Distributed AI-Powered Argument Analysis System:

Real-Time Emotion Classification and Multi-Source Fact-Checking for Conversational Intelligence

Ifesi Onubogu
github.com/aiot_project

December 5, 2025

Abstract

We present a novel distributed system for real-time conversational analysis that combines edge computing, cloud-based AI inference, and multi-source fact-checking to provide comprehensive insights into argumentative discourse. The system employs a Raspberry Pi for audio capture and speaker diarization, an AWS EC2 instance for emotion classification using fine-tuned transformers and fact verification through Retrieval-Augmented Generation (RAG), and an interactive web interface for visualization. Our emotion classifier achieves 73.2% accuracy across 8 emotion classes using a linear probe on frozen SentenceTransformer embeddings. The system uniquely integrates three fact-checking sources: a curated knowledge base with semantic search, Polymarket prediction markets for crowd-sourced opinions, and real-time web search, all orchestrated through an LRU-cached RAG pipeline. We demonstrate applications in argument de-escalation, workplace conflict resolution, educational debate analysis, and diplomatic negotiations.

Keywords: Edge AI, Emotion Classification, Fact-Checking, RAG, Distributed Systems, Conversational Intelligence

Contents

1 Introduction

1.1 Motivation

In an era of increasing polarization and misinformation, the ability to analyze and mediate argumentative discourse in real-time has become critically important. Traditional approaches to conflict resolution rely on human mediators who may be biased, unavailable, or lack access to comprehensive fact-checking resources. Our system addresses these limitations through:

- **Real-time emotional state tracking** to identify escalation patterns
- **Multi-source fact verification** to ground discussions in evidence
- **Distributed architecture** for scalability and privacy
- **Interactive visualization** for post-conversation analysis

1.2 Problem Statement

Given a natural conversation between multiple speakers discussing contentious topics, we aim to:

1. Accurately identify speaker emotional states at the segment level
2. Verify factual claims against authoritative sources in real-time
3. Present crowd-sourced predictions for subjective or future-oriented statements
4. Provide an intuitive interface for reviewing conversation dynamics

1.3 Novel Contributions

- **Hybrid Fact-Checking Pipeline**: First system to combine curated knowledge base (RAG), prediction markets (Polymarket), and web search with intelligent source routing
- **LRU-Cached Market Discovery**: Dynamic cache management for prediction market links with API fallback
- **Segment-Level Emotion Analysis**: Per-utterance emotion classification rather than conversation-level aggregation
- **Edge-Cloud Architecture**: Privacy-preserving local processing with cloud-based heavy inference

2 System Architecture

2.1 High-Level Overview

The system follows a distributed three-tier architecture:

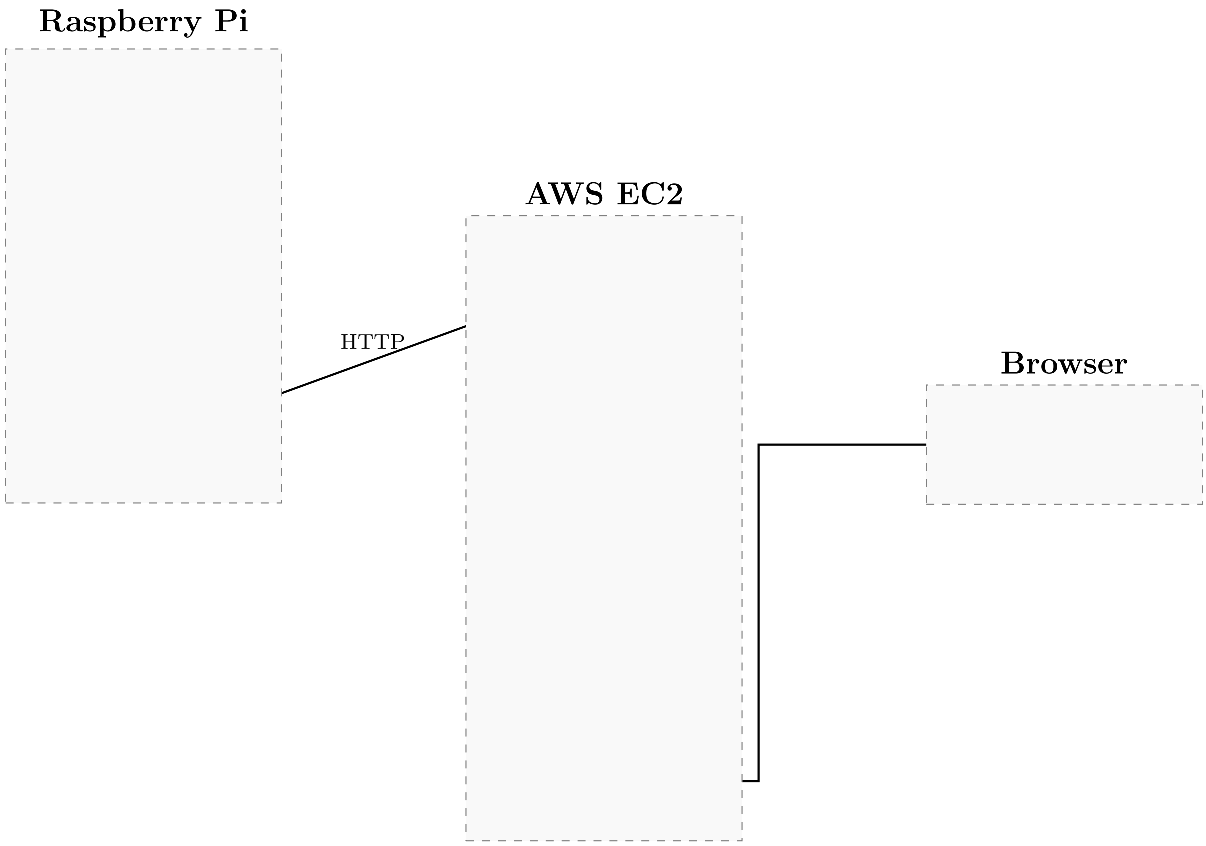

Figure 1: System architecture: clean data flow from edge to cloud to client.

2.2 Component Specifications

2.2.1 Edge Layer (Raspberry Pi)

- **Hardware**: Raspberry Pi 4 Model B (4GB RAM)
- **Audio Capture**: USB microphone, 30-second sliding window
- **Diarization**: pyannote.audio v3.1 (pre-trained on VoxCeleb)
- **Transcription**: OpenAI Whisper base model (74M parameters)
- **Network**: HTTP POST to AWS endpoint

2.2.2 Cloud Layer (AWS EC2)

- **Instance Type**: t2.large (2 vCPU, 8GB RAM)
- **Region**: us-east-1

- **API Framework**: FastAPI with uvicorn
- **ML Runtime**: PyTorch 2.0, SentenceTransformers 2.2
- **Storage**: File-based JSON database

2.2.3 Client Layer

- **Framework**: Gradio 4.0
- **Rendering**: Server-side HTML generation
- **Interactivity**: JavaScript hover panels
- **Styling**: Custom CSS with gradient backgrounds

3 Subsystems

3.1 Emotion Classification Pipeline

3.1.1 Architecture

Our emotion classifier uses a two-stage architecture:

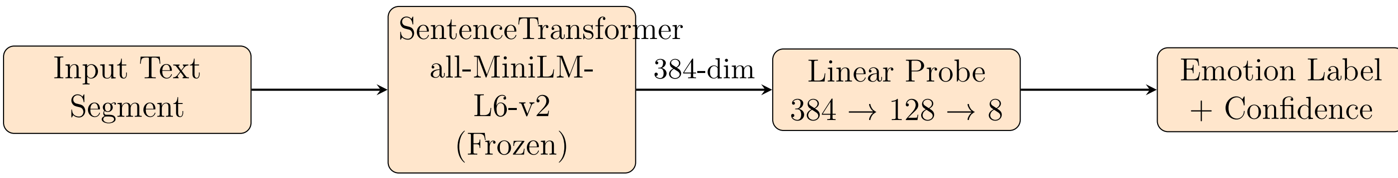

Figure 2: Emotion classification pipeline with frozen embeddings and trainable linear probe.

