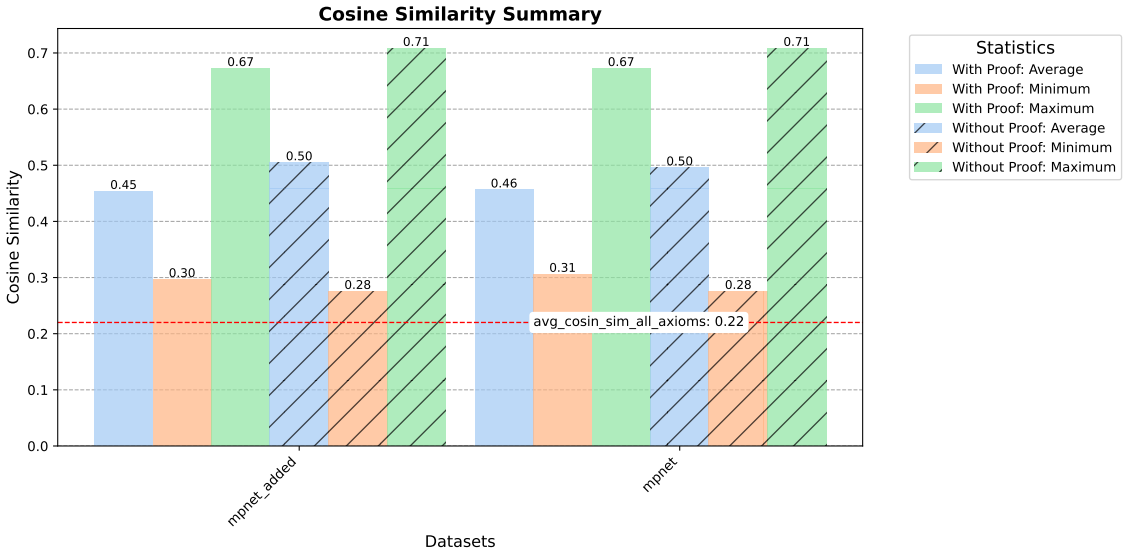

Figure 2.9: Cosine Similarity Mpnet

2.6 Analyze different E settings

– Satauto Using search heuristic

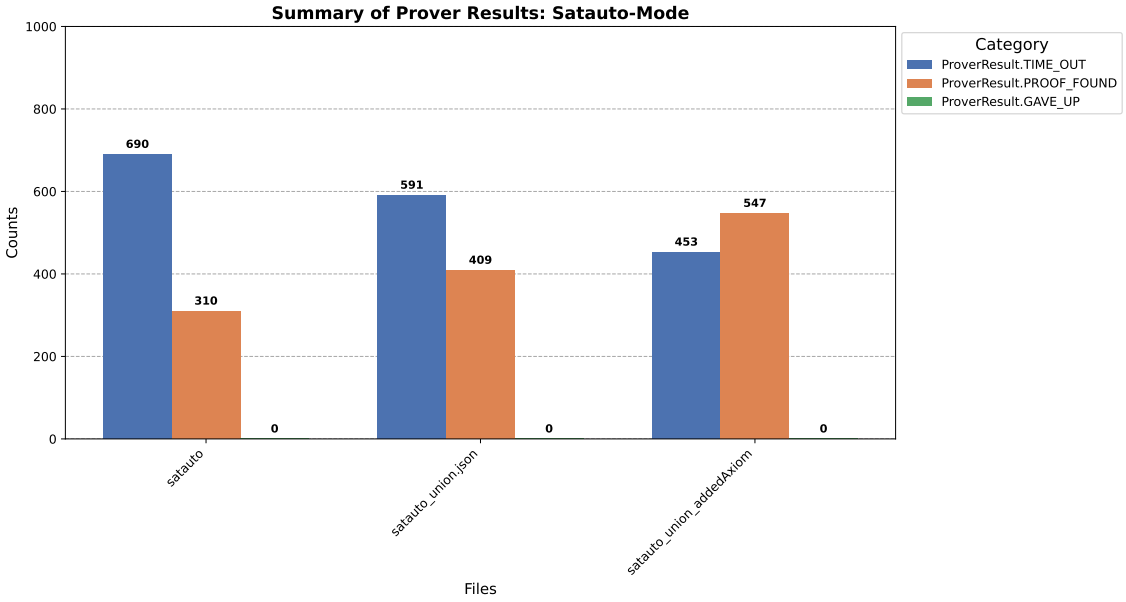

Figure 2.10: Summary of Prover Results: Satauto Mode

