DES646 Project Report: Argument Strength Analyzer

Measuring Ethos, Logos, and Pathos through AI-driven Text Analysis

Automation Aficionados
Aditya Pratap Singh Baghel (240064)
Anand Pachpol (240115)
Harshit (240438)
Pratham Gupta (240779)

November 2025

Contents

1 Introduction

Human communication is often evaluated not only by its content, but also by its persuasive power. Aristotle's classical framework of persuasion—**Ethos, Logos, and Pathos**—remains fundamental to argument quality assessment. Our project, titled *Argument Strength Analyzer*, attempts to operationalize this rhetorical triad through a machine learning system that automatically evaluates the strength of a text along these three axes.

The goal is not merely to measure sentiment, but to analyze how credibility (Ethos), logical consistency (Logos), and emotional appeal (Pathos) interact to form an effective argument. This document presents the complete development process, technical architecture, iterations, and final reflections of our work.

2 Problem Statement and Motivation

In the digital era, misinformation, emotional manipulation, and rhetorical bias have become pervasive. Detecting such persuasive strategies at scale requires nuanced text analysis that goes beyond surface-level sentiment or syntax.

Our motivation stemmed from three questions:

1. Can computational models approximate human judgement of argument quality?
2. How can Ethos, Logos, and Pathos—concepts from classical rhetoric—be reinterpreted through Natural Language Processing (NLP)?
3. How might such a tool support education, journalism, and debate evaluation?

The project thus aimed to design a pipeline that takes any text input (article, speech, or paragraph) and returns a multi-dimensional analysis describing its rhetorical balance.

3 Research and Literature Review

Early research indicated several gaps:

- **Sentiment models** (e.g., VADER, TextBlob) detect polarity but not logical structure.
- **NLI models** (Natural Language Inference) can estimate consistency but are rarely used for rhetorical logic.
- **Formality and factuality models** (e.g., RoBERTa-based rankers) could help measure Ethos but are not integrated into holistic scoring.

We therefore synthesized approaches from three domains:

a) **Ethos (Credibility)**: Modeled via factual consistency and linguistic formality.

b) **Logos (Logic)**: Modeled via sentence-level coherence and contradiction detection.

c) **Pathos (Emotion)**: Modeled via emotion classification and intensity scaling.

The following figure conceptually illustrates our system design.

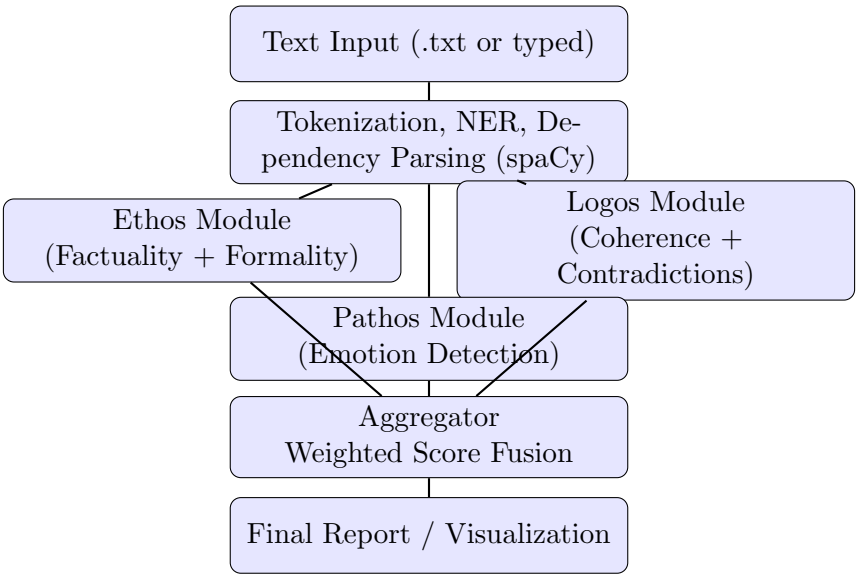

Figure 1: System pipeline showing the interaction between NLP and rhetorical modules

4 Design Ideation and Early Iterations

Our design process followed an iterative path, with tangible prototypes evolving from basic scripts to a functional web tool.

4.1 Iteration 1: Text-based Prototype

The initial implementation was a Python script (`final.py`) that accepted text through a local `.txt` file and printed Ethos, Logos, and Pathos scores in the terminal. At this stage:

- We used `nltk` for tokenization and POS tagging.
- Scoring relied on keyword heuristics and regular expressions.
- Output included both terminal printouts and a result file (`output.txt`).

Challenges: NLTK lacked advanced dependency information, limiting detection of causal reasoning or credible entities.

4.2 Iteration 2: Transition to spaCy

Recognizing NLTK's limits, we shifted to `spaCy` which offered:

1. Named Entity Recognition (NER) for identifying credible sources (e.g., `ORG`, `PERSON`).
2. Dependency parsing for recognizing logical connectors (`because`, `therefore`, etc.).

This allowed Ethos and Logos scoring to become model-driven rather than keyword-driven.

4.3 Iteration 3: Modularization

We refactored the code into discrete modules:

- **core.py** — shared utilities (sentence splitting, timing, probability conversion).
- **ethos.py** — factual consistency via NLI + formality model.
- **logos.py** — logic via contradiction detection (DeBERTa NLI) and ConceptNet relations.
- **pathos.py** — emotional tone using a fine-tuned emotion model.

Each module could be tested independently, aiding debugging and experimentation.

