# DebGraph: Graph Based Debate Evaluation and Feedback

**Priya Pitre**
Virginia Tech
priyapitre@vt.edu

**Joshua Flashner**
Virginia Tech
jflashner@vt.edu

**Prayash Joshi**
Virginia Tech
prayash@vt.edu

**Evar Jones**
Virginia Tech
jevar@vt.edu

## Abstract

The evaluation of debates requires a nuanced understanding of argument structure, factual accuracy, and persuasion. While prior methods in computational debate analysis have advanced in assessing individual dimensions of argument quality, they lack a holistic framework to integrate these dimensions effectively. We propose DebGraph, a graph-based system that combines the structural representation of Knowledge Graphs (KGs) with the contextual reasoning capabilities of Large Language Models (LLMs) to deliver multidimensional debate evaluation and feedback. By leveraging heterogeneous graph structures and iterative refinement techniques, our system captures the interplay of arguments across rounds of debate, evaluating their coherence, persuasiveness, and factual integrity. Experiments on the DebateArt and BP-Competition datasets demonstrate that DebGraph outperforms state-of-the-art models in score-based evaluations and feedback generation. The system paves the way for enhanced applications in education, public discourse, and AI-assisted moderation.

## 1 Introduction

The ability to evaluate and analyze debates is increasingly important in the digital era, where arguments are generated and consumed at unprecedented scales across diverse contexts, from academic discourse to online social platforms. Debate evaluation systems, which assess argument quality, coherence, and persuasiveness, have traditionally relied on hand-crafted features or statistical models. However, advances in machine learning, particularly in large language models (LLM) and knowledge graphs (KG), offer transformative potential for developing multidimensional frameworks that integrate structural and contextual reasoning.

Recent developments in argumentation and discourse analysis emphasize the role of heterogeneous graph structures in capturing argument relationships. Studies such as those by Guan et al. (Guan et al., 2023) highlight the effectiveness of semantics-aware graph representation learning in preserving both lexical and structural relevance in argument mining tasks. Similarly, Bhatia (Bhatia, 2023) demonstrated the use of heterogeneous graphs for policy-oriented debate structuring, enabling a more nuanced understanding of argument dynamics.

LLMs have also shown remarkable capabilities in reasoning tasks, yet challenges persist in ensuring robustness and consistency when applied to real-world debate scenarios. Wachsmuth et al. (Wachsmuth et al., 2023) proposed new benchmarks for evaluating argument quality in LLM outputs, focusing on alignment with human reasoning. Although these works provide significant insights, they lack the integration of graph-based techniques to address the inherent complexity of argument structures and relations.

Efforts to bridge the gap between graph-based frameworks and LLMs have shown promise. Shang and Huang (Shang and Huang, 2024) explored the interplay between generative models and graph analytics for complex argument representation, while Khan et al. (Khan et al., 2024) investigated debate settings as a test for truthfulness and persuasiveness in LLMs. However, existing approaches tend to focus on isolated aspects of debate evaluation, such as argument quality or semantic relations, without providing a comprehensive multidimensional framework adaptable to diverse contexts.

This paper introduces a novel debate evaluation framework that synergizes KGs and LLMs to deliver holistic, multidimensional feedback across varied real-world debate scenarios. By leveraging the structural power of structured complex graph relationship and the contextual understanding of LLMs, our approach captures complex argument relationships, evaluates debates on multiple dimensions, and provides actionable feedback. Unlike existing systems, our framework extends beyond

traditional metrics, offering broader applicability to domains such as op-eds, social media debates, and structured political discussions.

### 1.1 Contributions

The primary contributions of this paper are as follows:

- We present a novel methodology that integrates KGs and LLMs for multidimensional debate evaluation across diverse contexts.
- We conducted extensive experiments that demonstrate the effectiveness of our approach in capturing the diversity, coherence, and persuasiveness of arguments in different real-world scenarios.

## 2 Related Works

Argumentation has long been a key area of focus in natural language processing (NLP) and computational linguistics, with significant strides made in understanding and evaluating debates. Feng and Hirst (2011) presented foundational work on argument mining, focusing on the identification and extraction of argument structures from text. Their research paved the way for integrating linguistic features with computational models (Feng and Hirst, 2011). Recent research has highlighted the growing interplay between graph-based models and argument mining, showcasing the advantages of using structured graph relationships for richer and more effective argument analysis.

Zhang et al. (2023) explored the potential of graph representation learning for argument mining, identifying critical issues such as cascading error propagation and how graph structures mitigate these challenges (Zhang et al., 2023). Similarly, Ruggeri et al. (2021) introduced tree-constrained graphs to capture sentence-level dependencies, emphasizing the structural nuances essential for effective argument mining (Ruggeri et al., 2021).

Knowledge graphs have emerged as an essential tool in the creation and visualization of arguments. Al Khatib et al. demonstrated how knowledge graphs enhance argument diversity and richness (Khatib et al., 2021). Another study by Plenz et al. employed Neo4J-based graphs to visualize argument structures effectively, emphasizing their utility in deliberation analysis (Plenz et al., 2024). This body of work underscores the potential of graph-based frameworks in capturing complex argument relationships, such as effect relations, concepts, and relation types.

