MemoTag: Speech Intelligence Module for Early Cognitive Impairment Detection

Project Objective

This module is designed to analyze voice samples and identify early signs of cognitive impairment by leveraging speech patterns, linguistic markers, and acoustic features. It integrates advanced audio processing, natural language processing (NLP), and machine learning to deliver an end-to-end pipeline that can process anonymized voice clips, extract cognitive health indicators, and flag samples that deviate from healthy baselines.

System Overview

Input

- Audio file (e.g., `.wav` format)

Output

A JSON response with:

- Transcription of audio
- Language detection
- Sentiment and emotion classification
- Cognitive-linguistic feature analysis
- Acoustic feature analysis
- Word recall inconsistency detection
- Text embedding vector for unsupervised ML

API Endpoint

`POST /analyze-cognition/`

Testable via

Postman or Python HTTP client

Architecture and Components

Component	Technology
ASR (Transcription)	OpenAI Whisper
Language Detection	`langdetect`
Sentiment Analysis	HuggingFace Transformers
Emotion Classification	`j-hartmann/emotion` model
Keyword Extraction	YAKE
Audio Analysis	Librosa
Embedding Model	SentenceTransformers
Server Framework	FastAPI + Uvicorn

Feature Engineering for Cognitive Analysis

This module extracts the following features:

1. Linguistic and Acoustic Markers

Feature	Description
Speech Rate	Words per second, based on audio duration
Pauses per Sentence	Count of silence markers per sentence
Hesitation Count	Occurrences of filler words ("uh", "um", "er", etc.)
Pitch Variability	Standard deviation of pitch estimated using YIN algorithm
Word Count	Total words spoken
Sentence Count	Count of complete sentences

2. Word Recall Consistency

- **Important keywords** extracted using YAKE.
- **Noun extraction** using POS tagging.
- **Recall gap**: Key concepts present in speech but missing from noun set.

3. Semantic Embedding

- Each transcript is encoded using `all-MiniLM-L6-v2` to generate a 384-dimensional vector.
- These embeddings are used for unsupervised ML methods (e.g., clustering, anomaly detection).

Machine Learning Approach

Unsupervised Anomaly Detection

- A collection of text embeddings is clustered using **KMeans** to group similar cognitive profiles.
- Outliers can be interpreted as potential impairment cases based on divergence in semantic expression.

Similarity Scoring (optional)

- Cosine similarity between known healthy and at-risk transcript embeddings can serve as a screening threshold.

Model Interpretability

- All features are domain-relevant and explainable to clinicians.
- Emphasis on feature-level insights (e.g., speech rate, hesitation count) rather than opaque model outputs.

- Embeddings and clustering results can be visualized using t-SNE or PCA for transparency.

Testing and Evaluation Workflow

1. Upload `.wav` file using Postman to `http://localhost:8000/analyze-cognition/`
2. Receive structured JSON output.
3. Aggregate output from multiple users for comparative analysis.
4. Visualize embedding clusters and validate flagged anomalies manually or via basic heuristics.

Example Output

json
CopyEdit


```json
{
  "transcription": "I was going to the uh, the place, but I forgot the name.",
  "language": "en",
  "sentiment": [{"label": "NEUTRAL", "score": 0.89}],
  "emotion": [{"label": "confusion", "score": 0.76}],
  "duration_sec": 6.3,
  "cognitive_features": {
    "num_sentences": 1,
    "num_words": 13,
    "speech_rate_wps": 2.06,
    "pauses_per_sentence": 3.0,
    "hesitation_count": 2,
    "pitch_variability": 18.42
  },
  "recall_issues": {
    "important_keywords": ["place", "name"],
    "missing_keywords": ["name"]
  },
```

```
  "text_embedding": [0.043, 0.117, ..., 0.071]
}
```

Clinical Relevance

The extracted markers correlate with linguistic and acoustic symptoms observed in early stages of:

- Mild Cognitive Impairment (MCI)
- Alzheimer’s Disease
- Other neurodegenerative disorders

This module serves as a digital pre-screening assistant and can be used in telehealth or home-monitoring applications.

Next Steps for Production Deployment

- Integrate speaker diarization and timestamped pause detection.
- Normalize features across age, gender, and native language.
- Extend to longitudinal tracking across multiple sessions.
- Validate with clinician-labeled datasets and refine feature thresholds.
- Containerize using Docker for deployment on GCP/AWS.

Deliverables

- Jupyter Notebook containing full codebase
- FastAPI endpoint hosted locally (or optionally on cloud)
- Test suite using Postman

- Sample output files
- Feature extraction and ML logic
- This report for review and technical evaluation