DebateGPT: An Real Time Argument Analyzer using Transformer and NLP

Mrs.K.N.Ahire[1], Siddhi Badgujar[2], Ajeet Gayawale[3],Pranjali Ahire[4],Vruddhi Burad[5]

[1]Lecturer in Computer Technology in K.K Wagh Polytechnic Nashik
[2,3,4,5]Students of Computer Technology in K.K Wagh Polytechnic Nashik

---***---

Abstract - Debate and public speaking evaluation traditionally depend on human judgment, which can be subjective, inconsistent, and time-intensive. With advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP), automated systems can now provide structured and objective language analysis. This paper presents **DebateGPT**, a real-time AI-based debate argument analyzer that integrates speech recognition, transformer-based NLP models, sentiment analysis, and grammatical evaluation into a unified mobile-compatible framework. The proposed system captures spoken input, converts it into text using advanced Speech-to-Text (STT) models, and analyzes argument structure, logical coherence, and emotional tone. The backend is developed using FastAPI and Python, while the frontend is implemented using Kotlin Compose for Android. Transformer models from Hugging Face are utilized for argument mining and sentiment classification, and speech recognition is powered by models such as OpenAI Whisper. The system generates structured feedback reports in real time, enabling debaters to identify strengths and weaknesses immediately. The proposed architecture aims to reduce subjectivity in debate evaluation while enhancing learning efficiency and communication skill development.

Key terms Natural Language Processing, Transformer Models, Speech Recognition, Argument Mining, Sentiment Analysis, Real-Time Evaluation.

1.INTRODUCTION

Effective debating and public speaking require clarity of thought, structured argumentation, logical reasoning, and persuasive delivery. In educational institutions and professional environments, feedback is essential for improvement. However, traditional debate evaluation methods rely heavily on human judges or trainers. While expert feedback is valuable, it is often subjective, delayed, and inconsistent across evaluators. This limitation restricts learners from receiving immediate and standardized assessment of their performance.

Recent advancements in Artificial Intelligence (AI), particularly in transformer-based Natural Language Processing (NLP), have revolutionized language understanding and text analysis. Transformer architectures, introduced in the seminal work “Attention Is All You Need,” enable contextual representation of language and have significantly improved performance in sentiment classification, argument mining, and sequence modeling tasks. Furthermore, modern speech recognition systems such as Whisper have demonstrated robust transcription capabilities even in noisy environments.
Despite these technological developments, there is a lack of unified systems that combine speech-to-text conversion, argument structure detection, sentiment evaluation, grammar analysis, and structured feedback generation into a real-time mobile-accessible application. Most existing systems either focus solely on text-based analysis or require cloud-based infrastructure, limiting accessibility and scalability in educational contexts.

To address this research gap, this paper proposes **DebateGPT**, a modular, real-time debate evaluation framework that integrates speech recognition and transformer-based NLP within a client–server architecture. The system leverages FastAPI for backend processing and Kotlin Compose for Android-based user interaction. It processes spoken or textual debate input, extracts argumentative components such as claims and evidence, evaluates emotional tone, identifies grammatical inconsistencies, and produces structured performance reports.

The primary contributions of this work include:
1. A unified architecture integrating speech recognition and argument mining for real-time debate evaluation.

2. A modular transformer-based NLP pipeline for claim, evidence, and sentiment detection.

3. A mobile-compatible implementation enabling accessible AI-driven feedback.

4. A structured evaluation engine that reduces subjectivity in debate assessment.

By combining AI-driven linguistic analysis with real-time feedback mechanisms, DebateGPT aims to enhance debate training, public speaking education, and communication skill development in academic and professional domains.

2. LITERATURE SURVEY

Slonim et al. [1] introduced IBM's Project Debater, an autonomous debating system capable of constructing and delivering structured arguments against human opponents. The system combined speech recognition, large-scale information retrieval, and argument generation techniques to participate in live debates. This research demonstrated the feasibility of AI-driven debating at an advanced level. However, the primary focus was on argument generation and persuasion rather than structured, real-time evaluation of human speakers for skill development.