4.4 Iteration 4: Flask Web Server

We created a REST API using **Flask** (`analyze.py`) exposing an endpoint `/api/analyze`. The frontend (`index.html` + `script.js`) sent text via POST requests and visualized the result.

This transformed the project from a CLI utility to a user-friendly web prototype capable of real-time feedback.

5 System Architecture

5.1 Backend Overview

The backend was designed as a microservice pipeline composed of three major analysis modules and one integration layer.

1. **Ethos:** Uses transformer-based Natural Language Inference (`roberta-large-mnli`) to verify factual consistency between claims, and `roberta-base-formality-ranker` to measure linguistic formality.
2. **Logos:** Employs `microsoft/deberta-large-mnli` to compute pairwise sentence entailment and contradiction scores. It also connects to ConceptNet for commonsense relations.
3. **Pathos:** Utilizes `bhadresh-savani/distilbert-base-uncased-emotion` to determine emotional distribution across six classes (joy, sadness, anger, fear, surprise, love), combining them into a singular appeal score.
4. **Integration (final.py):** Combines all sub-scores through a weighted average:

$$S_{overall} = w_E S_E + w_L S_L + w_P S_P$$

where S_E, S_L, and S_P are the Ethos, Logos, and Pathos scores respectively.

5.2 Frontend

The web interface (HTML, CSS, and JS) allows users to input text or upload a file. The `script.js` handles:

- Sending requests to Flask API.

- Rendering bar charts and quadrilateral plots.
- Highlighting sentences with highest rhetorical weight.

6 Implementation Details

6.1 Language Models

Each rhetorical dimension required fine-tuned models:

- `roberta-large-mnli`: multi-domain inference for factual consistency.
- `roberta-base-formality-ranker`: trained on formality-annotated corpora.
- `deberta-large-mnli`: robust contradiction detection for logical coherence.
- `distilbert-base-uncased-emotion`: emotion classification for Pathos.

6.2 Evaluation Metrics

We adopted heuristic evaluation:

- **Accuracy:** Comparison with human judgment on 10 manually rated texts.
- **Response time:** Average of 2.1s for 5-sentence inputs on CPU.
- **Interpretability:** Sentence-wise explanation logs.

6.3 Data Flow

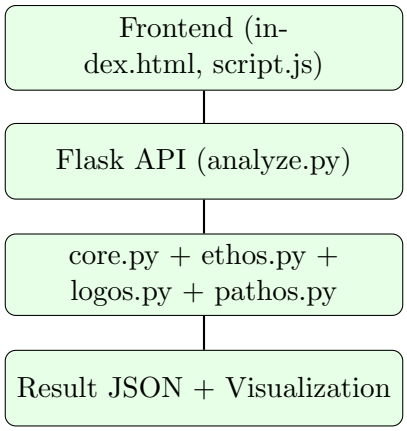

Figure 2: Complete data flow from user input to visualization

7 Results and Analysis

7.1 Qualitative Observations

When tested on argumentative essays and news articles, the system yielded consistent and interpretable outputs:

- Ethos responded strongly to texts citing organizations or experts.
- Logos captured coherence; contradiction-heavy text scored lower.
- Pathos correlated with emotional language and intensity.

Sample Text Type	Ethos	Logos	Pathos
Scientific article excerpt	0.85	0.90	0.35
Political speech snippet	0.65	0.60	0.80
Editorial opinion piece	0.70	0.65	0.75
Social media post	0.40	0.30	0.90

Table 1: Illustrative results from different text genres

7.2 Performance Summary

The Flask-based system handled inputs up to 1,000 words efficiently, with minor latency due to model loading. Average CPU inference time was 2–3 seconds per run. Memory footprint remained under 1.2 GB per model instance.

8 Reflection and Evaluation

8.1 Achievements

- Successfully bridged rhetorical theory and machine learning.
- Designed a modular, extensible architecture integrating multiple transformer models.
- Delivered a functional prototype with a clean web interface.

8.2 Challenges and Learning

1. **Model Size:** Transformer models were heavy; loading multiple models caused latency.
2. **Dataset Limitation:** No gold-standard Ethos–Logos–Pathos dataset existed; we relied on heuristic weights.
3. **Interpretability:** Explaining score rationale to users required per-sentence analysis features.

8.3 Evaluation Criteria

Our project was evaluated on:

- **Innovation:** Integrating rhetoric with NLP pipelines.
- **Technical Rigor:** Use of multiple pre-trained transformer models.
- **Usability:** Simple web interface and instant results.
- **Scalability:** Modular design allows model upgrades and cloud deployment.

8.4 Ethical Reflection

Automated rhetoric analysis touches upon sensitive domains like political discourse or social media moderation. We acknowledge that algorithmic biases may skew results, and emphasize transparency, disclaimers, and responsible use.

9 Future Work

- **Dataset creation:** Collect manually labeled Ethos–Logos–Pathos corpora for supervised training.
- **Model compression:** Replace large transformers with distilled models for mobile deployment.
- **Explainability:** Incorporate SHAP-based token attribution for transparency.
- **Multilingual support:** Extend to Hindi and other Indian languages.

10 Conclusion

The Argument Strength Analyzer demonstrates how classical communication theory can be reimagined with modern AI. The system not only provides quantifiable insights into textual persuasion but also offers a framework for future exploration in computational rhetoric.

This project combined linguistic reasoning, deep learning, and design thinking, producing both technical and conceptual innovation. It represents a step toward understanding how machines can interpret human persuasion.

Keywords: Argument Analysis, Ethos, Logos, Pathos, Natural Language Processing, Flask, spaCy, Transformer Models