# AUTOKPI

## 1. Introduction

### Background

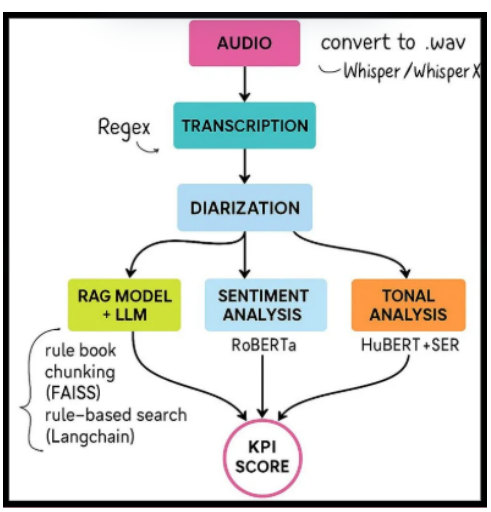
In today's era customer service is very essential in every area. Manual evaluation of any call center interaction between the agent and the customer is very time-consuming, subjective and prone to bias. Not only this, it's unscalable. Quality assurance (QA) of these calls requires a huge amount of financial resources that yield very low results; only 1-2% calls undergo manual QA process which in no way is a true representation of anything. The money spent on QA can be spent on further training and hiring of agents to reduce wait time and mishandling of any customer. With advancements in AI, there is a growing opportunity to automate agent-customer call analysis using transcription, speaker diarization, and emotion and sentiment detection. This project proposes a multimodal AI pipeline that processes call center audio calls and extracts meaningful information for evaluating each call and agent on an individual level. This system provides deep insights about customer-agent interaction.

### Objective

This system aims to automatically transcribe and analyze call center conversations. It performs speaker-wise diarized transcription, sentiment and tone detection, and evaluates conversations against a predefined rulebook using a large language model (LLM) for objective quality assessment.

## 2. System Architecture

### 2.1 Overview Diagram




### 2.2 Module Summary

- Transcription and Diarization: WhisperX / AssemblyAI
- Sentiment Analysis: CardiffNLP RoBERTa-base
- Tone Detection: Fine-tuned HuBERT model
- Rulebook-based Evaluation: LangChain + FAISS + Mistral LLM 7B

# 3. Speaker Diarization and Transcription

## 3.1 Tool Used

The system uses WhisperX for high-quality transcription and speaker diarization. WhisperX uses Voice Activity Detection (VAD) that gives word level timestamps and performs speaker diarization accordingly. First it transcribes the audio and then creates segments based on VAD and then it aligns the transcription with the audio. The models take audio files in different formats, for our model we mainly used .wav format as they are uncompressed, as input. It uses the core Whisper model for transcription. The model then uses VAD to create chunks of speech and by using Wav2Vec it aligns the audio with timestamps. It then uses pynnote-audio to identify different speakers. We hardcoded the number of speakers to 2 or 3 depending on the scenario for better and quicker results, 2 speakers where it was just the agent or customer and 3 where an automated message was at the start of call. Regardless of this specification the model is very capable of identifying on its own.

## 3.2 Output Format

Each utterance includes:

```
{
  "speaker": "SPEAKER_00",
  "text": "Hi, this is Alex from Support. How can I help you?",
  "start": 0.5,
  "end": 4.2
}
```

Text file format:

```
{
  “[7.04 - 19.44] Agent: Customer support. EICA models.”
  “[19.92 - 25.60] Customer: Hi, I'm just finding up. Is the third Ibiza, the black
one, still available?”
}
```

## 3.3 Importance

Diarization enables accurate speaker-level sentiment and tone attribution, which is performed later on. It allows us to see the call at a deeper level than just overall. It will help us see where the agent had a sentiment and tone other than neutral and help better customer experience.

# 4. Sentiment and Tonal Analysis

## 4.1 Sentiment Analysis

- **Model**: CardiffNLP's RoBERTa-base for sentiment classification. Twitter Roberta Base Sentiment is a model that analyzes the sentiment of English text. It is trained on 59 million tweets and then fine tuned for sentiment analysis. It has 3 classes which helps with quick classification. The model has a limitation of just being able to process up to 512 tokens. To deal with this we gave it diarized input to shorten text and created chunks for it to deal with long texts.

- **Classes**: Positive, Neutral, Negative.

- **Output Example**:


```
{
  "text": "Thanks so much for your help!",
  "sentiment": "positive"
}
```


## 4.2 Tone Detection

- **Model**: A HuBERT model fine-tuned for tone classification. It works with raw audio. We used it to perform speech emotion recognition (SER). it takes the audio and converts it into Mel-spectrogram. Then it learns the speech acoustics and extracts embeddings and aggregates into a single vector and then predicts probabilities of tone. To find tone of agent and caller separately, we take the time stamps from diarization performed and pick the voice form at those times to perform a separate tonal analysis for each.

- **Classes**: Happy, Sad, Angry, Calm, Fearful, Disgust, Surprise which were mapped.