– Auto Using SInE + search heuristic

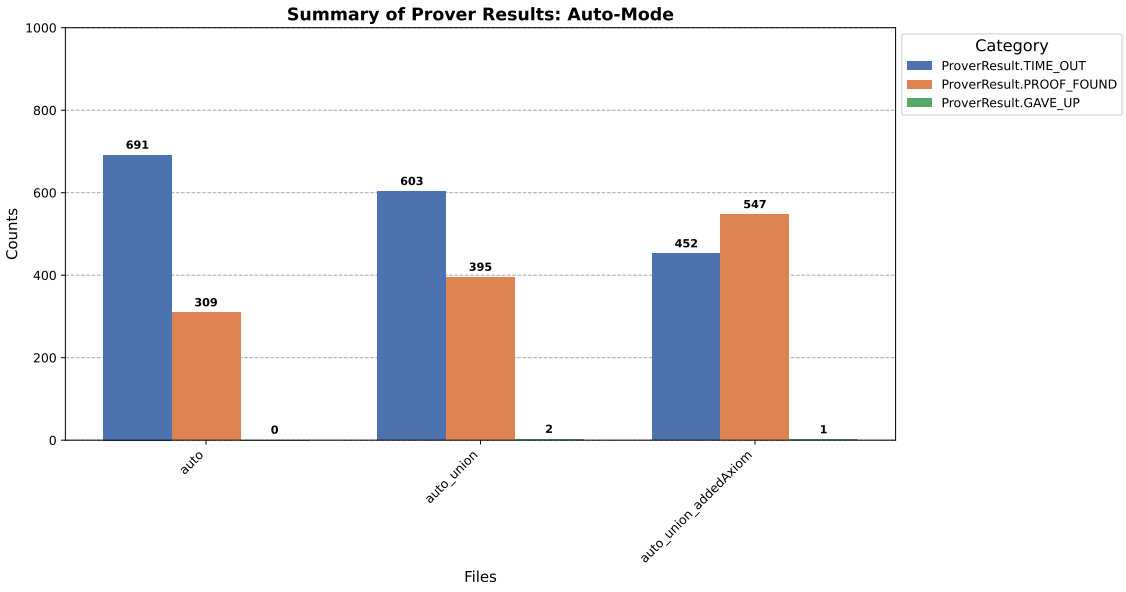

Figure 2.11: Summary of Prover Results: Auto Mode

2.7 Different prover

– Vampire

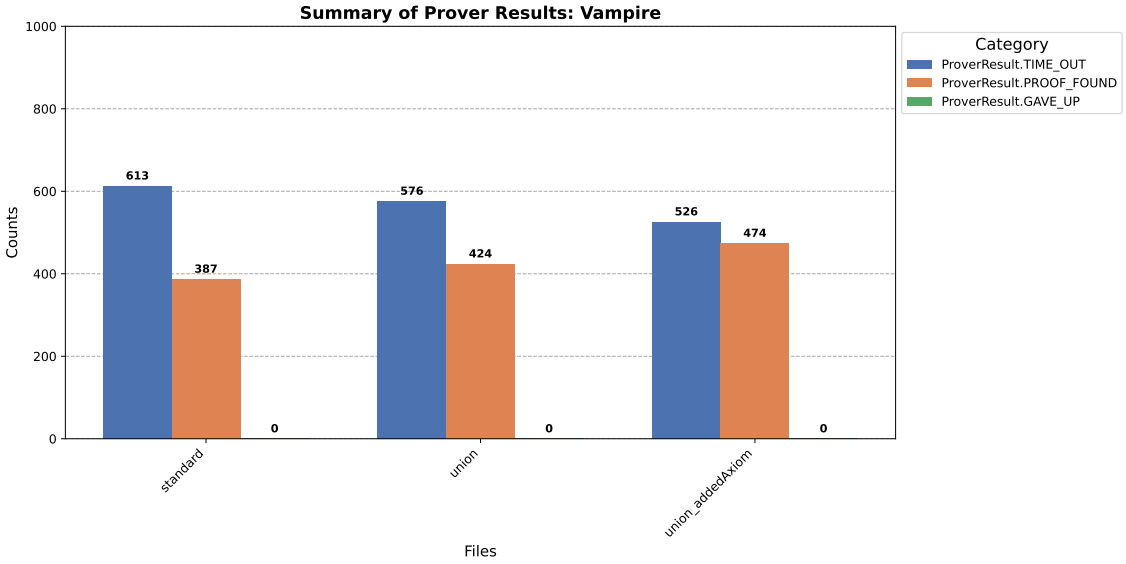

