Problem & Motivation

Our project helps students understand dense philosophical texts that contain many intertwined claims, premises, objections, and replies. Such arguments are often written as long paragraphs, so readers must keep many elements in their working memory and can easily lose track of the reasoning. Research on argument mapping shows that representing arguments as visual maps of claims and their inferential relations can support students in analyzing and evaluating complex text-based arguments, making the underlying logical structure clearer and easier to follow (Dwyer et al., 2011). We aim to build a system that takes a philosophical text and turns its main argumentative structure into an interactive graph, so readers can quickly see the central claims and how they connect.

System Objectives:

1. Automatically extract an argument structure from philosophical or other argumentative texts and store it in a graph representation with nodes for central claims, supporting premises, objections and replies.
2. Build a web application where users can paste text or upload a PDF, trigger the extraction process from objective 1, and see the resulting argument graph.
3. Make the argument graph interactive, so users can click nodes to see the original sentences and a short paraphrase and ask natural language questions about specific nodes or subgraphs that are answered by the LLM powering the system.

Existing Technology and Theoretical Review:

Existing argument mapping tools such as Rationale and Argunet and platforms like Kialo demonstrate how visual argument graphs and pro or con trees can support structured discussion and critical thinking, but they require users to manually construct the map and therefore already understand the underlying argument structure. Large language models can summarize texts and list main points, yet they typically output linear summaries rather than an explicit argument graph. Our system combines these directions by using LLM based argument mining to propose candidate claims and simple support or attack relations, then rendering these suggestions as an interactive argument graph that users can inspect and query.

The design is informed by Cognitive Load Theory, which highlights limits of working memory and the benefits of externalizing structure in diagrams (Sweller, 1988), by Dual Coding Theory on combining verbal and visual representations to enhance comprehension (Clark & Paivio, 1991), and by guidelines for human AI interaction that emphasize transparency and user control (Amershi et al., 2019). In our system, transparency and control are supported by showing, for each node, the original text span and an LLM generated paraphrase and by allowing users to choose which nodes or subgraphs to focus on and what natural language questions to ask about them, instead of receiving a single opaque summary.

Approach

The graph construction system will consist of three main components. The first component is an argument-mining backend using classical NLP pipelines, that identify argumentative components and predict relations between them (Lawrence & Reed, 2019). It takes the input text, segments it into paragraphs, and uses known cues (e.g., therefore, because, however) to mark argumentative candidate sentences. The second component is a prompt-engineered LLM that, given the relevant text segment, classifies these candidates into roles (e.g., main claim, premise, objection, non-argumentative) and infers support/attack relations between nearby components. After this step, we have a set of sentences labeled by type and linked via support/attack relations. The third component applies deterministic rules to turn this into an argument graph with a consistent layout akin to existing research (Lenz & Bergmann, 2024). Once the graph is constructed, it is displayed for the user as an interactive argument map. Clicking on a node reveals the original sentence and a short LLM-generated paraphrase. This will help users to parse complex arguments, in line with research on argument visualization that shows graphing arguments improves reasoning performance compared to linear text (Ngajie et al., 2025).

Plan

- **Backend Review** – 18/12, an overview of relevant techniques for the backend.
- **Initial UI Design** – 23/12, a mockup of the UI for the app, with sample data.
- **Backend Structure Design** – 23/12, a review of potential backend solutions.
- **Backend Intelligence** – 11/1, a working implementation of the backed.
- **UI Refinement** – 23/1, a final UI based on the backend & user feedback.
- **Demo** – 25/1, a live demo; the project should be in its essentially final state.
- **Final Prototype** – 29/1, the final version of the project, together with evaluation.

Bibliography

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3290605.3300233;TOPIC:TOPIC:CONFERENCE-COLLECTIONS

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review*, *3*(3), 149–210. https://doi.org/10.1007/BF01320076/METRICS

Dwyer, C., Hogan, M. J., & Stewart, I. (2011). The Promotion of Critical Thinking Skills Through Argument Mapping. *Critical Thinking*. https://www.researchgate.net/publication/259286333_The_promotion_of_critical_thinking_skills_through_argument_mapping

Lawrence, J., & Reed, C. (2019). Argument Mining: A Survey. *Computational Linguistics*, *45*(4), 765–818. https://doi.org/10.1162/COLI_A_00364

Lenz, M., & Bergmann, R. (2024). ArgueMapper Assistant: Interactive Argument Mining Using Generative Language Models. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* , *15446 LNAI*, 189–203. https://doi.org/10.1007/978-3-031-77915-2_14

Ngajie, B. N., Su, C. Y., Li, Y., & Tiruneh, D. T. (2025). Impact of a systematically designed computer-supported argument visualization tutorial on the skill of argument analysis. *Thinking Skills and Creativity*, *56*, 101786. https://doi.org/10.1016/J.TSC.2025.101786

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, *12*(2), 257–285. https://doi.org/10.1207/S15516709COG1202_4