- **Segment Format**:


```
{
  "speaker": "SPEAKER_00",
  "start": 1.0,
  "end": 4.0,
  "emotion": "happy"
}
```


## 4.3 Emotion Mapping Strategy

To simplify interpretation:

- Sad + Fearful = Angry

- Calm = Neutral
- Other emotions = Mapped to nearest core emotion (Positive, Neutral, Negative)

# 5. Rulebook-Based LLM Evaluation

## 5.1 Rulebook

- This step involved the use of a language model i.e. Mistral to assess how the agent performed based on a given set of instructions and rules given in a rulebook. The rulebook gives expected behaviour rules which can vary from call center to call center. So based on those rules the LLM looks at the transcribed and diarized call dialogues and checks how closely the rules were followed and gives scores based on that.

## 5.2 Retrieval Method

- FAISS vector index + LangChain Retriever pulls relevant rules based on conversation context. FAISS uses vector similarity to search through text for similarity. This was combined with LangChain that allows the retrieval to be based on chunks for search of most relevant semantic search.

## 5.3 LLM Prompting Strategy

- The model checks each rule for compliance.Mistral receives call transcription on which it performs rule by rule evaluation. It gives a score based on how well the rule was followed and in the end it provides a comment stating where the agent lost marks or where it was not compliant.

## 5.4 Output

- JSON containing boolean flags per rule and an overall performance score.


*{*
*“Resolution”: 8,*
*“Compliance”: 9,*
*“Satisfaction”: 8,*
*“Final_rating”: 8.33,*
*“Evaluation”: comments by LLM*
*}*


# 6. Applications and Benefits

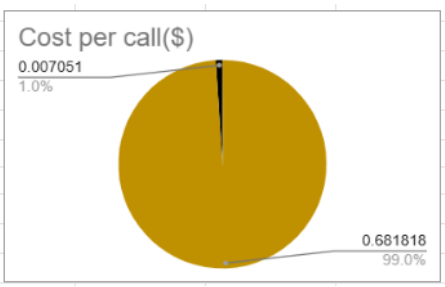
- Agent Observation: Continuous feedback based on actual calls for each individual call. Helps every agent know where they can improve.

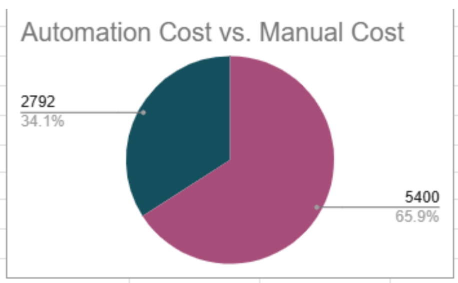
- Anomaly Detection: Warning for emotional or tonal distress in agents.
- Customer Experience Monitoring: Detects customer frustration or delight.
- Scalability: Handles thousands of hours of calls per day.

# 7. Conclusion

This system provides an end-to-end pipeline for analyzing call center audio. It delivers diarized transcription, sentiment and emotion detection, and interprets results with LLMs to evaluate agent performance objectively. With further improvements, it can serve as a core component in modern, automated QA platforms. It is very cost effective.

| | | | | | |
|---|---|---|---|---|---|
| Calls per day | 60 | | | | |
| Agents | 300 | | | | |
| Working hrs per agent | 22 | | Monthly calls | 396000 | |
| Manual QA sampling rate | 2% | | Calls sampled for QA | 7920 | |
| Calls per QA officer can handle manually | 40 | | | | |
| calls QA per months | 880 | | QA officer required | 9 | |
| Salary per agent($) | 600 | | QA officer salary($) | 5400 | |
| AUTOMATED | | | | | |
| whisper | 792 | | COST PER CALL($) | | in PKR |
| sentiment + tonal | 1000 | | MANUAL | 0.681818 | 190.9091 |
| rulebook | 1000 | | AUTOMATED | 0.007051 | 1.974141 |
| total | 2792 | | | | |
| | | | Saving per call | 188.9349 | |
| | | | YEARLY SAVING | 2992730 | |