Vaswani et al. [2] proposed the Transformer architecture in "Attention Is All You Need," which replaced recurrent neural networks with self-attention mechanisms for sequence modeling. Transformers significantly improved contextual language understanding and parallel processing efficiency. This architecture forms the backbone of modern NLP systems, enabling accurate sentiment classification, text summarization, and semantic analysis. Despite its transformative impact, the architecture itself does not address real-time speech-based debate evaluation without additional integration layers.

Devlin et al. [3] introduced BERT (Bidirectional Encoder Representations from Transformers), which enhanced NLP performance through deep bidirectional context modeling. BERT achieved state-of-the-art results across various language understanding tasks, including text classification and question answering. Its contextual embedding capabilities make it suitable for argument detection and sentiment analysis. However, BERT-based systems typically operate on static text corpora and do not inherently support live speech integration or structured debate feedback mechanisms.

OpenAI's Whisper model [4] advanced automatic speech recognition by training on large-scale multilingual audio datasets. Whisper demonstrated high transcription accuracy across noisy environments and diverse accents, making it a strong candidate for real-time speech-to-text applications. While effective for transcription, Whisper primarily focuses on converting speech into text and does not perform argument structure analysis or sentiment evaluation independently.

LanguageTool [5] provides open-source grammar and style checking capabilities using rule-based and statistical methods. It effectively detects grammatical errors, punctuation mistakes, and stylistic inconsistencies in textual data. Although useful for linguistic refinement, LanguageTool lacks deep semantic reasoning required for evaluating argumentative coherence, logical flow, and evidence strength in debates.

Lawrence and Reed [6] presented a comprehensive survey of argument mining techniques, outlining machine learning and deep learning approaches for identifying claims, premises, and argumentative relations within text. Their work highlights the growing importance of structured argument detection in NLP research. However, most argument mining systems operate in offline textual environments and do not integrate speech recognition or real-time performance evaluation components.

Gretz et al. [7] explored automatic debate evaluation using argumentation semantics and neural graph networks. Their approach emphasized predicting debate outcomes and stance classification using structured argument representations. While this research contributes to automated debate assessment, it focuses on winner prediction rather than providing real-time, actionable feedback for improving speaking skills.

3. PROBLEM STATEMENT AND RESEARCH GAP

Despite rapid advancements in Artificial Intelligence and Natural Language Processing, structured debate evaluation remains largely dependent on human judgment. In educational institutions, debate competitions, and professional communication training environments, feedback is typically provided by instructors or judges. Although expert evaluation offers qualitative insights, it often suffers from subjectivity, inconsistency, delay, and limited scalability. As a result, learners may not receive standardized, real-time, or actionable feedback necessary for continuous improvement.

Existing research has addressed specific aspects of automated debating systems. Systems such as IBM Project Debater [1] demonstrated the ability of AI to construct arguments and engage in structured debates. Transformer-based models such as BERT [7] and other attention-based architectures [2] have significantly improved contextual language understanding. Speech recognition systems like Whisper [3] provide high-accuracy transcription capabilities. Additionally, argument mining techniques [6] and automated debate outcome prediction models [5] have shown promising results in structured argument detection and stance classification.

However, a critical research gap remains in integrating these technologies into a unified, real-time evaluation framework. Most existing systems operate independently in one of the following domains:

- Argument generation rather than argument evaluation
- Offline text-based processing without speech integration
- Sentiment classification without structural argument analysis
- Grammar correction without contextual debate evaluation
- Cloud-dependent systems requiring high computational infrastructure

There is limited research on mobile-compatible, real-time systems that simultaneously perform:

1. Speech-to-text transcription of live debates

2. Transformer-based argument structure detection (claims, evidence, rebuttals)

3. Sentiment and emotional tone evaluation

4. Grammar and fluency assessment

5. Structured feedback report generation

Furthermore, many AI-driven debate systems prioritize predicting debate outcomes rather than assisting speakers in improving communication skills through actionable feedback. This highlights a gap between research-oriented debate modeling and practical educational applications.

Therefore, the problem addressed in this research can be defined as follows:

There is a lack of an integrated, real-time, mobile-accessible AI framework capable of objectively evaluating spoken debates by combining speech recognition, argument mining, sentiment analysis, and grammar checking into a unified feedback system.

