Enhancing Argumentation Structure Parsing with Deep Probabilistic Answer Set Programming

Jonas Rodrigues Lima Gonçalves
Undergraduate

Denis Deratani Mauá
Advisor

April 15, 2025

1 Introduction

Argumentation Mining (AM) is a significant area within Natural Language Processing (NLP) focused on extracting argumentative structures from text. Stab and Gurevych have contributed significantly to this field by proposing a pipeline approach for parsing argumentation structures in persuasive essays. Their method involves segmenting the extraction process into several sub-steps, including identifying argument components, classifying their types, and identifying argumentative relations. A crucial final step in their pipeline utilizes Integer Linear Programming (ILP) to enforce coherence between the outputs of these sub-steps and generate a globally consistent argumentation graph.

While the ILP approach offers a way to integrate local predictions into a global structure, it can face limitations in terms of capturing complex probabilistic dependencies and being seamlessly integrated with end-to-end differentiable learning. Recent advancements in neural-symbolic reasoning, particularly in Deep Probabilistic Logic Programming frameworks like dPASP, offer promising alternatives. dPASP extends Answer Set Programming (ASP) with neural predicates, probabilistic choices, and tight integration with deep learning frameworks like PyTorch, enabling end-to-end training through automatic differentiation.

This thesis project aims to **investigate the potential of replacing the integer programming-based coherence enforcement step in Stab and Gurevych's argumentation parsing pipeline with dPASP**. By leveraging dPASP's capabilities for probabilistic modeling and integration with neural networks, this project seeks to develop a more flexible and potentially more accurate approach to ensuring the coherence of extracted argumentation structures.

2 Objectives and Goals

The primary objective of this thesis project is to **design, implement, and evaluate a dPASP-based module for enforcing coherence in an argumentation structure parsing pipeline, drawing inspiration from the work of Stab and Gurevych**.

The specific goals of this project include:

- **Understanding the Existing Pipeline:** Thoroughly analyze the argumentation structure parsing pipeline proposed by Stab and Gurevych, with a specific focus on the role and implementation of the integer linear programming step. This includes understanding the input to this step (outputs of the component identification, classification, and relation identification models) and its output (the coherent argumentation graph).
- **Conceptualizing a dPASP-based Coherence Model:** Develop a probabilistic logic program in dPASP that can take the probabilistic outputs of the preceding pipeline stages (or their representations) as input and reason about the coherent structure of the argument. This will involve defining neural predicates that can incorporate the uncertainty from the earlier stages.
- **Integrating dPASP with the Pipeline:** Implement the integration of the dPASP module into the existing argumentation parsing pipeline, potentially using the publicly available resources from Stab and Gurevych's work. This will involve handling the interface between the machine learning classifiers and the dPASP framework.
- **Training the dPASP Model (if applicable):** Explore the possibility of end-to-end or joint training of the dPASP module, potentially leveraging dPASP's integration with PyTorch for automatic differentiation. This might involve defining appropriate loss functions, which can be specified via dPASP constraints, that consider the desired properties of coherent argumentation structures.
- **Evaluating the dPASP-enhanced Pipeline:** Evaluate the performance of the modified argumentation parsing pipeline, comparing it to the original approach that uses integer programming. This evaluation will likely involve using the argumentation structure annotated corpus created by Stab and Gurevych and standard evaluation metrics for argumentation mining, such as F1-score for component identification, classification, and relation identification, as well as measures of overall graph coherence.
- **Analyzing the Results:** Analyze the strengths and weaknesses of the dPASP-based approach, identifying potential improvements and discussing the implications for the field of argumentation mining and neural-symbolic reasoning.

3 Planning Major Activities

The project will be conducted through the following major activities:

- **Phase 1: Literature Review and Background Study (Month 1):**
  - In-depth study of Stab and Gurevych's work on argumentation structure parsing.
  - Comprehensive review of the dPASP framework, its syntax, semantics, and capabilities.
  - Exploration of related work in neural-symbolic reasoning for NLP tasks, including DeepProbLog, NeurASP, and other approaches to combining deep learning and logical inference.
  - Familiarization with the argumentation structure annotated corpus from Stab and Gurevych.
- **Phase 2: Conceptualization and Design (Month 2):**
  - Detailed design of the dPASP-based model for enforcing coherence in argumentation structures.
  - Defining how the probabilistic outputs (or relevant information) from the earlier stages of Stab and Gurevych's pipeline can be represented and utilized within dPASP, possibly through neural predicates.
  - Developing the logical rules and probabilistic choices within the dPASP program to capture the desired properties of coherent argumentation graphs (e.g., hierarchical structure, support and attack relations, consistency of component types and relations).
  - Planning the integration strategy with the existing pipeline.
- **Phase 3: Implementation and Integration (Months 3-4):**
  - Implementation of the dPASP-based coherence module using the dPASP framework and potentially PyTorch.
  - Development of the necessary interface to connect this module with the preceding stages of Stab and Gurevych's pipeline (which might involve pre-trained machine learning models).
  - Handling data input and output formats for the dPASP module.
- **Phase 4: Training and Optimization (Month 5):**
  - If end-to-end or joint training is feasible and deemed beneficial, define the training data, loss functions (PASP rules), and optimization procedures. This might involve adapting the training data and labels from Stab and Gurevych's corpus or exploring new forms of supervision.
  - Tune the parameters of the dPASP model and the training process.