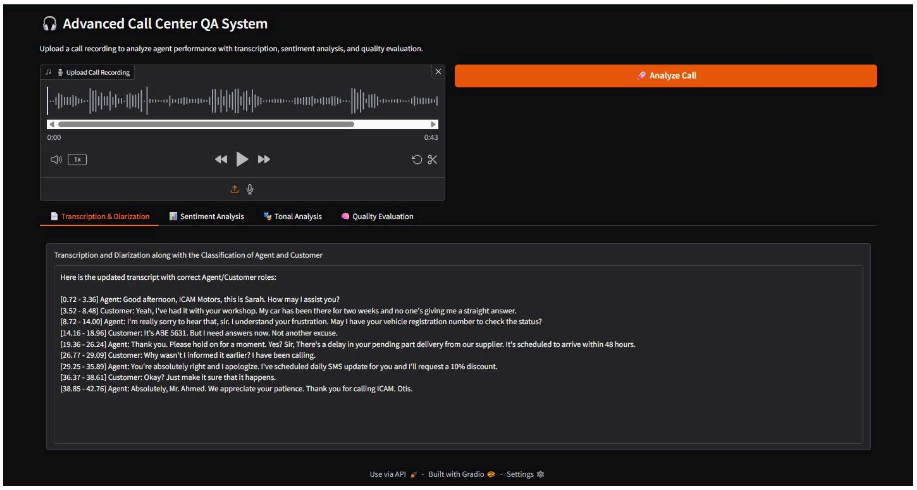


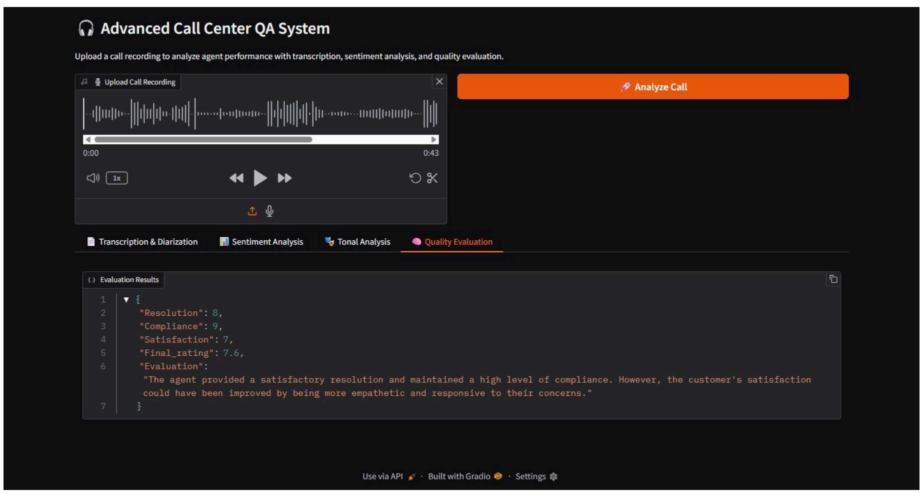



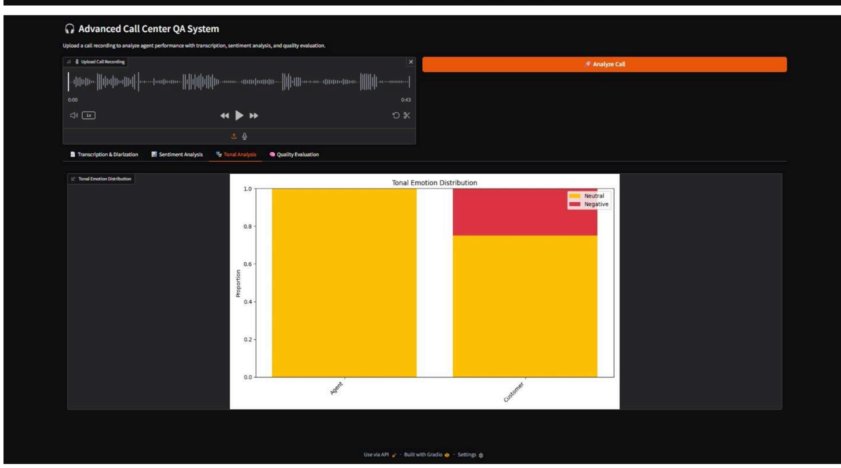

# 9. Related Works

- Scorebuddy - Calculates call center KPIs but doesn't perform sentiment and tonal analysis or a check with rulebook. Also doesn't perform agent KPI analysis.
- enthu.ai - gives a form to fill and is not fully automated to analyse itself.
- AmplifAI  - no tonal analysis or rulebook check.
- CXone - no tonal analysis or rulebook check.

# 10. Dashboard



## Contribution Matrix

| **Team Member** | **Work Done** | **Total Contribution(%)** |
|---|---|---|
| Kashaf Gohar<br>24280009 | ● Developed the Rulebook for RAG-based evaluation<br>● Implemented transcription using WhisperX and Diarization<br>● Analyzed real call center recordings for system testing<br>● Manually segregated sentiments<br>● Ensured accurate sentiment and tone annotation | 100%<br>20%<br>80%<br>80%<br>30% |
| M. Arslan Rafique<br>24280064 | ● Tested out various transcription and diarization techniques.<br>● Tested out different LLM models and then acquired good output via Prompt tuning.<br>● Defined Rulebook and Rag implementation | 80%<br>50%<br>30% |
| M. Annus Shabbir<br>24280015 | ● Implemented Transcription, Diarization<br>● Speaker Classification<br>● Code Integration of all the separate modules (pipeline making)<br>● Tried Deployment & Dockerization<br>● Presentation<br>● Report | 80 %<br>80%<br>100 %<br>70 %<br>20%<br>10% |
| Talha Nasir<br>24280040 | ● RAG implementation<br>● Sentiment analysis<br>● Tonal analysis | 80%<br>20%<br>20% |
| Eeman Adnan<br>24280022 | ● Sentiment analysis for customer and agent separately and overall on entire call<br>● Tonal analysis for customer and agent separately and overall on entire call<br>● Optimized these for higher accuracy<br>● Presentation<br>● Dockerization<br>● Report | 80%<br>80%<br>100%<br>60%<br>10%<br>90% |