In debate evaluation, Potash et al. (2017) developed an RNN model aimed at maximizing audience favorability, paving the way for computational approaches to debate winner prediction (Potash and Rumshisky, 2017). Ruiz-Dolz et al. (2022) proposed an argument graph methodology that constructs argument acceptability semantics using Roberta embeddings and employs MLPs for binary classification of debate outcomes (Ruiz-Dolz et al., 2022). Although effective, these approaches predate the transformative impact of LLMs.

Liang et al. (2024) introduced a novel architecture leveraging LLMs for multidimensional debate judgment in British Parliamentary debates. Their framework integrates chronological argument analysis, advancing the scope of debate outcome prediction beyond binary metrics (Liang et al., 2024). Mou et al. focuses narrowly on political debate settings using a unified framework that integrates local and global knowledge via knowledge graphs. Their approach demonstrates the potential of LLMs in complex political contexts (Mou et al., 2023).

These studies collectively highlight the intersection of graph-based methods and argumentation, forming the foundation for innovative applications in debate evaluation and feedback generation. However, current methods often lack the ability to holistically evaluate debates, particularly in diverse real-world contexts such as op-eds, social media discussions, and presidential debates. This presents an opportunity to integrate LLMs and exploit relationships within a graph structure to develop more generalizable debate evaluation systems.

## 3 Methods

We explain the methodology for the generation of the graph, scoring system and winner determination in this section. The detailed architecture for our work can be found in Figure 1.

### 3.1 Dataset Preparation

We utilized the DebateArt PanelBench dataset, which contains structured debate information including motions, individual debater speeches, and ground truth winner annotations. The dataset captures 1v1 debates across a variety of topics, providing a rich source of argumentation data. Debates were processed to extract motion-specific arguments, categorizing them into components such

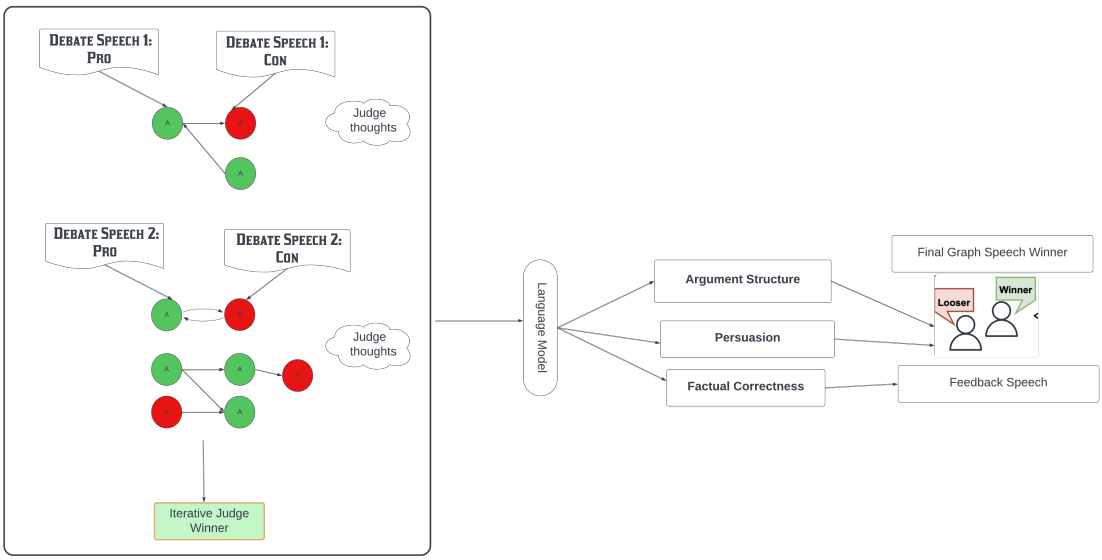



Figure 1: Overall Architecture: In Step 1, we create one graph for both sides of the debate. In Step 2, we use our scoring system to score both sides on various parameters. In Step 3, we determine the winner of the debate using the graph and our scoring system.

as claims, premises, and rebuttals. This preprocessing step enabled structured representation and downstream analysis.