- **Phase 5: Evaluation (Month 6):**
  - Evaluate the performance of the integrated argumentation parsing pipeline using the Stab and Gurevych corpus.
  - Compare the results with the performance reported by Stab and Gurevych for their original pipeline, focusing on the impact of replacing the ILP step with dPASP.
  - Analyze the quantitative results based on standard argumentation mining evaluation metrics.
- **Phase 6: Analysis and Thesis Writing (Month 7):**
  - In-depth analysis of the results, including the strengths, weaknesses, and potential of the dPASP-based approach.
  - Discussion of the findings in the context of related work and the broader field of neural-symbolic reasoning and argumentation mining.
  - Writing and finalizing the undergraduate thesis document, including an introduction, literature review, methodology, results, discussion, and conclusion.

4 Methodology

This project will employ a neural-symbolic approach, leveraging the strengths of both deep learning and logical reasoning. The core methodology will involve:

- **Replication and Adaptation:** Starting with the well-defined pipeline of Stab and Gurevych, the project will focus on replicating the overall structure while specifically replacing the ILP-based coherence enforcement with dPASP.
- **Probabilistic Logic Programming with Neural Predicates:** The coherence of the extracted argument components and relations will be modeled using dPASP, a framework that allows for defining probabilistic logic programs with neural predicates. These neural predicates can be designed to incorporate the confidence scores or probability distributions output by the preceding machine learning models in the pipeline.
- **Answer Set Programming (ASP):** dPASP builds upon ASP, which provides a declarative language for knowledge representation and reasoning. ASP rules will be used to define constraints and preferences for a coherent argumentation structure, such as the expected relationships between claims, premises, and potentially major claims.
- **Integration with Deep Learning (PyTorch):** dPASP's tight integration with PyTorch will be crucial for potentially training the coherence model end-to-end or for leveraging pre-trained neural networks from the

earlier stages of the pipeline. Automatic differentiation capabilities will be explored for learning the parameters of the dPASP model, if applicable.

- **Experimental Evaluation:** The performance of the dPASP-enhanced pipeline will be rigorously evaluated on the annotated corpus of persuasive essays created by Stab and Gurevych, using standard evaluation metrics for argumentation mining. Comparison with the original ILP-based approach will be a key aspect of the evaluation.

5 Expected Outcomes and Contributions

This thesis project is expected to yield the following outcomes and contributions:

- **A dPASP-based module for enforcing coherence in argumentation structure parsing.** This module will demonstrate the application of deep probabilistic logic programming to a challenging NLP task.
- **An evaluation of the effectiveness of dPASP compared to integer programming for ensuring coherence in argumentation parsing.** This comparison will provide insights into the strengths and weaknesses of neural-symbolic approaches for this task.
- **Potential improvements in the accuracy and robustness of argumentation structure parsing.** By leveraging the probabilistic reasoning and learning capabilities of dPASP, the project may lead to a more accurate and flexible approach to coherence enforcement.
- **A deeper understanding of the integration of deep learning and logical reasoning for argumentation mining.** The project will contribute to the growing body of work on neural-symbolic computing in NLP.
- **A documented implementation and evaluation of the proposed approach**, which could serve as a foundation for future research in this area.

6 Timeline

Time Period	**Major Activities**
Month 1	Literature Review and Background Study
Month 2	Conceptualization and Design of the dPASP Model
Months 3-4	Implementation and Integration of the dPASP Module
Month 5	Training and Optimization
Month 6	Evaluation of the Integrated Pipeline
Month 7	Analysis and Thesis Writing

7 Required Resources

- **Computational Resources:** Access to adequate computational resources, including GPUs, for training and evaluating deep learning models and running dPASP programs.
- **Software Resources:** Python programming environment, PyTorch for deep learning, and dPASP framework. Access to additional libraries for NLP tasks like tokenization, parsing, and evaluation.
- **Data Resources:** The annotated corpus of persuasive essays created by Stab and Gurevych, potentially other argumentation mining datasets for additional evaluation or transfer learning experiments.
- **Academic Resources:** Access to relevant scientific literature, including papers on argumentation mining, neural-symbolic reasoning, and probabilistic logic programming.

8 Risks and Mitigation Strategies

- **Risk:** The integration of dPASP with the existing pipeline components may face technical challenges or incompatibilities. **Mitigation:** Start with a simple, well-defined interface between components and gradually increase complexity. Develop a modular approach that allows for easy testing and debugging of different parts of the system.
- **Risk:** The dPASP-based approach may not outperform the original ILP approach in terms of accuracy or efficiency. **Mitigation:** Design the experiments to focus not just on overall performance metrics but also on specific aspects where dPASP might offer advantages (e.g., handling uncertainty, integration with learning, flexibility in defining constraints). This will ensure valuable insights even if the approach does not outperform ILP on all metrics.
- **Risk:** The computational resources required for training and evaluation may exceed available capacity. **Mitigation:** Start with smaller-scale experiments and optimized implementations, gradually scaling up as needed. Consider using pre-trained models or transfer learning to reduce the computational burden.
- **Risk:** The project timeline may be affected by unforeseen challenges or delays. **Mitigation:** Build flexibility into the timeline and have contingency plans for prioritizing essential components if time becomes limited. Regular monitoring of progress against the timeline will help identify potential delays early.

9 Conclusion

This undergraduate thesis project aims to investigate the potential of deep probabilistic logic programming, specifically dPASP, for enhancing argumentation structure parsing. By replacing the integer linear programming step in Stab and Gurevych's pipeline with a dPASP-based approach, the project seeks to leverage the strengths of neural-symbolic reasoning for ensuring coherence in extracted argumentation structures. The expected outcomes include a novel implementation, empirical evaluation, and insights into the integration of deep learning and logical reasoning for this challenging NLP task. The results of this project could potentially lead to more accurate and flexible approaches to argumentation mining, with implications for applications in areas such as automated essay scoring, legal text analysis, and scientific discourse understanding.