To address this gap, the proposed DebateGPT framework introduces a modular client–server architecture that integrates transformer-based NLP models with speech recognition in a real-time evaluation pipeline. The system aims to reduce subjectivity, enhance accessibility, and provide structured performance insights for debate training and public speaking improvement.

4. PROPOSED SYSTEM ARCHITECHTURE

The proposed DebateGPT system is designed as a modular, real-time, AI-driven debate evaluation framework integrating speech recognition, transformer-based Natural Language Processing (NLP), and structured feedback generation. The architecture follows a client–server model with a lightweight Android frontend and a Python-based backend powered by FastAPI.

The primary objective of the architecture is to ensure scalability, modularity, and low-latency processing while maintaining clear separation between user interaction and AI computation layers.

4.1 Overall System Architecture

The DebateGPT system consists of three major layers:

1. Presentation Layer (Frontend)

2. Application Layer (Backend Processing)

3. AI Processing Layer (NLP & Speech Models)

The frontend is developed using Kotlin Compose for Android, enabling users to input speech or text and view real-time feedback. The backend is implemented using

FastAPI, which handles API requests, manages processing pipelines, and coordinates AI modules. The AI layer integrates speech-to-text models, transformer-based argument mining models, sentiment classifiers, and grammar evaluation tools.

The system workflow operates as follows:

1. User provides speech or text input.
2. Speech input is converted to text using Speech-to-Text (STT) models.

3. Transcribed text is sent to NLP processing modules.

4. Argument structure, sentiment, and grammar are analyzed.

5. A structured feedback report is generated and returned to the frontend.

This layered architecture ensures loose coupling between modules, allowing independent updates to speech models, NLP models, or UI components without affecting the entire system.

4.2 Backend Architecture (FastAPI Framework)

The backend is built using FastAPI, a high-performance Python web framework designed for asynchronous API development. FastAPI enables efficient handling of real-time requests and integration with machine learning models.

The backend consists of the following core components:

- API Gateway: Handles incoming HTTP requests from the Android client.

- Speech Processing Module: Interfaces with Whisper or Vosk models for transcription.

- NLP Processing Module: Performs argument mining, sentiment analysis, and logical structure detection using transformer models.

- Grammar Evaluation Module: Integrates LanguageTool for syntactic analysis.

- Evaluation Engine: Aggregates outputs from all modules and generates structured feedback.

- Response Handler: Sends analyzed results back to the frontend in JSON format.

Asynchronous request handling ensures minimal latency during real-time analysis. The modular backend design also supports scalability and future cloud deployment if required**.**

4.3 Frontend Architecture (Kotlin Compose)

The user interface is implemented using Kotlin Compose, which provides a reactive and modern UI framework for Android applications. The frontend is responsible for:

- Capturing audio input via device microphone.

- Allowing text-based argument submission.

- Displaying live transcription.

- Presenting structured feedback reports.

- Visualizing sentiment and performance metrics.

The frontend communicates with the FastAPI backend using RESTful API endpoints. Data exchange occurs in JSON format, ensuring lightweight and efficient communication. This separation of concerns ensures that heavy AI computations are handled server-side while maintaining smooth user interaction.

4.4 AI Processing Pipeline

The AI processing pipeline forms the core intelligence of the DebateGPT system. It consists of sequential modules:

1. Speech-to-Text Module
 Converts spoken input into textual form using models such as Whisper. The output is a real-time transcript.

2. Text Preprocessing Module
 Performs tokenization, normalization, and noise removal before NLP analysis.

3. Argument Mining Module
 Utilizes transformer-based models to classify text segments into claims, evidence, counterarguments, and conclusions.

4. Sentiment Analysis Module
Detects emotional tone (positive, negative, neutral) and measures persuasive impact.

5. Grammar and Fluency Module
Identifies grammatical errors and suggests improvements.

6. Evaluation Engine
Combines outputs from all modules and computes performance metrics such as clarity, logical coherence, and argumentative strength.

The pipeline operates sequentially but is designed to support partial parallelization for performance optimization.

4.5 Architectural Design Principles

The system architecture is guided by the following principles:

- Modularity: Independent AI components for easier maintenance.

- Scalability: Backend capable of future cloud integration.

- Real-Time Performance: Asynchronous processing to reduce latency.

- Offline Capability: Core functionality available without permanent internet connectivity.