### 3.2 Graph Generation

A graph-based approach was employed to represent debate arguments, using GPT-4o for the generation and refinement of scene graphs. The process involved the following steps:

- **Isolate Speeches:** For each round of debate the pro- and con-side speeches are extracted from the debate data and an empty Pydantic scene graph object is initialized for the debate graph.
- **Initial Scene Graph Creation:** GPT-4o is prompted to generate an initial scene graph as a JSON string for the pro side debater's first speech, which is then converted into a Pydantic object. The graphs included objects, attributes, and relationships relevant to the core arguments, constrained by standard debate terminology such as claims, premises, rebuttals, and first principles. The agent is instructed to derive each node and edge directly from the debaters' speech content. The full prompt for graph generation including the required constraints is available in figure 5.
- **Ensure Compliance:** Whenever a scene graph is generated, its compliance with the Pydantic object attributes is checked to ensure it is a valid graph. If the generated text is not in a valid graph format, GPT-4o is prompted again until a valid generation occurs.
- **Iterative Refinement:** Counterfactual arguments are then generated using prompts 6 and 7 with the same constraints as listed above and incorporated into the scene graph. For each round, GPT-4o adds to the graph by merging the newly generated counterfactual elements with the existing graph structure, ensuring continuity and evolution of argument representation over the course of the debate.
- **Judge Annotation:** A virtual judge maintains a history of thought processes and provides direct predictions of the winner at each refinement stage without relying on scoring metrics.

This iterative process yields a comprehensive representation of debate arguments, capturing their inherent logical structure and the back-and-forth relationship between the pro and con debaters. The detailed process for debate graph generation is shown in Algorithm 1 and all prompts can be viewed in Appendix C.

**Algorithm 1** Debate graph generation algorithm.

**Require:** Debate Speeches
**Ensure:** Direct Winner, Scene Graph
Initialize Variables:
$hist \leftarrow []$
$T \leftarrow$ topic
$SG, CG \leftarrow$ Scene Graph, Counter Graph
Initialize Functions:
$GG, GJ \leftarrow$ Generate Graph, Gen. Judge
$GPC \leftarrow$ Gen. Pro Counter
$GCC \leftarrow$ Gen. Con Counter
$MG \leftarrow$ Merge Graphs
**for** $(PS, CS)$ in Speech Pairs **do**
  **if** first round **then**
    $SG \leftarrow GG(T, PS, \texttt{hist})$
  **else**
    $CG \leftarrow GPC(T, PS, SG, \texttt{hist})$
    $SG \leftarrow MG(SG, CG)$
  **end if**
  $CG \leftarrow GCC(T, CS, SG, \texttt{hist})$
  $SG \leftarrow MG(SG, CG)$
  $\texttt{dw}, \texttt{thought} \leftarrow GJ(T, SG, \texttt{hist})$
  `hist.append(thought)`
**end for**
**return** dw, SG

### 3.3 Score Comparison

The second phase involved quantitative evaluation of arguments based on three metrics: factual accuracy, persuasion, and argument structure. These metrics were computed as follows:

- **Factual Accuracy:** Each argument was assessed for factual correctness by querying the model to provide a binary label (true or false) along with a confidence score. The scores were normalized to a [0,1] range by multiplying the binary label and confidence score. (Tian et al., 2023) showed how asking a model for its confidence often results in a more explainabe and accurate output. We don't use an external fact-checker, or connect the model to google because we need our judge to serve the role of an actual judge in a debate, who would just serve as a "common knowledge" fact checker.

- **Persuasion Score:** For rebuttal pairs in the graph, the model was asked which argument was more persuasive based on contextual guidelines derived from a debate judging handbook. This binary strategy is used in multiple Argumentation papers to find which side is more persuasive (Gretz et al., 2019; Joshi et al., 2023). The winning argument in each pair was assigned a normalized persuasion score.

- **Argument Structure:** A structural score was assigned to claims that were supported by premises or rebuttals. Sides got points for each claim that had a supporting premise, and for each rebuttal to the opposing side's claim. This is to give teams credit for expanding on their claims and responding to the other side respectively. The total structural score was normalized for consistency across debates.

### 3.4 Feedback Generation

For factual accuracy and persuasion, the model was also prompted to give a max of 250 word feedback. For factual accuracy, this was restricted to facts the model thought was incorrect, and for persuasion, and overall feedback was given to both sides, comparing which side was better and why. This detailed feedback, along with a compressed 500 word feedback was given to the user. The compressed feedback was created by prompting the model to summarize the key findings of the original feedback.

The scores from these metrics were aggregated to determine the overall performance of each debater from 0-3. This aggregation provided a quantitative basis for winner determination and informed the feedback generation process. The detailed algorithm for the scoring process is shown in Algorithm 2.

### 3.5 Winner Determination and Feedback Generation

The debate winner was determined based on the combined scores from the above metrics. Feedback was generated to provide detailed insights into the factual accuracy, persuasiveness, and structural validity of each argument. For closely contested debates, additional feedback highlighted the competitive nature of the results.

## 4 Results

### 4.1 Baseline Comparisons

We evaluated the performance of our system against several baselines, including GPT-3.5, GPT-

| Model | DebateArt | | BP-Competition |
|---|---|---|---|
| | SC RMSE ↓ | DP RMSE ↓ | Accuracy ↑ |
| GPT-3.5 | 69.24 | 63.24 | 17.2 |
| GPT-3.5/COT | 63.4 | 62.66 | 20.6 |
| GPT-4 | 55.23 | 44.7 | 46.26 |
| Chronological | 48.61 | 48.7 | 30.30 |
| Dimensional | 44.91 | 45.01 | 12.12 |
| NonIterative | 44.18 | 44.03 | 36.36 |
| Debatrix | 42.21 | **41.75** | 51.52 |
| Ours | **40.31** | 43.61 | **57.89** |

Table 1: Results Table. Best score is shown in bold. Lower is better for RMSE.