Figure 2.12: Summary of Prover Results: Vampire Mode

2.8 Time study

– Proofs found in first named

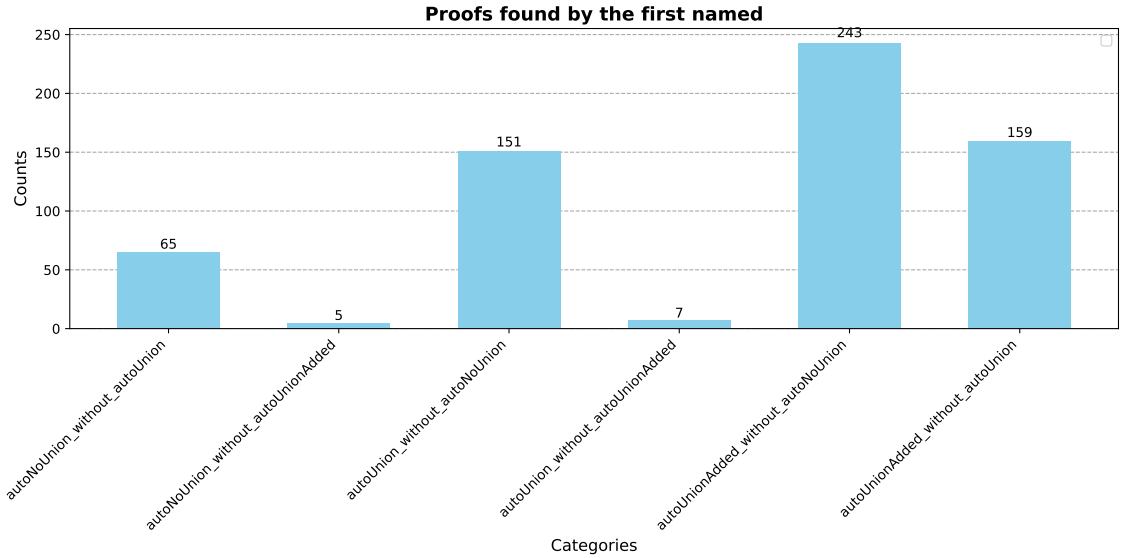

Figure 2.13: Proof found on other modes

– Time to find proof