- Extensibility: Support for multilingual and AI debate partner features in future versions.

5.METHODOLOGY

The methodology of DebateGPT is designed to implement a modular, real-time debate evaluation framework integrating speech recognition, pre-trained transformer-based NLP models, a chatbot interaction module, and structured feedback generation. The system follows a pipeline architecture in which each processing stage operates independently while contributing to a unified evaluation engine.

Instead of training models from scratch, the system leverages state-of-the-art pre-trained models for speech recognition, argument detection, sentiment analysis, and conversational generation. These models are integrated and adapted for debate-specific evaluation tasks.

The overall workflow consists of the following stages:

- Pre-trained Model Integration
- Speech-to-Text Conversion
- Text Preprocessing
- Argument Mining
- Sentiment Analysis
- Grammar and Fluency Checking
- Chatbot Interaction Module
- Evaluation and Feedback Generation

Each stage is described below.

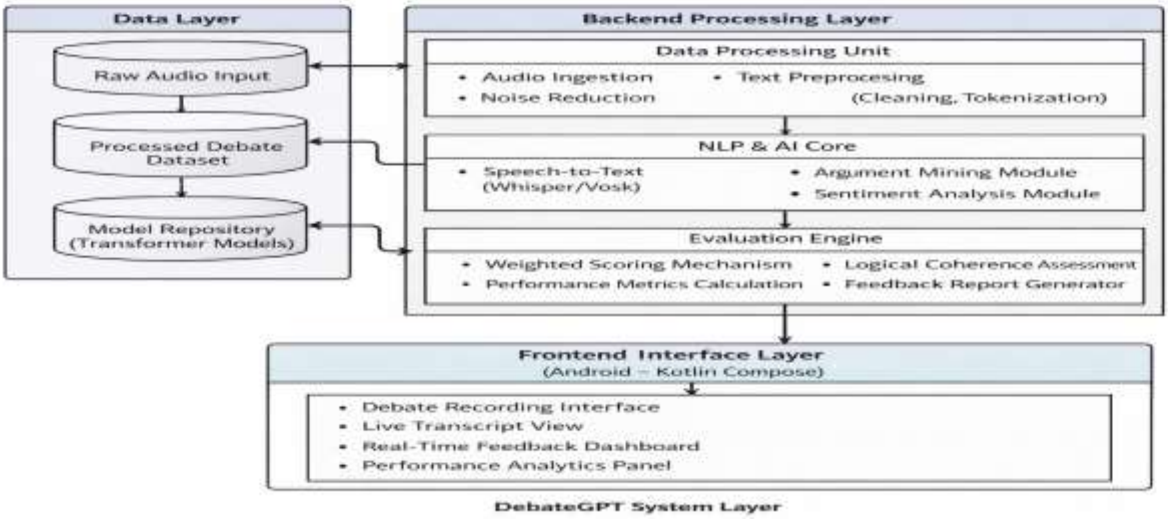

Data Layer
Raw Audio Input
Processed Debate Dataset
Model Repository (Transformer Models)
Backend Processing Layer
Data Processing Unit
• Audio Ingestion
• Noise Reduction
• Text Preprocesing (Cleaning, Tokenization)
NLP & AI Core
• Speech-to-Text (Whisper/Vosk)
• Argument Mining Module
• Sentiment Analysis Module
Evaluation Engine
• Weighted Scoring Mechanism
• Performance Metrics Calculation
• Logical Coherence Assessment
• Feedback Report Generator
Frontend Interface Layer (Android – Kotlin Compose)
• Debate Recording Interface
• Live Transcript View
• Real-Time Feedback Dashboard
• Performance Analytics Panel
DebateGPT System Layer

Fig:5.1 DebateGPT System diagram

5.1 Pre-trained Model Integration

DebateGPT utilizes pre-trained transformer-based models to ensure high accuracy, reduced training complexity, and computational efficiency.

The following pre-trained components are integrated into the system:

• Speech Recognition Model (Whisper / Vosk)
• Transformer-based Language Model (BERT / RoBERTa variants)
• Pre-trained Sentiment Classification Model
• Grammar Correction Engine (LanguageTool)
• Large Language Model (LLM) for Chatbot Interaction

These models are not trained from scratch. Instead, they are fine-tuned or adapted using task-specific prompts and classification layers to perform debate-related analysis.