3.5 with Chain-of-Thought (CoT) reasoning, GPT-4, and Debatrix (a state-of-the-art system for winner prediction without feedback). The results in table 1 show that our method is comparable to existing SOTA methods on the DebateArt dataset, achieving marginally better RMSE on score comparison tasks while slightly trailing on direct prediction tasks.

### 4.2 Performance on DebateArt Dataset

Our method demonstrated competitive results in the DebateArt dataset:

- **Direct Prediction (DP):** The judge agent directly determined the debate winner and was evaluated against the ground truth using RMSE metrics.
- **Score Comparison (SC):** Scores across three metrics (factual accuracy, persuasion, and argument structure) were summed, and the predicted winner was compared to the ground truth. Our approach edged out Debatrix in this evaluation.

Overall, we see that our method combined with score comparison outperforms all existing baselines, including Debatrix using direction prediction, which is the current state-of-the-art.

### 4.3 Performance on BP-Competition Dataset

On the BP-Competition dataset, which involves multi-team debates, our system achieved state-of-the-art accuracy, outperforming all baselines.

### 4.4 Feedback Quality

Feedback was evaluated using three methods:

- **Inter-Annotator Agreement:** We achieved a Cohen's Kappa score of 0.82, indicating strong agreement among annotators.
- **Argument F1 Score:** This metric, the geometric mean of argument recall and precision, was significantly higher for our system (0.78) compared to GPT-3.5 (0.22), GPT-3.5 + CoT (0.24), and GPT-4 (0.43).
- **Expert Judgments:** Human experts preferred our feedback 90% of the time and rated it as more specific compared to other methods.

Figure 2 shows an example feedback that our users see.

### 4.5 Analysis of Example Graph

In figure 3 we see an example of a fully formed graph. Notice that this graph contains a multitude of connections linking the two debaters' arguments together. In Appendix A, one can see the evolution of the debate graph over each round of debate. It is interesting to note that after the pro debater's first turn there are only claims, premises, and first principles as the debater is setting up their argument and has not yet had an opportunity to respond to the opposing debater's arguments with a rebuttal. However, in the con debater's first turn rebuttals begin to be included in the graph. In rounds 2 and 3 the debaters are mostly finished setting up their claims and premises and more attention is focused on rebutting the opposing debater's arguments; correpondingly, we see that the majority of new relations introduced in rounds 2 and 3 of the debate graph generation process are rebuttal relations, indicating a shift in the argument structure between rounds.

### 4.6 Overall Results

Our approach delivered robust debate evaluation and feedback capabilities. While marginally trailing on the DP task, we excelled in SC tasks and

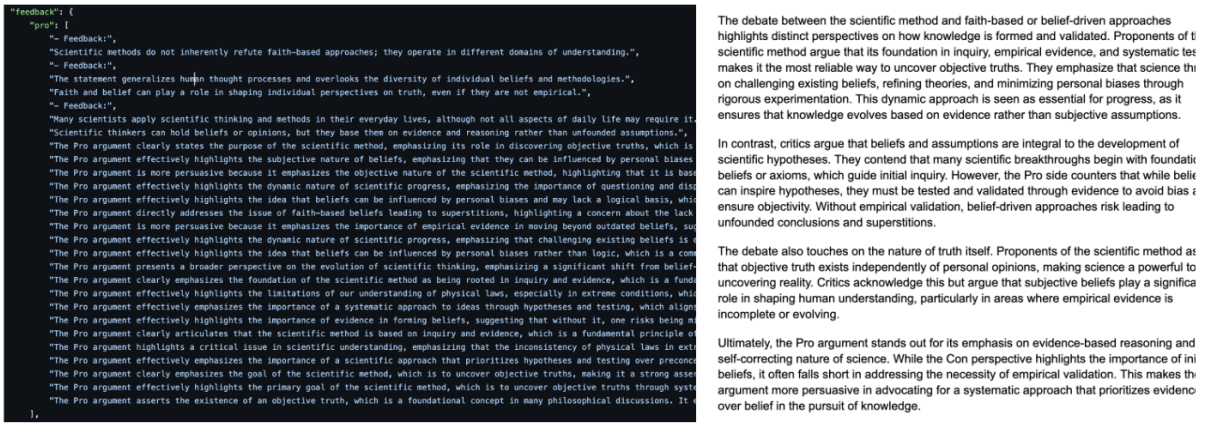



Figure 2: Example feedback: Left: Detailed feedback provided by our original models; Right: Condensed feedback after summarizing key points from the main feedback

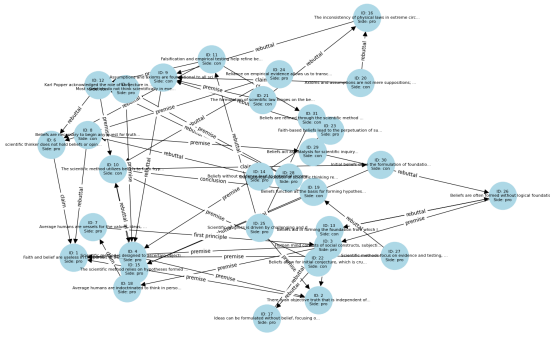


Figure 3: Final debate graph for topic: "Faith and belief are useless in the pursuit of truth."

feedback quality. The results highlight the strength of our graph-based approach in providing both accurate predictions and meaningful feedback.

