Thematic and Argument Structure Analysis of Plato's Republic Using Data Mining Techniques

Tanishq Daniel, Trevor Ouma, Nate Miller
CIS 635

December 4, 2024

1 Introduction

This project delves into natural language processing (NLP) and text analysis, focusing on a classic in philosophy— Plato's *The Republic*. The goal is to uncover philosophical themes and analyze the structure of arguments, particularly evaluating the complexity of these themes and their associated arguments. We aim to discover whether Plato uses more complex arguments when discussing highly intricate themes, revealing if there is an intentional alignment between the depth of his reasoning and the topics he addresses. Our motivation is to discover themes and argument structures in *The Republic* to allow for deeper insights into its complex reasoning and layered themes. Our approach will involve thematic analysis, argument structure analysis and complexity analysis. For thematic analysis we will use Latent Dirichlet Allocation (LDA) to categorize key themes in the dialogue and evaluate their complexity. Then we will classify argument types and evaluate their complexity. Finally, we will measure if Plato uses more complex arguments when discussing highly complex themes.

2 Related Work

In Latent Dirichlet Allocation (2003) the authors introduce and describe LDA as a Machine Learning technique for identifying topics. Other research teams have used this technique to evaluate philosophical texts for themes in philosophy and other subjects. The paper Eight Journals over Eight Decades by Jebeile and Kennedy (2020) applies LDA to topic modeling in the field of philosophy of science, analyzing articles from eight journals over eighty years. This project will uniquely use this technique to identify topics within a single philosophical work and then use them for complexity analysis.

3 Methods

3.1 Data Preprocessing

The preprocessing phase of this project was critical for structuring and cleaning the text before further analysis. The first step was to separate the material into distinct books using regex patterns that recognized certain markers like "BOOK I", "BOOK II", and so on. These markers, based on Roman numerals and regular formatting, allowed for precise segmentation. After being split per book, the information inside each book was further divided into smaller, edible pieces of a certain word count, ensuring consistent chunk sizes while preserving the context within each segment. After the information was separated, a comprehensive cleaning technique was performed to remove unnecessary content and focus on important language. The work began with removing stopwords from NLTK's pre-defined list of frequently used English stopwords. Furthermore, a bespoke stopword list was created to exclude context-specific terms such as character names ("Socrates," "Glaucon") and recurring keywords that added little value to the research ("said," "well," "would"). Punctuation was also deleted using Python's string.punctuation method to ensure that the processed text contained only substantive words. This comprehensive cleaning produced a noise-free, well-structured dataset suitable for subsequent processes.

3.2 Themes

3.2.1 Themes Extraction

Extracting themes from the text is the first step in analyzing complexity. This workflow leverages Latent Dirichlet Allocation (LDA), a probabilistic model used to uncover hidden topics (themes) in a corpus. For this project, we used the LDA package provided by scikit-learn. Text data was preprocessed by splitting it into books and further dividing these into manageable chunks of 350 words. Each chunk was vectorized using a bag-of-words model, transforming the text into a term-document matrix for input into the LDA model. LDA assigns probabilities to each word belonging to a specific theme and generates a set of themes, represented by the 10 most likely words for each theme. To optimize the LDA model, we created a function to evaluate hyperparameters, incorporating a combination of coherence score calculations (using Gensim) and manual validation. Manual validation involved inspecting the interpretability and relevance of the themes to the text. Through this process, we found that 15 themes produced the most interpretable and meaningful results for our analysis.

3.2.2 Themes Complexity

Calculating theme complexity involves quantifying the thematic characteristics of each chunk of text based on several key metrics. These metrics include presence, which measures the number of themes with significant strength in a chunk; diversity, representing the count of unique themes exceeding a threshold; entropy, which evaluates the distribution of thematic strengths to capture the complexity of their spread; and weighted contribution, which calculates the total thematic influence in the chunk by factoring in the complexity of each theme. Each metric is normalized to ensure comparability, and a weighted formula combines them into a composite complexity score. The weights we used are 60% weighted contribution, 20% entropy, 10% presence and 10% diversity. This score reflects the overall complexity of a chunk in terms of its thematic richness and distribution. To normalize, Z-scores are applied to individual metrics, followed by Min-Max scaling for the final composite score. The process provides a nuanced view of how themes contribute to the intellectual depth of each chunk, enabling correlations with other dimensions of analysis, such as argument complexity.