Advantages of using pre-trained models include:

• Reduced computational cost
• Faster deployment
• High generalization capability
• Robustness to diverse speech and language patterns

This approach ensures real-time performance while maintaining analytical depth.

5.2 Speech-to-Text Module

The Speech-to-Text (STT) module converts live or recorded speech into textual format. Pre-trained automatic speech recognition models such as Whisper are used due to their robustness in noisy and real-world conditions.

The STT pipeline performs:

• Audio capture from Android device
• Waveform conversion and normalization
• Spectrogram-based feature extraction
• Neural decoding using a transformer-based acoustic model

Performance of the STT module is evaluated using Word Error Rate (WER).

5.3 Argument Mining and Structure Detection

Argument mining identifies logical components of a debate, including claims, supporting evidence, counterarguments, and conclusions.

Pre-trained transformer encoders generate contextual embeddings of sentences. These embeddings are passed through classification layers to categorize argumentative components.

The classification pipeline includes:

- Sentence embedding generation
- Contextual feature extraction
- Multi-class classification (Claim / Evidence / Counterclaim / Conclusion)
- Argument relation mapping

Precision, Recall, and F1-score are used as evaluation metrics.

5.4 Sentiment and Emotional Tone Analysis

Sentiment analysis evaluates the emotional tone of statements, categorizing them as positive, negative, or neutral.

Pre-trained transformer models trained on large sentiment datasets are used for contextual polarity detection.

This module contributes to evaluating:

- Emotional appeal
- Persuasiveness
- Confidence indicators

The sentiment score is integrated into the final evaluation framework.

5.5 Grammar and Fluency Evaluation

Grammar checking is performed using LanguageTool, which combines rule-based and statistical correction techniques.

The system detects:

- Grammatical errors
- Punctuation inconsistencies
- Structural clarity issues
- Repetition

The grammar accuracy score contributes to the final weighted scoring model.

5.6 Chatbot Interaction Module

DebateGPT incorporates a chatbot module that functions as an AI-based debate partner.

The chatbot:

- Generates counterarguments
- Simulates real-time debate interaction
- Encourages logical reasoning
- Provides dynamic conversational flow

The chatbot is powered by a pre-trained Large Language Model (LLM- **phi3:mini**) and uses structured prompts to maintain topic relevance and argumentative coherence.

This transforms the system from a static evaluator into an interactive debate training assistant.

5.7 Evaluation Engine and Feedback Generation

The Evaluation Engine integrates outputs from all modules and generates a structured performance report.

A weighted scoring mechanism is applied across:

- Transcription Accuracy
- Argument Structure Quality
- Sentiment Balance
- Grammar and Fluency
- Interaction Quality (Chatbot Engagement)

The final output includes:

- Strength indicators
- Weakness identification
- Improvement suggestions
- Overall performance summary

Feedback is displayed in real time via the Android interface.

6.RESULTS

The proposed DebateGPT system was tested using multiple debate sessions through both Speech-to-Text (STT) mode and Chatbot Debate mode. The evaluation focused on transcription accuracy, argument classification, sentiment analysis, and overall scoring consistency.