## 5 Conclusion and Future Work

This paper introduced **DebGraph**, a graph-based framework combining Knowledge Graphs (KGs) and Large Language Models (LLMs) for multidimensional debate evaluation. DebGraph excels in winner prediction and feedback generation, leveraging structural representations and contextual reasoning to assess debates comprehensively. Through the project, we identified key challenges and opportunities in automated debate evaluation, offering valuable lessons for advancing this field.

### 5.1 Lessons Learned

- Constructing a graph-based representation that accurately captures the nuances of debates revealed the challenges of encoding relationships between arguments, rebuttals, and counterfactuals. We learned that balancing graph complexity with computational efficiency is crucial to maintaining scalability without sacrificing performance.

- While graphs provide a structural view of debates, incorporating contextual reasoning from LLMs proved essential for evaluating factors such as persuasiveness and rebuttal strength. This project underscored the importance of blending structural and contextual elements to capture the full dynamics of argumentation.

- Scoring debates based on discrete metrics such as factual accuracy, persuasion, and structure showed the limitations of predefined criteria. Arguments often operate on multiple dimensions simultaneously, making it challenging to holistically evaluate their impact. This highlighted the need for more granular and adaptable scoring mechanisms.

- Generating feedback that is both actionable and meaningful required designing scoring criteria that not only evaluate arguments but also offer constructive insights for improvement. This highlighted the importance of prioritizing clarity and specificity in feedback generation to make the system useful for real-world applications.

### 5.2 Future Work

Building on these lessons, we propose the following directions for improvement:

- **Weighted Edges:** Enhance graph representations by assigning weighted edges to reflect

the relative importance of arguments. This will allow the system to better evaluate debates where one strong argument outweighs multiple weaker ones.

- **Expanded Scoring Criteria:** Incorporate additional factors such as rebuttal strength, clarity, relevance, and burden of the motion to develop a more nuanced scoring system that aligns with real-world judging standards.

- **Diverse Datasets:** Expand the dataset to include debates from social media platforms, news forums, and Presidential debates to improve the model's ability to generalize across different formats and argumentation styles.

These improvements will further strengthen DebGraph's utility as a comprehensive tool for automated debate evaluation and feedback generation.

## References


A. Bhatia. 2023. Advancing policy insights: Opinion data analysis and discourse structuring using llms. *Journal of Computational Policy Analysis*.

V. Feng and G. Hirst. 2011. Classifying arguments by scheme. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 987–996.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2019. A large-scale dataset for argument quality ranking: Construction and analysis.

M. Guan, Z. Qiu, F. Li, and Y. Xue. 2023. Semantics-aware dual graph convolutional networks for argument pair extraction. In *Proceedings of the 2023 Language Resources and Evaluation Conference (LREC)*. ACL Anthology.

Omkar Joshi, Priya Pitre, and Yashodhara Haribhakta. 2023. ArgAnalysis35K : A large-scale dataset for argument quality analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13916–13931, Toronto, Canada. Association for Computational Linguistics.

A. Khan, J. Hughes, D. Valentine, and L. Ruis. 2024. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*.

Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. 2021. Employing argumentation knowledge graphs for neural argument generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4744–4754. Association for Computational Linguistics.

J. Liang, M. Ye, R. Han, R. Lai, X. Zhang, X. Huang, and Z. Wei. 2024. Debatrix: Multi-dimensional debate judge with iterative chronological analysis based on llm. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14575–14595. Association for Computational Linguistics.

X. Mou, Z. Li, H. Lyu, J. Luo, and Z. Wei. 2023. Unifying local and global knowledge: Empowering large language models as political experts with knowledge graphs. In *Proceedings of the ACM Web Conference*. Association for Computing Machinery.

Moritz Plenz, Philipp Heinisch, Anette Frank, and Philipp Cimiano. 2024. Pakt: Perspectivized argumentation knowledge graph and tool for deliberation analysis. In *Findings of the Association for Computational Linguistics: ACL 2024*. With supplementary materials.

P. Potash and A. Rumshisky. 2017. Towards debate automation: A recurrent model for predicting debate winners. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2475. Association for Computational Linguistics.

F. Ruggeri, M. Lippi, and P. Torroni. 2021. Tree-constrained graph neural networks for argument mining. *arXiv preprint arXiv:2110.00124*.

