**SPEAKER DIARIZATION RESEARCH**

Readily available solutions for speaker diarization

**Audio-Based Tonal Level**

1. **Google Cloud Speech-to-Text API:**

* Google Cloud Speech-to-Text API does include speaker diarization capabilities. It can recognize and differentiate multiple speakers allowing to identify “who spoke when” in an audio segment. The API provides speaker labels, indicating which portions of the audio were spoken by each identified speaker. These labels can be used to create a transcript that specifies different speakers.
* The API can be easily integrated into various applications and services through API calls, making it accessible and convenient for developers.
* <https://cloud.google.com/speech-to-text/docs/samples/speech-transcribe-diarization>

1. **Pyannote-Audio:**

* Pyannote-Audio is an open-source Python library designed for speaker diarization, speaker embedding, speaker change detection. It is built on top of Pyannote Core which provides metrics computation in speaker diarization.
* It achieves speaker diarization through various deep-learning approaches including clustering methods.
* It supports different types of audio data and features like MFCC(Mel-Frequency Cepstral Coefficient).
* <https://pyannote.github.io/pyannote-audio/>

1. **OpenAI Whisper:**

* It is open-source (Automatic Speech Recognition) trained model for detecting speech automatically which is trained on 680,000 hours of multilingual and multitask supervised data.
* It enables transcription in multiple languages, as well as translation from those languages into English.
* It supports different types of audio data and features like MFCC(Mel-Frequency Cepstral Coefficient).
* <https://openai.com/research/whisper>

1. **Deepgram:**

* Deepgram offers a speaker diarization API that can separate individual speakers in an audio stream and bucket their utterances together based on their unique audio characteristics.
* It is broken down into four major subtasks:
* Detection - Find regions of audio that contain speech as opposed to silence or noise.
* Segmentation - Divide and separate those detected regions into smaller audio sections.
* Representation - Use a discriminative vector to represent those segments.
* Attribution - Add a speaker label to each segment based on its discriminative representation.
* <https://deepgram.com/learn/what-is-speaker-diarization>

1. **AssemblyAI:**

* AssemblyAI provides a speaker diarization API that can identify who spoke when in an audio file and label each speaker’s utterances.
* They also provide sample code snippet for transcribe an audio file and Enable Speaker Diarization to detect speakers in an audio file.
* <https://www.assemblyai.com/docs/getting-started/transcribe-an-audio-file>

1. **Kaldi:**

* Kaldi is an open-source toolkit for speech recognition that includes tools for speaker diarization. It provides a range of tools for speech recognition tasks.
* The speaker diarization module in Kaldi uses a combination of Gaussian Mixture Models (GMMs) and i-vectors to perform speaker diarization.
* <https://www.kaldi-asr.org/doc/about.html>

**NLP-Based Level**

1. **LIUM SpkDiarization:**

* LIUM SpkDiarization is an open-source speaker diarization tool based on Gaussian Mixture Model (GMM) and Hidden Markov Models (HMM). It is useful for diarization tasks in both broadcast news and telephone conversations.
* It is designed to separate speakers in an audio recording in various applications such as transcription services, audio indexing, and speaker recognition systems.
* It include MFCC computation, speech/non-speech detection, and speaker diarization methods.
* <https://projets-lium.univ-lemans.fr/spkdiarization/>

**Integration of Audio and NLP Approaches**

1. **pyAudioAnalysis:**

* This Python library combines audio analysis and NLP techniques for various tasks, including speaker diarization. It provides modules for audio feature extraction and machine learning-based classification. By integrating audio features with NLP methods, it offers a comprehensive solution for speaker diarization.
* The library offers a variety of audio analysis functionalities, including feature extraction (e.g. MFCC, chroma feature), classification (e.g. SVM, k-NN), and segmentation. These capabilities make it useful for music genre classification, speaker recognition, and emotion recognition.
* <https://github.com/tyiannak/pyAudioAnalysis>

**Speaker Diarization Solutions: A Comparative Analysis (100,000 Minutes of Data)**

**Google Cloud Speech-to-Text API**

**Pros:**

* High accuracy due to advanced machine learning models.
* Robust speaker diarization capabilities.
* Scalable for large volumes of data.
* Reliable technical support.