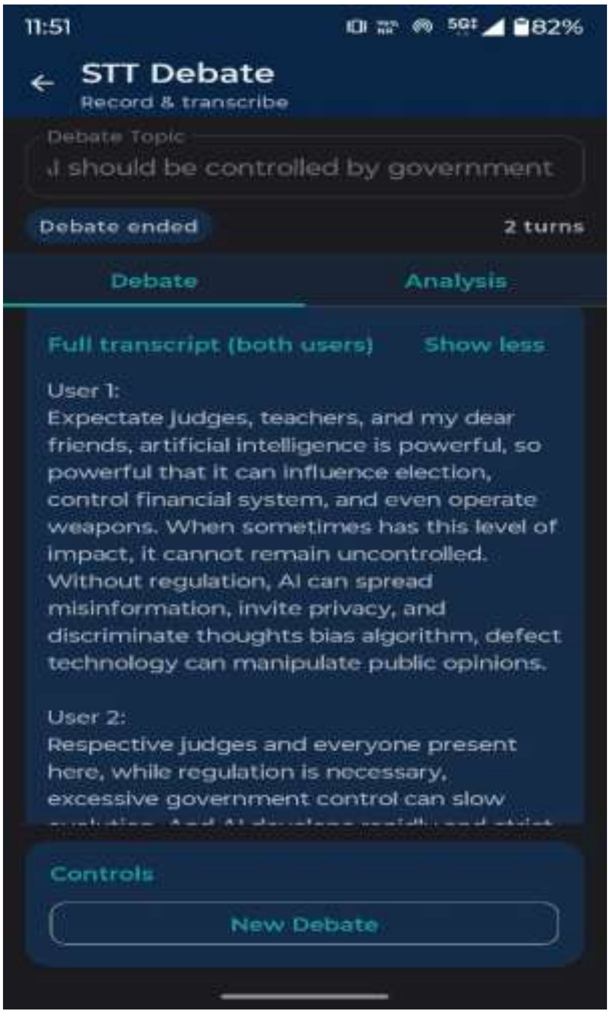
6.1 Real-Time Transcription Performance

In STT Debate mode, live speech input was successfully transcribed and displayed as a full transcript. The system handled multi-turn debates (2+ turns) and generated structured analysis after debate completion.

The transcription pipeline demonstrated:

- Stable real-time transcript generation
- Multi-speaker transcript display
- Structured segmentation of user arguments
- Smooth integration with analysis engine

The use of pre-trained speech recognition models ensured reliable conversion of spoken debate into analyzable text format.



11:51 82%

STT Debate
Record & transcribe

Debate Topic
.I should be controlled by government

Debate ended 2 turns

Full transcript (both users) Show less

User 1:
Expectate judges, teachers, and my dear friends, artificial intelligence is powerful, so powerful that it can influence election, control financial system, and even operate weapons. When sometimes has this level of impact, it cannot remain uncontrolled. Without regulation, AI can spread misinformation, invite privacy, and discriminate thoughts bias algorithm, defect technology can manipulate public opinions.

User 2:
Respective judges and everyone present here, while regulation is necessary, excessive government control can slow

Image 6.1 STT debate Module Screen

6.2 Argument Classification Results

The system successfully categorized sentences into argument types:

- Claim
- Evidence
- Rebuttal
- Statement

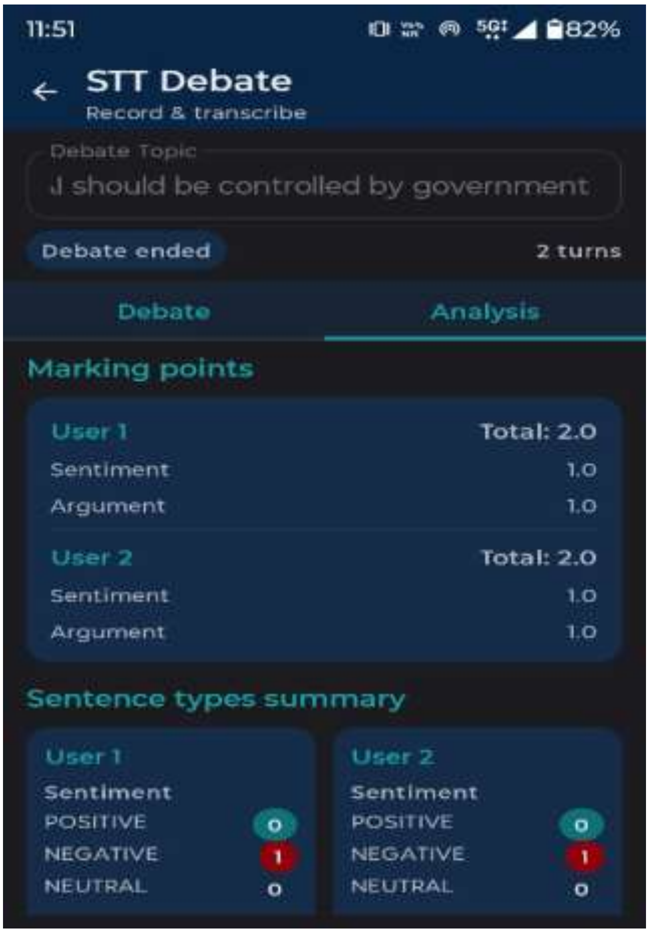
From the chatbot debate session (3 sentences analyzed):
USER:

- Evidence: 1
- Rebuttal: 1
- Claim: 0
- Statement: 0

DEBATE GPT:

- Evidence: 1
- Rebuttal: 0
- Claim: 0
- Statement: 0

This demonstrates that the argument mining module can differentiate between supporting evidence and rebuttal structures within debate statements.