R. Ruiz-Dolz, S. Heras, and A. García-Fornes. 2022. Automatic debate evaluation with argumentation semantics and natural language argument graph networks. *arXiv preprint arXiv:2203.14647*.

W. Shang and X. Huang. 2024. A survey of large language models on generative graph analytics: Query, learning, and applications. In *Proceedings of the Annual Conference on Neural Information Processing*. Neural Information Processing Systems.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

H. Wachsmuth, G. Lapesa, E. Cabrio, and A. Lauscher. 2023. Argument quality assessment in the age of instruction-following large language models. In *Proceedings of the International Conference on Computational Models of Argument*. IOS Press.

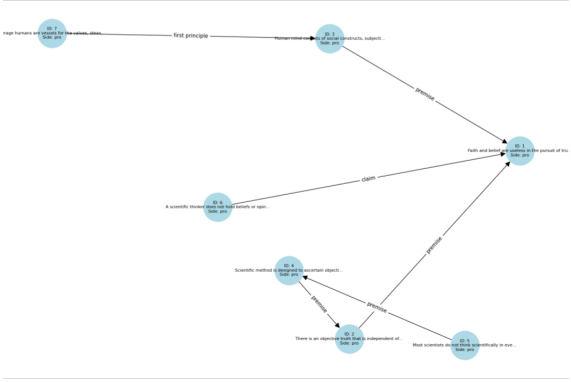
Gechuan Zhang, Paul Nulty, and David Lillis. 2023. Argument mining with graph representation learning. In *Proceedings of the Nineteenth International*

*Conference on Artificial Intelligence and Law (ICAIL '23)*, pages 371–380. Association for Computing Machinery.

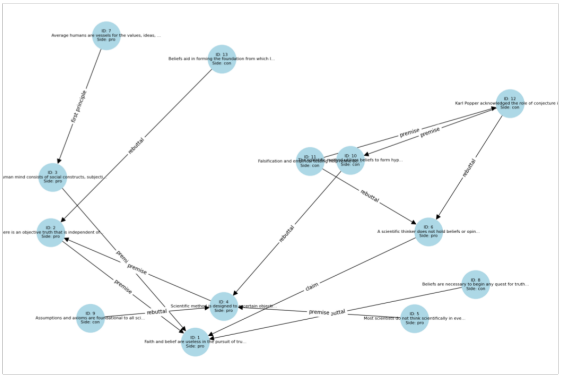
# A Appendix: Debate Graph Evolution

Round 1:

Pro:

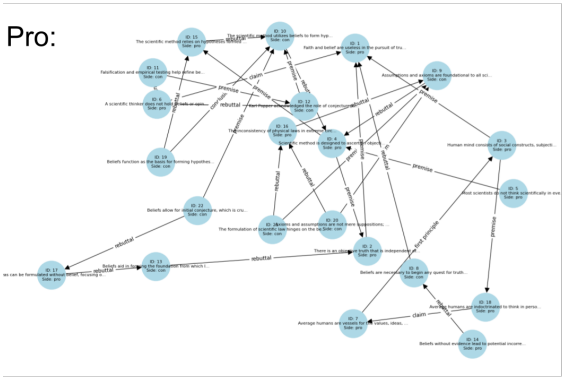


Con:

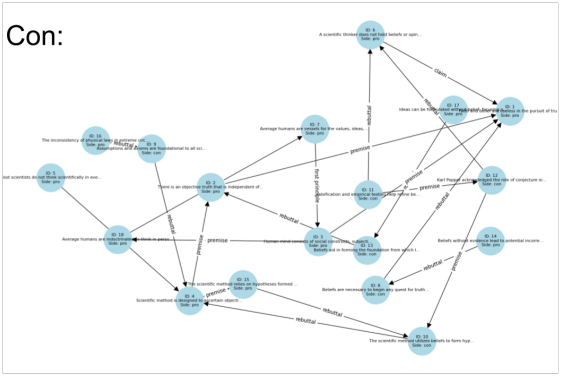


Round 2:

Pro:

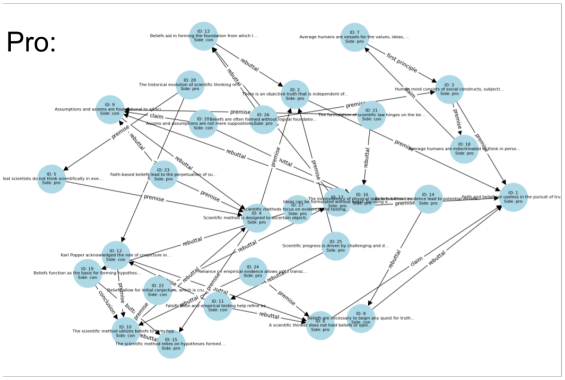


Con:



Round 3:

Pro:



Con:

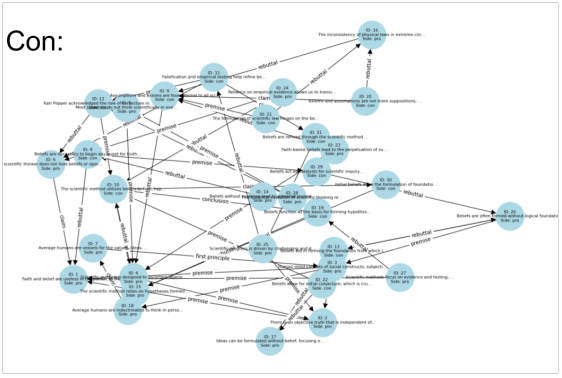


Figure 4: Complete debate graph history for topic: "Faith and belief are useless in the pursuit of truth."