**Cons:**

* Costly for extensive data volumes.
* Limited customization options for lower-tier plans.

**Projected Accuracy: High**

**Projected Costs: 60 Minutes/Month – Free, Over 60 Minutes/Month - $0.024 / minute.**

**Pyannote-Audio**

**Pros:**

* Its versatility makes it useful for a wide range of applications in speaker analysis.
* It provides a unified framework for managing annotations and evaluating diarization results.
* It enables users to assess the accuracy of their diarization algorithms and compare different approaches.

**Cons:**

* It might be resource-intensive in terms of memory and processing power, especially for large-scale applications.
* It can be slow when processing long audio files.

**Projected Accuracy: High**

**Projected Costs: open-source.**

**OpenAI Whisper**

**Pros:**

* It is robust and able to perform multilingual speech recognition and speech translation.
* It has a simple end-to-end architecture, implemented as an encoder-decoder Transformer, which makes it easy to use and modify.

**Cons:**

* It does not provide any speaker tracking.
* Whisper may not be able to transcribe noisy, distorted and low quality audio files.

**Projected Accuracy: High**

**Projected Costs: open-source and For High Level Implementation it supports Large-V2 model - $0.006/minute.**

**Deepgram**

**Pros:**

* Deepgram provides real-time transcription capabilities.
* Deepgram utilizes deep learning algorithms, which are capable of capturing complex patterns in audio data.
* Deepgram allows for customization of its models.

**Cons:**

* Like all machine learning-based systems, Deepgram's accuracy heavily depends on the quality and diversity of the training data.
* Privacy and Security Concerns.

**Projected Accuracy: High**

**Projected Costs: $200 free credit for individuals, $4K/year for teams building their voice apps.**

**AssemblyAI:**

**Pros:**

* AssemblyAI provides a user-friendly API and SDKs, making it easy for developers to integrate their services into various applications and platforms.
* AssemblyAI allows users to upload custom vocabulary lists, enabling better recognition and enhancing accuracy for specialized domains.

**Cons:**

* While AssemblyAI supports multiple languages, the range might be more limited compared to some other ASR services, particularly for less common languages and dialects.
* Limited Offline Capabilities.

**Projected Accuracy: High**

**Projected Costs: Core Transcription - $0.650016 per hour and Real Time Transcription - $0.75024 per hour**

**Kaldi:**

**Pros:**

* Kaldi is open-source software, allowing users to access, modify, and customize the toolkit according to their specific needs.
* Kaldi integrates state-of-the-art algorithms for various speech processing tasks, including feature extraction, acoustic modeling, and language modeling.
* Kaldi is capable of handling large volumes of data and can be scaled for processing extensive datasets, making it suitable for tasks involving substantial amounts of audio data.

**Cons:**

* Training complex models in Kaldi requires significant computational resources, including powerful GPUs.
* Configuring and tuning various components of the toolkit can be complex and time-consuming.

**Projected Accuracy: High**

**Projected Costs: open-source**

**LIUM SpkDiarization:**

**Pros:**

* As an open-source project, LIUM SpkDiarization benefits from contributions and support from a community of developers and researchers. Users can find help, share knowledge, and collaborate on improvements.
* Since it's open-source, LIUM SpkDiarization is cost-effective and budget-friendly, making it accessible to individuals, researchers, and small organizations with limited financial resources.
* Users have the flexibility to customize and fine-tune the diarization process based on their specific use cases.

**Cons:**

* LIUM SpkDiarization might face challenges with very large volumes of data.
* Scalability can be a limitation for projects with extensive audio recordings.

**Projected Accuracy: High**

**Projected Costs: open-source**