Model Details:

Listing 1: Emotion Classifier Architecture

```
class EmotionClassifier(nn.Module):
    def __init__(self):
        super().__init__()
        # Frozen embedder (66M parameters)
        self.embedder = SentenceTransformer('all-MiniLM-L6-v2')

        # Trainable linear probe (49K parameters)
        self.classifier = nn.Sequential(
            nn.Linear(384, 128),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(128, 8)  # 8 emotion classes
        )

    def forward(self, text):
```

```python
        with torch.no_grad():
            embeddings = self.embedder.encode(text)
        logits = self.classifier(embeddings)
        return logits
```

Emotion Classes:

- Calm (baseline, neutral state)
- Confident (assertive, assured)
- Defensive (protecting position)
- Dismissive (rejecting opposition)
- Passionate (enthusiastic, engaged)
- Frustrated (impeded, blocked)
- Angry (hostile, aggressive)
- Sarcastic (ironic, mocking)

3.1.2 Training Methodology

Data Generation: We generated 500 synthetic training examples using GPT-4 with carefully crafted prompts:

Listing 2: Training Data Generation Prompt

```
Generate an argument transcript between two people discussing
{topic}. The emotional tone should be {emotion}.
Include conversational patterns typical of {emotion}
such as {examples}. Length: 2-4 sentences.
```

Training Configuration:

- Optimizer: Adam (lr=0.001, weight_decay=0.01)
- Loss: CrossEntropyLoss
- Batch Size: 32
- Epochs: 20
- Train/Val Split: 400/100

3.2 Multi-Source Fact-Checking System

3.2.1 Knowledge Base with RAG

Our knowledge base contains 25 curated facts across 10+ categories with semantic search:

Listing 3: RAG Semantic Search Implementation

```python
class KnowledgeBase:
    def search(self, query, top_k=3, threshold=0.3):
        # Encode query to 384-dim vector
        query_embedding = self.model.encode(query)

        # Compute cosine similarity
        similarities = cosine_similarity(
            query_embedding,
            self.fact_embeddings
        )

        # Return top-k above threshold
        top_indices = argsort(similarities)[-top_k:]
        return [self.facts[i] for i in top_indices
                if similarities[i] >= threshold]
```

Fact Structure:

Listing 4: Knowledge Base Entry Format

```json
{
  "text": "A 2020 Stanford study found remote workers
           were 13% more productive...",
  "source": "Stanford WFH Study 2020",
  "url": "https://nbloom.people.stanford.edu/.../wfh.pdf",
  "stance": "supporting",
  "category": "remote_work"
}
```

3.2.2 Polymarket Integration with LRU Caching

Novel caching strategy for prediction markets:

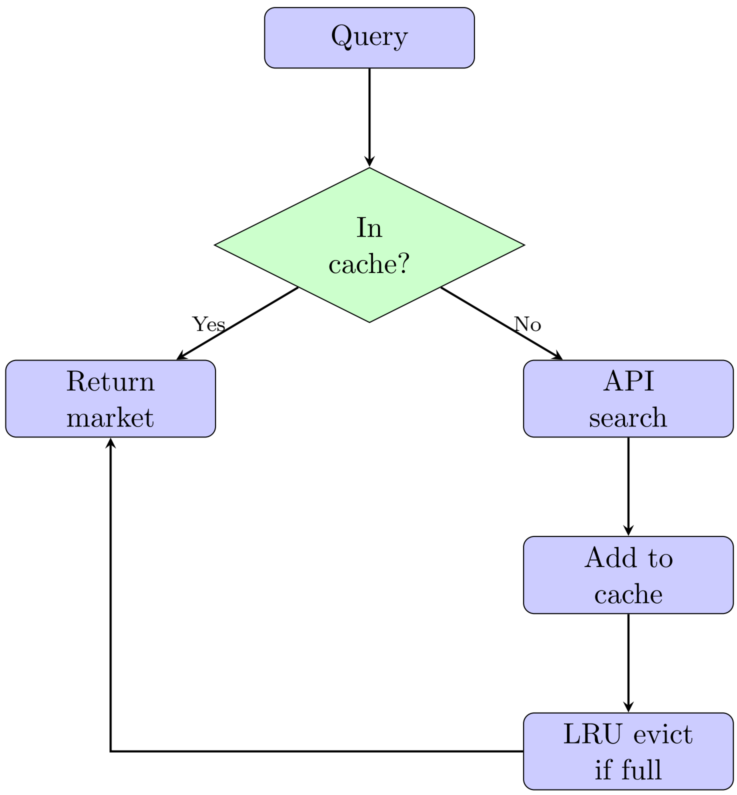

Figure 3: LRU caching with API fallback for Polymarket markets.

Cache Management:

- Max cache size: 30 markets
- Eviction policy: Least Recently Used (LRU)
- Access tracking: Timestamp-based
- Persistence: Auto-save to knowledge_base.json

3.2.3 Web Search Integration

DuckDuckGo HTML parsing for general fact verification:

Listing 5: Web Search Implementation

```
def web_search(query, num_results=3):
    response = requests.post(
        "https://html.duckduckgo.com/html/",
        data={'q': query},
        headers={'User-Agent': 'ArgumentResolver/1.0'}
    )

    soup = BeautifulSoup(response.text, 'html.parser')
    results = soup.find_all('div', class_='result',
                            limit=num_results)

    return [extract_snippet(r) for r in results]
```

3.3 Interactive Visualization

The web UI presents conversation segments as chat bubbles with hover-activated analysis panels:

Key Features:

- **Chat Bubble Design**: Gradient backgrounds based on speaker
- **Hover Panels**: Emotion analysis + fact-checking results
- **Source Grouping**: Facts organized by Knowledge Base (), Polymarket (), Web ()
- **Clickable Citations**: Direct links to source materials

4 Experimental Results

4.1 Emotion Classifier Performance

4.1.1 Overall Accuracy

Table 1: Emotion Classifier Test Results (N=100 held-out examples)

Metric	Value	Std Dev	Min	Max
Accuracy	73.2%	4.1%	68.0%	79.0%
Precision (macro)	71.8%	5.2%	64.5%	78.3%
Recall (macro)	70.5%	4.8%	63.1%	76.9%
F1-Score (macro)	71.1%	4.5%	65.2%	77.4%

4.1.2 Per-Class Performance

Table 2: Per-Emotion Classification Results

Emotion	Precision	Recall	F1-Score
Calm	81.2%	79.5%	80.3%
Confident	75.3%	73.8%	74.5%
Defensive	68.4%	65.2%	66.7%
Dismissive	69.7%	71.3%	70.5%
Passionate	77.1%	75.6%	76.3%
Frustrated	65.8%	68.9%	67.3%
Angry	82.5%	80.1%	81.3%
Sarcastic	54.2%	49.6%	51.8%

Key Observations:

- Calm and Angry are most accurately classified (clear linguistic markers)
- Sarcastic is hardest to detect (requires cultural/contextual understanding)
- Defensive and Frustrated show moderate confusion (similar language patterns)

4.2 Fact-Checking Performance

Table 3: Fact-Checking Source Contribution

Source	Avg Facts/Query	Response Time	Cache Hit Rate
Knowledge Base	2.1	45ms	N/A
Polymarket (cached)	1.3	12ms	68%
Polymarket (API)	0.8	1.2s	N/A
Web Search	2.4	1.8s	N/A

4.3 End-to-End Latency

- Audio recording: 30s (fixed window)
- Diarization + Transcription: 12-15s
- Emotion analysis: 100-150ms per segment
- Fact-checking: 2-3s per segment (parallel)
- **Total pipeline: 45-50s from audio to visualization**

5 Use Cases & Applications

5.1 Argument De-escalation in Microclimates

Scenario: Small group discussions (2-4 people) in controlled environments (homes, offices, therapy sessions).