# B Appendix: Methodology: Algorithm for scoring

**Algorithm 2** Debate Evaluation Algorithm

**Require:** `Pro_Arguments`, `Con_Arguments`, `Rebuttals`, `Relationships`, `Entities`
**Ensure:** `Overall Winner`, `Feedback`

Initialize `Pro_Score`, `Con_Score` $\leftarrow 0$
Initialize `Pro_Feedback`, `Con_Feedback` $\leftarrow \emptyset$
**Step 1: Factual Accuracy Evaluation**
**for** each argument $A_i$ in `Pro_Arguments` **do**
  $(F_i, C_i, \texttt{Feedback}) \leftarrow$ Evaluate_Factual_Accuracy$(A_i)$
  Update `Pro_Score` with $F_i \times C_i$
  **if** `Feedback` is not `None` **then**
    Append `Feedback` to `Pro_Feedback`
  **end if**
**end for**
**for** each argument $A_i$ in `Con_Arguments` **do**
  $(F_i, C_i, \texttt{Feedback}) \leftarrow$ Evaluate_Factual_Accuracy$(A_i)$
  Update `Con_Score` with $F_i \times C_i$
  **if** `Feedback` is not `None` **then**
    Append `Feedback` to `Con_Feedback`
  **end if**
**end for**
Normalize `Pro_Score` and `Con_Score` by dividing by the total number of arguments.
**Step 2: Persuasiveness Evaluation**
**for** each `Pro`, `Con` rebuttal pair $(R_{\text{Pro}}, R_{\text{Con}})$ **do**
  $(P, \texttt{Feedback}) \leftarrow$ Evaluate_Persuasiveness$(R_{\text{Pro}}, R_{\text{Con}})$
  **if** $P$ is `Pro` **then**
    Increment `Pro_Score_Persuasion`
    **if** `Feedback` is not `None` **then**
      Append `Feedback` to `Pro_Feedback`
    **end if**
  **else**
    Increment `Con_Score_Persuasion`
    **if** `Feedback` is not `None` **then**
      Append `Feedback` to `Con_Feedback`
    **end if**
  **end if**
**end for**
Normalize `Pro_Score_Persuasion` and `Con_Score_Persuasion`.
**Step 3: Argument Structure Evaluation**
(`Pro_Count`, `Con_Count`) $\leftarrow$ Count_Nodes_With_Relationship(`Relationships`, `Entities`)
Normalize the counts to calculate structural scores for both sides.
**Step 4: Overall Score Calculation**
`Total_Pro_Score` $\leftarrow$ `Pro_Score` + `Pro_Score_Persuasion` + `Structural_Pro_Score`
`Total_Con_Score` $\leftarrow$ `Con_Score` + `Con_Score_Persuasion` + `Structural_Con_Score`
**if** `Total_Pro_Score` $>$ `Total_Con_Score` **then**
  `Overall Winner` $\leftarrow$ `Pro`
**else if** `Total_Con_Score` $>$ `Total_Pro_Score` **then**
  `Overall Winner` $\leftarrow$ `Con`
**else**
  `Overall Winner` $\leftarrow$ `Tie`
**end if**
**Output:** Return `Overall Winner` and `Feedback`

# C Appendix: Graph Generation Prompts

```
def create_scene_graph_prompt(topic, side, speech):
    base_prompt = f"""
    You are an AI agent tasked with analyzing a debate. The topic of the debate is: "{topic}".
    You are representing the {side} side. Below is a speech from your side:

    "{speech}"

    Based on this speech without making up new arguments, generate a scene graph in JSON format that includes the following:
    1. Arguments, and relationships that are most relevant to understanding your side's speech and point of view.
    2. Ensure the relations used are one of the following: ["claim", "premise", "conclusion", "first principle", "rebuttal"]
    3. The scene graph should be structured to clearly represent the key points and relationships in the argument.
    4. All ids should be unique

    Here is an example scene graph for climate change, you should follow this format in your generated scene graph:

    "{example_graph}"

    Please provide the scene graph between the markers <SCENE_GRAPH> and </SCENE_GRAPH>
    """

    return base_prompt
```