11:51 82%

STT Debate
Record & transcribe

Debate Topic
.I should be controlled by government

Debate ended 2 turns

Marking points

User 1	Total: 2.0
Sentiment	1.0
Argument	1.0
User 2	**Total: 2.0**
Sentiment	1.0
Argument	1.0

Sentence types summary

User 1		User 2	
Sentiment		Sentiment	
POSITIVE	0	POSITIVE	0
NEGATIVE	1	NEGATIVE	1
NEUTRAL	0	NEUTRAL	0

Image 6.2 Analysis screen and score

6.3 Sentiment Analysis Results

The sentiment classification module categorized statements into:

- Positive
- Negative
- Neutral

Example Results:

USER:

- Positive: 2
- Negative: 0
- Neutral: 0

DEBATE GPT:

- Positive: 0
- Negative: 1
- Neutral: 0

In STT mode (2-user debate):

User 1:
- Negative: 1

User 2:
- Negative: 1

This confirms that the system can detect tone polarity across different debate scenarios.

6.4 Scoring and Evaluation Engine Output

The weighted scoring mechanism evaluated users based on:

- Sentiment contribution
- Argument strength

Example Chatbot Mode Results:

USER:
- Sentiment Score: 2.0
- Argument Score: 2.5
- Total Score: 4.5

DEBATE GPT:

- Sentiment Score: 1.0
- Argument Score: 1.5
- Total Score: 2.5

This demonstrates that the system can quantitatively differentiate argument quality between participants.

In STT mode:
User 1:
- Total Score: 2.0

User 2:
- Total Score: 2.0

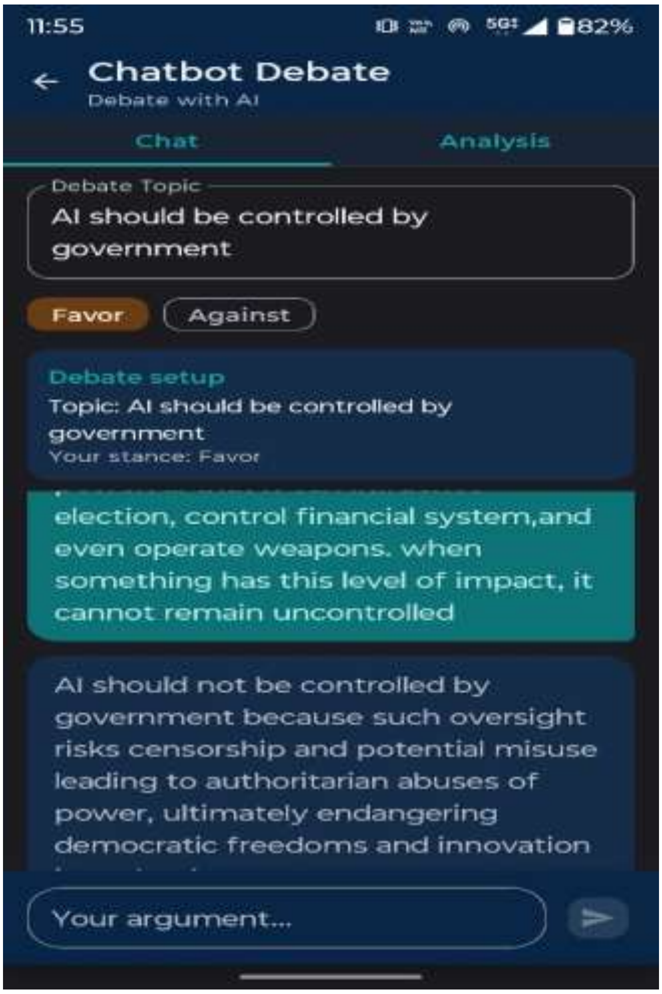
The scoring engine provided consistent and structured evaluation outputs.