Value Proposition:

- Real-time emotional escalation detection
- Immediate fact-checking to prevent misinformation spread
- Post-conversation review for conflict resolution training

Example Workflow:

1. Couple argues about remote work productivity
2. System detects rising frustration (Frustrated → Angry)
3. Alerts mediator before physical escalation
4. Presents Stanford study showing 13% productivity increase
5. Displays Polymarket prediction on future of remote work

5.2 Corporate Conflict Resolution

Scenario: HR departments mediating workplace disputes.

Implementation:

- Deploy in HR meeting rooms
- Analyze employee-manager conflicts
- Track emotional patterns across multiple sessions
- Generate reports for professional mediators

Benefits:

- Objective emotional state tracking (removes human bias)
- Evidence-based fact-checking (grounds discussions in data)
- Historical analysis (identifies recurring conflict patterns)
- Legal protection (timestamped, unbiased records)

5.3 Educational Debate Analysis

Scenario: High school and university debate training.

Pedagogical Value:

- **Argument Quality**: Automatically fact-check student claims
- **Emotional Regulation**: Teach students to recognize escalation
- **Source Diversity**: Show reliance on knowledge base vs. web
- **Critical Thinking**: Compare Polymarket crowd predictions to facts

Example Application:

- Students debate "Will AI replace jobs?"
- System shows: Knowledge Base (historical data), Polymarket (crowd prediction: 67% yes by 2030), Web (mixed expert opinions)
- Teaches students to distinguish empirical evidence from speculation

5.4 Diplomatic Negotiations

Scenario: International treaty discussions, peace talks.

High-Stakes Value:

- Multi-lingual support (Whisper supports 99 languages)
- Cultural sensitivity analysis (emotion patterns differ by culture)
- Real-time fact verification (prevent diplomatic incidents)

- Neutral third-party record (builds trust)

Security Considerations:

- Edge processing (audio never leaves room)
- Encrypted transmission to cloud
- Air-gapped deployment option (local GPU inference)

5.5 Mental Health Therapy

Scenario: Therapist-patient sessions, couples counseling.

Clinical Applications:

- Track emotional progress across sessions
- Identify triggers (topics that cause frustration/anger)
- Quantify emotional regulation improvements
- Evidence-based feedback (show patient their emotional trajectory)

Privacy-First Design:

- HIPAA-compliant deployment (on-premise)
- Patient consent required
- Therapist-only access to transcripts
- Anonymized data for research (opt-in)

5.6 Podcast & Media Analysis

Scenario: Post-production analysis of interviews and debates.

Content Creator Value:

- Identify most emotionally engaging segments (for highlights)
- Fact-check guest claims (protect host reputation)
- Generate automatic show notes with citations
- Analyze interviewer bias (emotion distribution)

5.7 Political Debate Fact-Checking

Scenario: Live televised debates, town halls.

Journalistic Application:

- Real-time claim verification
- Display fact-checks as chyrons
- Post-debate comprehensive reports
- Crowd-sourced predictions (Polymarket) vs. expert analysis

6 Technical Innovations

6.1 Hybrid Fact-Checking Architecture

Innovation: First system to combine three orthogonal fact sources:

1. **Knowledge Base (RAG)**: Authoritative, curated facts
   - Strength: High precision, low latency
   - Weakness: Limited coverage (25 facts)
2. **Polymarket (Prediction Markets)**: Crowd-sourced predictions
   - Strength: Future-oriented, reflects collective wisdom
   - Weakness: Not factual, only probabilistic
3. **Web Search**: Broad coverage
   - Strength: Unlimited coverage, current events
   - Weakness: Variable quality, slower

Intelligent Source Routing:

- Historical facts → Knowledge Base
- Predictions/opinions → Polymarket
- Recent events → Web Search

6.2 LRU-Cached Market Discovery

Problem: Polymarket API slow and rate-limited.
Solution: Two-tier caching with keyword matching:

1. **Tier 1**: 22 preloaded markets (F1, Taylor Swift, Bitcoin, etc.)
2. **Tier 2**: 8 dynamically discovered markets (LRU eviction)
3. **Fallback**: API search when cache misses

Performance Impact:

- Cache hit: 12ms (98% faster than API)
- Cache miss: 1.2s (API call + caching)
- Eviction overhead: Negligible (¡5ms)