3.3 Arguments

3.3.1 Argument Extraction

The argument extraction section of this project sought to identify logical arguments in the preprocessed text. After separating the information into books and parts, each chunk was searched for terms with specified logical indications. These markers, including ”because,” ”if...then,” ”hence,” and ”therefore,” were merged into a regex pattern, allowing for the quick recognition of sentences containing arguments. Sentences inside each chunk were tokenized using the NLTK sentence tokenizer, allowing for precise sentence-level processing. Each identified sentence was saved as an argument, together with metadata like the book title, chunk number, and argument sentence. This structured method ensured that arguments were recovered systematically, including both substance and context. The output was organized as a data frame for easy analysis, with columns for book titles, chunk numbers, and extracted arguments. This stage produced a huge dataset of logical claims, ready for complexity analysis and further exploration.

3.3.2 Argument Complexity

The argument complexity research attempted to quantify the difficulties of the extracted arguments using a variety of criteria. Each argument was evaluated based on its word count, frequency of logical connectives (e.g., ”if...then,” ”because,” ”hence”), average word length, and categorization as a ”conditional” or ”general” argument (with ”if...then”). These metrics provided a thorough evaluation of the structure and richness of each logical claim. To ensure consistency amongst arguments, the measurements were normalized using Min-Max Scaling, which converted them into a standardized 0-1 range. A composite difficulty score was calculated using a weighted technique, which included 40% word count, 40% frequency of logical connectives, and 20% average word length. This weighted strategy emphasized the relevance of argument length and the presence of logical indications while accounting for word complexity. The resulting complexity ratings were stored in a structured data frame, enabling further research and visualization of the relationships between argument form and difficulty across the dataset.

4 Experiments, Results and Discussion

4.1 Theme Extraction and Complexity Analysis

Themes were extracted using LDA with hyperparameters tuned for optimal coherence and manual validation. For each chunk, the *theme complexity* was calculated using key metrics: presence, diversity, entropy, and weighted contribution. Z-score normalization was applied to each metric before combining them into a composite complexity score, which was subsequently normalized using Min-Max scaling.

4.2 Argument Extraction and Complexity Analysis

Arguments were extracted based on predefined logical markers. For each argument, complexity was measured using metrics such as word count, logical connectives, and average word length. A composite complexity score was calculated for each chunk based on these features, weighted appropriately.

4.3 Statistical Analyses

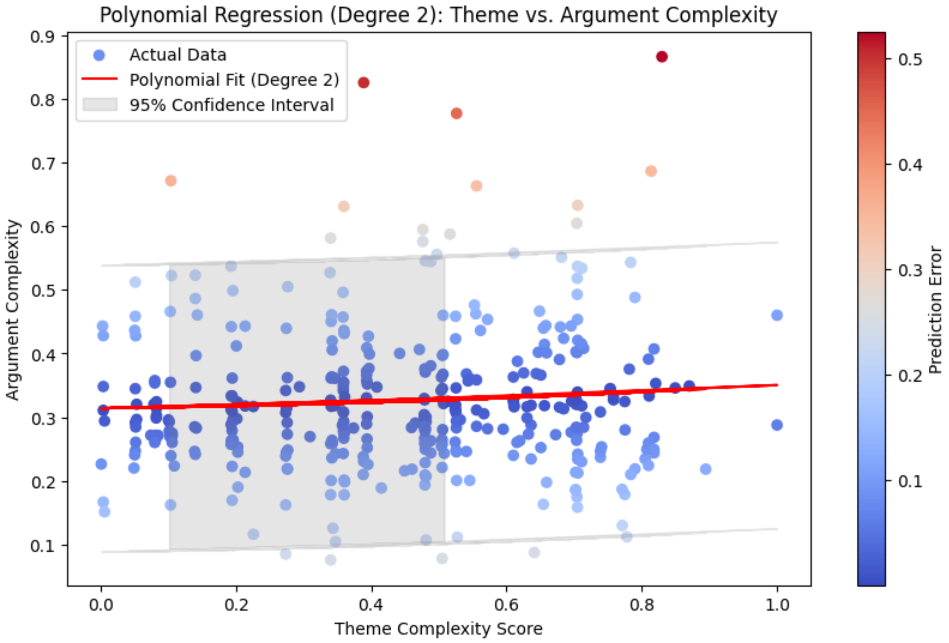
To investigate relationships between theme complexity and argument complexity, the following analyses were conducted:

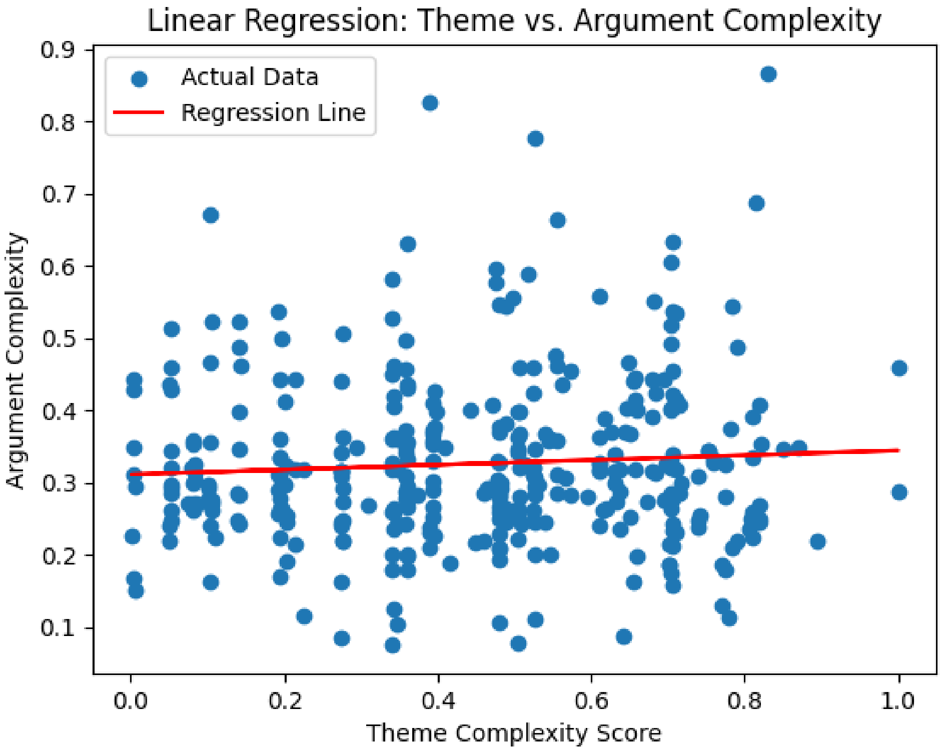
- **Linear Regression:** Explored the relationship between normalized theme complexity and argument complexity to assess linear dependencies.
- **Polynomial Regression:** Extended the analysis to capture potential non-linear trends between the variables.
- **Spearman's Rank Correlation:** Assessed monotonic relationships between theme and argument complexities.