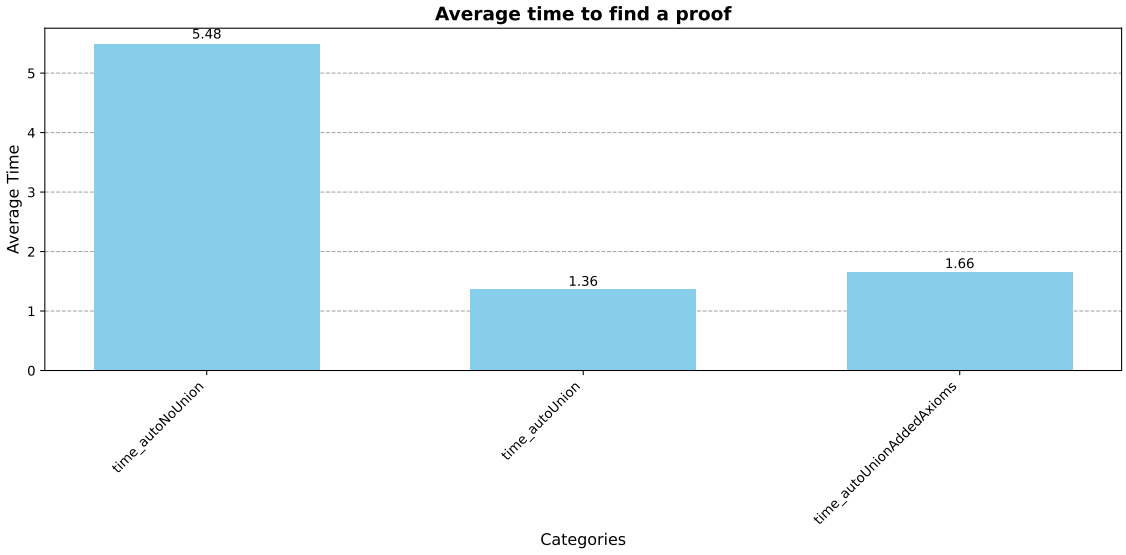

Figure 2.14: Proofs found time

– Time to find common proof

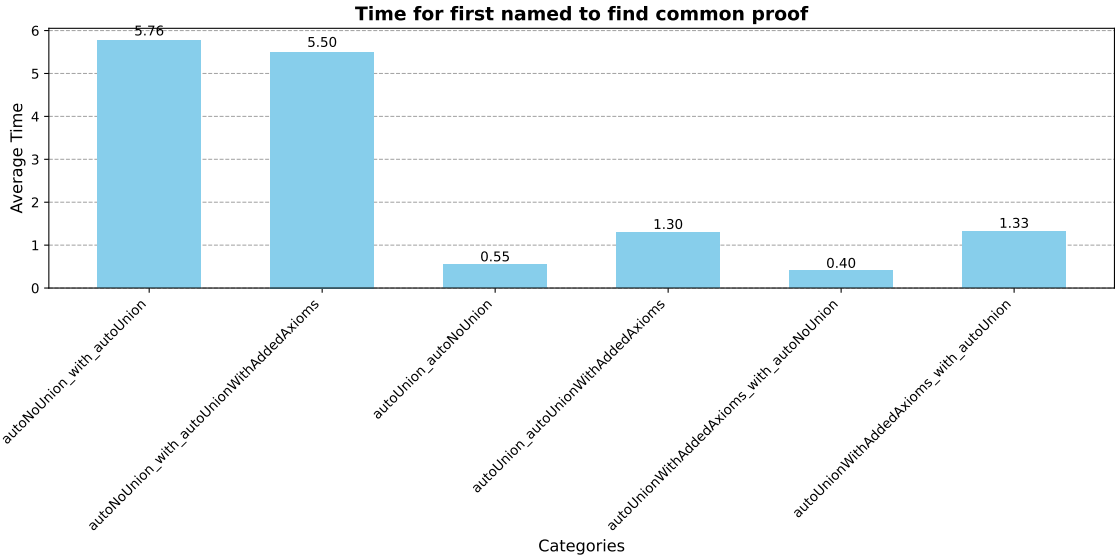

Figure 2.15: Common proofs found time

3

Conclusion

Conclusion.

4

Further work

Further work.

$$