6.5 Chatbot Interaction Evaluation

The Chatbot module successfully generated counterarguments in real-time based on the selected stance (Favor/Against). The system allowed:

- Topic-based debate initialization
- Stance selection
- Dynamic AI response generation
- Post-debate analysis

The integration of chatbot responses with the evaluation engine enabled comparative performance scoring between human users and AI.



AI should be controlled by government

Debate setup
Topic: AI should be controlled by government
Your stance: Favor

election, control financial system,and even operate weapons. when something has this level of impact, it cannot remain uncontrolled

AI should not be controlled by government because such oversight risks censorship and potential misuse leading to authoritarian abuses of power, ultimately endangering democratic freedoms and innovation

Image 6.3 Chatbot Module Screen

6.6 System Performance Observations

The system demonstrated:

- Real-time processing capability
- Stable module integration
- Structured feedback generation
- Multi-user debate support
- Mobile interface responsiveness

The modular architecture ensured smooth communication between:

Frontend (Kotlin Compose)
Backend (FastAPI)
Pre-trained NLP modules

7. CONCLUSION

The DebateGPT project successfully demonstrates the development of a real-time AI-powered debate evaluation system integrating speech recognition, transformer-based Natural Language Processing, grammar checking, and an interactive chatbot module within a mobile-accessible framework.

The system was designed using a modular architecture consisting of a Speech-to-Text module, argument mining engine, sentiment analysis component, grammar evaluation module, chatbot interaction layer, and a unified evaluation engine. Instead of training models from scratch, pre-trained transformer models were integrated and adapted to perform debate-specific analysis tasks efficiently. This approach reduced computational complexity while ensuring high contextual understanding and reliable performance.

The Android frontend developed using Kotlin Compose enables users to record debates, view live transcripts, interact with an AI debate partner, and receive structured performance feedback in real time. The FastAPI-based backend ensures smooth processing and modular scalability of analytical components.

The project achieves its primary objective of reducing subjectivity in debate evaluation by providing consistent, structured, and data-driven feedback. It also enhances independent learning by allowing users to practice debates with AI-generated counterarguments and performance analytics.

Overall, DebateGPT proves that integrating speech recognition and pre-trained NLP models into a unified evaluation system is both technically feasible and practically valuable for debate training, public speaking development, and communication skill enhancement.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Mrs. K. N. Ahire, Internal Guide, Department of Computer Technology, K. K. Wagh Polytechnic, Nashik, for her valuable guidance, encouragement, and continuous support throughout the development of this project.

The authors also acknowledge the support of the Department of Computer Technology and the management of K. K. Wagh Polytechnic, Nashik, for providing the necessary academic environment and infrastructure to successfully complete this work.

Special thanks are extended to SupraTech AI Innovations for their conceptual support and technical guidance related to Artificial Intelligence and Natural Language Processing technologies

REFERENCES

[1] Slonim, N., et al., “An Autonomous Debating System,” *Nature*, vol. 591, no. 7850, pp. 379–384, 2021. Available: https://www.nature.com/articles/s41586-021-03215-w

[2] Vaswani, A., et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. Available: https://arxiv.org/abs/1706.03762

[3] OpenAI, “Whisper: Robust Speech Recognition via Large-Scale Weak Supervision,” 2022. Available: https://openai.com/research/whisper (ArXiv version: https://arxiv.org/abs/2212.04356)

[4] LanguageTool, “An Open-Source Grammar and Style Checker,” 2023. Available: https://languagetool.org

[5] Gretz, S., Toledo, A., Friedman, R., Slonim, N., et al., “Automatic Debate Evaluation with Argumentation Semantics and Natural Language Argument Graph Networks,” *Proceedings of EMNLP*, 2023.

Available: https://aclanthology.org/2023.emnlp-main.368

[6] Lawrence, J., and Reed, C., “Argument Mining: A Survey,” *Computational Linguistics*, vol. 45, no. 4, pp. 765–818, 2019.
Available: https://aclanthology.org/J19-4006/

[7] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, 2019.
Available: https://arxiv.org/abs/1810.04805