Figure 5: Prompt used to generate the initial debate scene graph.

```
def create_pro_counterfactual_scene_graph_prompt(topic, pro_speech, con_scene_graph):
    base_prompt = f"""
    You are an AI agent representing the pro side in a debate. The topic of the debate is: "{topic}".
    Below is a speech from your side:

    "{pro_speech}"

    Additionally, here is the current scene graph representing the debate which has just been modified by the con side:

    {json.dumps(con_scene_graph, indent=2)}

    Your task is to generate counterfactual argument entities and relationships in the same JSON format that refutes the con side's arguments and introduces new arguments for the pro side.
    All of your arguments should be based on the given pro side speech, do not make up new arguments that are not mentioned in the pro side speech. The counterfactual scene graph should include the following:
    1. Arguments, and relationships that directly counter the con side's scene graph.
    2. Arguments, and relationships that support the pro side's arguments.
    3. The scene graph should be structured to clearly represent the key points and relationships in the argument.
    4. Make reference to the con side's arguments by referencing their ids in the your proposed relations.
    5. Do not include duplicates of the con side's arguments in your graph when you could instead reference their ids in your relations.
    6. Do not add any new fields to the json which are not in the provided examples.
    7. Source and Target Id should always refer to an integer of an existing argument.
    8. All generated argument ids should be unique from any other id in the current scene graph.
    9. Ensure the relations used are one of the following: ["claim", "premise", "conclusion", "first principle", "rebuttal"]

    Your proposed arguments and relationships will be added to the scene graph to form an update debate graph from which a judge can determine the winner.

    Here is an example another example scene graph for climate change which you can use as reference for the format of your json output:

    "{example_graph}"

    Please provide the counterfactual scene graph between the markers <SCENE_GRAPH> and </SCENE_GRAPH>.

    You should additionally provide a brief explanation of the counterfactual scene graph between the markers <EXPLANATION> and </EXPLANATION>.
    """

    return base_prompt
```

Figure 6: Prompt used to generate pro side counterfactual graph.

```
def create_counterfactual_scene_graph_prompt(topic, con_speech, pro_scene_graph):
    base_prompt = f"""
    You are an AI agent representing the con side in a debate. The topic of the debate is: "{topic}".
    Below is a speech from your side:

    "{con_speech}"

    Additionally, here is the current scene graph representing the debate which has just been modified by the pro side:

    {json.dumps(pro_scene_graph, indent=2)}

    Your task is to generate counterfactual argument entities and relationships in the same JSON format that refutes the pro side's arguments and introduces new arguments for the con side.
    All of your arguments should be based on the given con side speech, do not make up new arguments that are not mentioned in the con side speech. The counterfactual scene graph should include the following:
    1. Arguments, and relationships that directly counter the pro side's scene graph.
    2. Arguments, and relationships that support the con side's arguments.
    3. The scene graph should be structured to clearly represent the key points and relationships in the argument.
    4. Make reference to the pro side's arguments by referencing their ids in the your proposed relations.
    5. Do not include duplicates of the pro side's arguments in your graph when you could instead reference their ids in your relations.
    6. Do not add any new fields to the json which are not in the provided examples.
    7. Source and Target Id should always refer to an integer of an existing argument.
    8. All generated argument ids should be unique from any other id in the current scene graph.
    9. Ensure the relations used are one of the following: ["claim", "premise", "conclusion", "first principle", "rebuttal"]

    Your proposed arguments and relationships will be added to the scene graph to form an update debate graph from which a judge can determine the winner.

    Here is an example another example scene graph for climate change which you can use as reference for the format of your json output:

    "{example_graph}"

    Please provide the counterfactual scene graph between the markers <SCENE_GRAPH> and </SCENE_GRAPH>.

    You should additionally provide a brief explanation of the counterfactual scene graph between the markers <EXPLANATION> and </EXPLANATION>.
    """

    return base_prompt
```

Figure 7: Prompt used to generate con side counterfactual graph.

```python
def create_judge_prompt(topic, scene_graph, thoughts_history):
    base_prompt = f"""
    You are an AI judge tasked with evaluating the logical validity of arguments in a debate. The topic of the debate is: "{topic}".
    Below is the scene graph generated by the pro and con sides:

    Scene Graph:
    {json.dumps(scene_graph, indent=4)}

    Additionally, here is the history of your previous thoughts:
    {thoughts_history}

    Your task is to analyze the logical structure and validity of each side's arguments based on the scene graph.
    Consider the relationships and attributes in the graph to determine the strength and coherence of the arguments. Provide your reasoning and conclusion wrapped in <thoughts> tags.
    Based on this analysis and the history of your previous thoughts, decide which side has presented a stronger argument. If the pro side has a stronger argument, write <winner>pro</winner>.
    If the con side has a stronger argument, write <winner>con</winner>. If both sides are equally strong, write <winner>tie</winner>.

    Please provide your reasoning and conclusion between the markers <thoughts> and </thoughts>, and your decision between the markers <winner> and </winner>.
    """

    return base_prompt
```

Figure 8: Prompt used to generate judge thoughts and direct winner prediction.