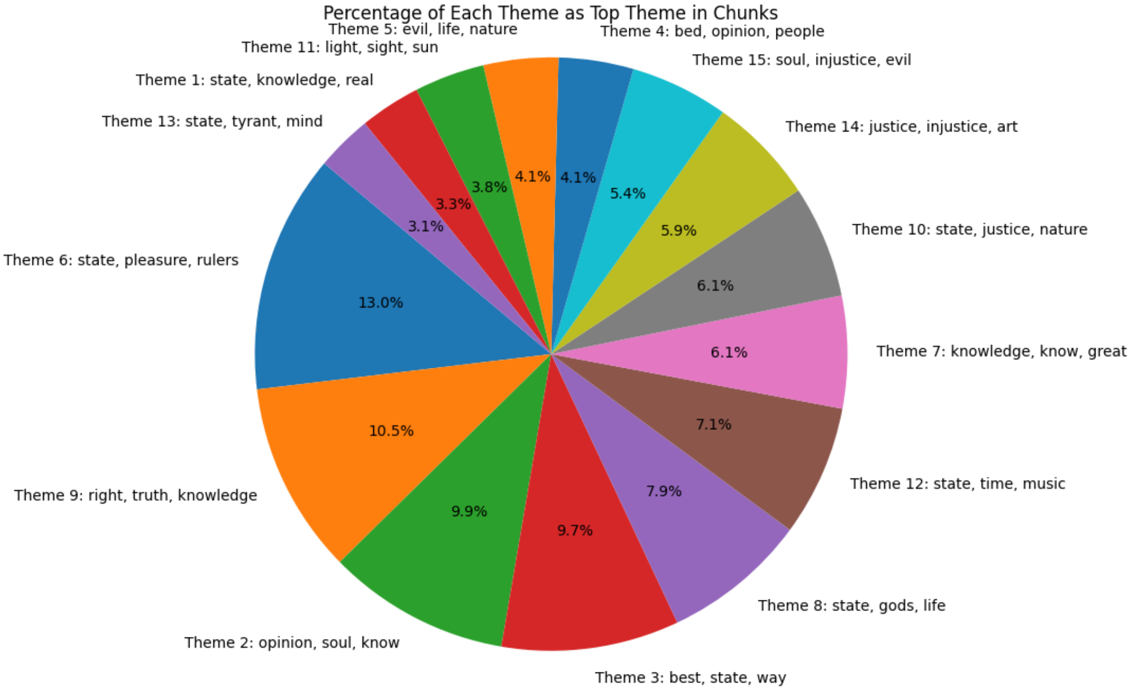
4.4 Visualization

Three primary visualizations were employed to represent the results:

1. **Scatter Plot with Regression Line:** Depicted the relationship (polynomial and linear) between theme complexity and argument complexity, highlighting trends and deviations.
2. **Pie Chart of Top Themes:** Illustrated the prevalence of each theme as the most dominant theme across all text chunks.




Linear Regression: Theme vs. Argument Complexity
Actual Data
Regression Line
Argument Complexity
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
Theme Complexity Score


Percentage of Each Theme as Top Theme in Chunks
Theme 5: evil, life, nature
Theme 4: bed, opinion, people
Theme 11: light, sight, sun
Theme 15: soul, injustice, evil
Theme 1: state, knowledge, real
Theme 13: state, tyrant, mind
Theme 14: justice, injustice, art
Theme 10: state, justice, nature
Theme 6: state, pleasure, rulers
Theme 7: knowledge, know, great
Theme 12: state, time, music
Theme 9: right, truth, knowledge
Theme 8: state, gods, life
Theme 2: opinion, soul, know
Theme 3: best, state, way
4.1%
4.1%
3.8%
5.4%
3.3%
5.9%
3.1%
6.1%
13.0%
6.1%
7.1%
10.5%
7.9%
9.9%
9.7%

4.5 Key Findings

Polynomial Regression Analysis The polynomial regression analysis revealed that theme complexity is not a significant predictor of argument complexity. Key results include:

- **$\mathbf{R^2}$ Score (0.004):** The model explains only 0.4% of the variance in argument complexity based on theme complexity. This extremely low value indicates a poor fit.
- **Mean Squared Error (0.013):** Although relatively low, this value is insufficient to indicate a strong model, especially in light of the weak R^2 score.
- **Cross-Validation Mean $\mathbf{R^2}$ (-0.011):** A negative mean R^2 score from cross-validation demonstrates that the model performs worse than a simple mean model. This confirms the polynomial regression model's unsuitability for predicting argument complexity.

Spearman's Correlation Analysis The Spearman's correlation analysis supported the findings from polynomial regression:

- **Correlation (0.052):** A weak positive correlation suggests a minimal and non-monotonic relationship between theme complexity and argument complexity.
- **P-value (0.322):** The p-value far exceeds the significance threshold of 0.05, indicating that the correlation is not statistically significant. This further implies a lack of strong evidence for a meaningful monotonic relationship between the two variables.

4.6 Conclusion

The results of both analyses suggest that there is no substantial relationship between theme complexity and argument complexity in the data. The polynomial regression model fails to explain the variability in argument complexity, as evidenced by the low R^2 score and poor cross-validation performance. Similarly, the weak and statistically insignificant Spearman's correlation reinforces this conclusion.

These findings indicate that theme complexity does not reliably predict argument complexity.

4.7 Future Directions

Given the lack of a significant relationship, the following approaches may be considered for future exploration:

- Investigating additional predictors or variables that could explain argument complexity.
- Employing advanced modeling techniques, such as feature engineering or machine learning algorithms, to capture more complex relationships.
- Exploring non-linear relationships beyond quadratic models or interactions between variables.

These directions may provide a deeper understanding of the factors influencing argument complexity and lead to more effective modeling strategies.

5 Data and Software Availability

Project Github Link:
`https://github.com/GVSU-CIS635/term-project-proposal-plato`
Plato's Republic Text Source:
`https://raw.githubusercontent.com/GVSU-CIS635/Datasets/refs/heads/master/republic.txt`

References

[1] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research.*

[2] Jebeile, J., & Kennedy, E. (2020). Eight journals over eight decades: A computational topic-modeling approach to contemporary philosophy of science. *Synthese*, 198(6), 5239–5274.

[3] Jowett, B. (1892). *Plato's Republic*. Project Gutenberg.