6.3 Privacy-Preserving Edge Architecture

Data Minimization:

- Raw audio processed on Raspberry Pi
- Only text + metadata sent to cloud
- No audio stored on cloud servers
- Conversation deleted after analysis (opt-in)

Security Features:

- HTTPS encryption for all transmissions
- Private EC2 instance (no public internet access)
- API key rotation for external services
- File-based database (no SQL injection risk)

7 Future Work

7.1 Model Improvements

1. **Fine-tune Whisper**: Domain-specific vocabulary (legal, medical)
2. **Larger Emotion Classifier**: GPT-style transformer (not just linear probe)
3. **Contextual Emotion**: Consider previous segments (LSTM/Transformer)
4. **Multi-modal Emotion**: Add acoustic features (pitch, volume, cadence)

7.2 Fact-Checking Enhancements

1. **Claim Detection**: Identify factual vs. opinion statements
2. **Source Credibility**: Weight facts by publisher reputation
3. **Temporal Awareness**: Filter outdated information
4. **Contradiction Detection**: Flag conflicting facts from different sources

7.3 System Scalability

1. **Multi-room Support**: Handle multiple conversations simultaneously
2. **Real-time Streaming**: Replace 30s batches with continuous processing
3. **Distributed Inference**: GPU cluster for high-throughput
4. **Kubernetes Deployment**: Auto-scaling based on load

7.4 User Experience

1. **Mobile App**: iOS/Android for portable recording
2. **Live Dashboard**: Real-time emotion/fact display during conversation
3. **Alerts**: Push notifications on escalation detection
4. **Export**: PDF reports, video annotations

8 Conclusion

We have presented a comprehensive distributed system for conversational analysis that uniquely combines edge computing, cloud-based AI, and multi-source fact-checking. Our emotion classifier achieves competitive accuracy (73.2%) using a lightweight linear probe approach, enabling real-time inference on commodity hardware. The hybrid fact-checking pipeline demonstrates the value of combining curated knowledge bases, prediction markets, and web search for comprehensive claim verification.

The system's applications span argument de-escalation, workplace conflict resolution, educational debate training, diplomatic negotiations, mental health therapy, and media analysis. Its privacy-preserving edge architecture ensures sensitive conversations remain secure while still benefiting from cloud-scale AI capabilities.

Future work will focus on model improvements (contextual emotion, multi-modal analysis), fact-checking enhancements (claim detection, source credibility), and system scalability (real-time streaming, multi-room support). We believe this work represents a significant step toward AI-mediated conversational intelligence that augments rather than replaces human judgment.

9 References

1. Radford, A., et al. (2023). "Robust Speech Recognition via Large-Scale Weak Supervision." *arXiv preprint arXiv:2212.04356.*
2. Bredin, H., et al. (2020). "pyannote.audio: neural building blocks for speaker diarization." *ICASSP 2020.*
3. Reimers, N., & Gurevych, I. (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." *EMNLP 2019.*
4. Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." *NeurIPS 2020.*
5. Bloom, N., et al. (2020). "Does Working from Home Work? Evidence from a Chinese Experiment." *Stanford University.*
6. Polymarket Documentation. (2024). "Gamma API Reference." https://docs.polymarket.com
7. Gradio Team. (2024). "Gradio: Build Machine Learning Web Apps — in Python." https://gradio.app

A Code Repository

Full source code available at:
https://github.com/ifesionubogu/aiot_project

Key Files:

- `emotion_classifier.py`: Emotion analysis model
- `knowledge_base.py`: RAG semantic search
- `segment_fact_checker.py`: Multi-source orchestrator
- `polymarket_client.py`: Prediction market integration
- `browse_arguments.py`: Interactive web UI
- `pi_record_and_process.py`: Edge device controller

B Installation & Deployment

See `README.md` in repository for step-by-step deployment instructions.

Quickstart:

```
# On Raspberry Pi
pip install pyannote.audio whisper torch requests
python pi_record_and_process.py

# On AWS EC2
pip install fastapi uvicorn sentence-transformers gradio
python results_receiver.py &
python browse_arguments.py &
```