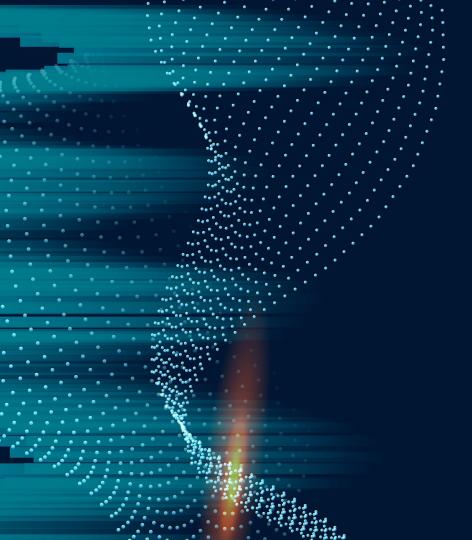
Kaldi ASR Modeling:

Robust Automatic Speech Recognition using. Kaldi for Radiotelephonic Applications



Dr. Jianhua Liu & Andrew Schneider

Product Owners

Tabitha O'Malley

Team Lead, Research, Documentation

Milan Haruyama

Coding, Research, Documentation

David Serfaty

Coding, Research, Documentation

Tamina Tisha

Research, Documentation

Adam Gallub

Coding, Documentation

What?

Develop acoustic ASR models capable of transcribing live ATC transmissions in real-time using the Kaldi ASR Toolkit and the ATC02 corpus.

Who?

- Ab initio pilots
- Flight Instructors
- Air Traffic Control Operators
- Hobbyists

Why?

- Miscommunications between ATC and Pilots due to a lack of formal training for Pilots
- Ability to review communications on the ground after flights for training purposes

End to End vs Acoustic Models

End-to-End (E2E) Models

- Relationship between audio signals and the words
- Large dataset is required
- Larger and more complex models.

Acoustic Models

- Relationships between audio signals and the phonetics
- Better results with small dataset
- Smaller models

Design Constraints (Training)

- Minimum storage size of 12.5 GB
- Minimum video memory (VRAM) of 12 GB
 - NVidia GeForce RTX 4080 (Training)
- AMD processor (recommended)

Design Constraints (Transcribing)

- All inputs as WAV files
 - Convert non-WAV files using FFMPEG
- General American English
 - E.g., "color" versus "colour"
- No punctuation or grammatical marks
 - Commas, colons, hyphens, etc.

Assumptions and Dependencies

- Clear and direct communication
 - Low interference and background noise
- General American English
- 20- to 100-hour dataset
- Sufficient storage
- Sufficient video memory
- Operating system
 - Linux (e.g., Ubuntu, Debian, etc.)
 - Windows Subsystem for Linux (WSL)
 - Virtual machine for MacOS

System Architecture (Preprocessing)

ATCO Corpus

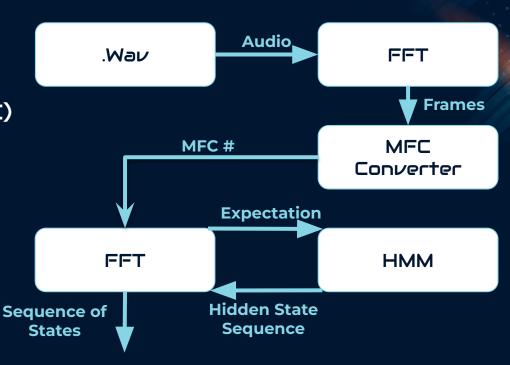
- Linguistic Data Consortium (LDC)
- 30-hour ATC dataset
 - audio files
 - text transcriptions

ATCO2 Corpus

- Repository for ASR and NLP research
- Provides preprocessing script written in Bash designed for Kaldi ASR Toolkit

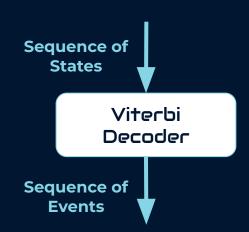
System Architecture (Preparation)

- Setting up audio for decoding
- Fast Fourier Transform (FFT)
- Mel-Frequency Cepstrum (MFC)
 - Mel-Frequency Cepstral Coefficients (MFCCs)
- Gaussian Mixture Model (GMM)
- Hidden Markov Model (HMM)



System Architecture (Preparation)

- Using hidden state
 sequence to find the most
 probable sequence of
 phonemes
- Viterbi decoder

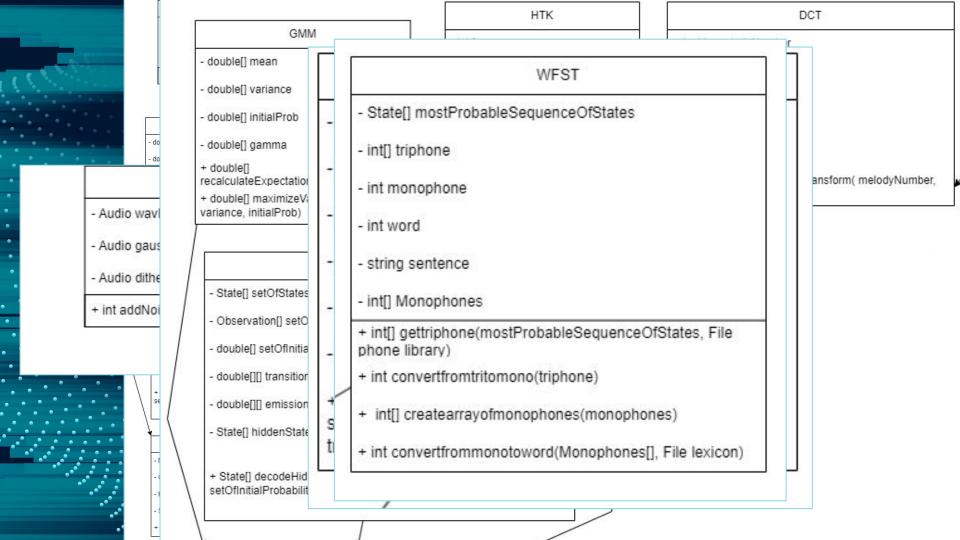


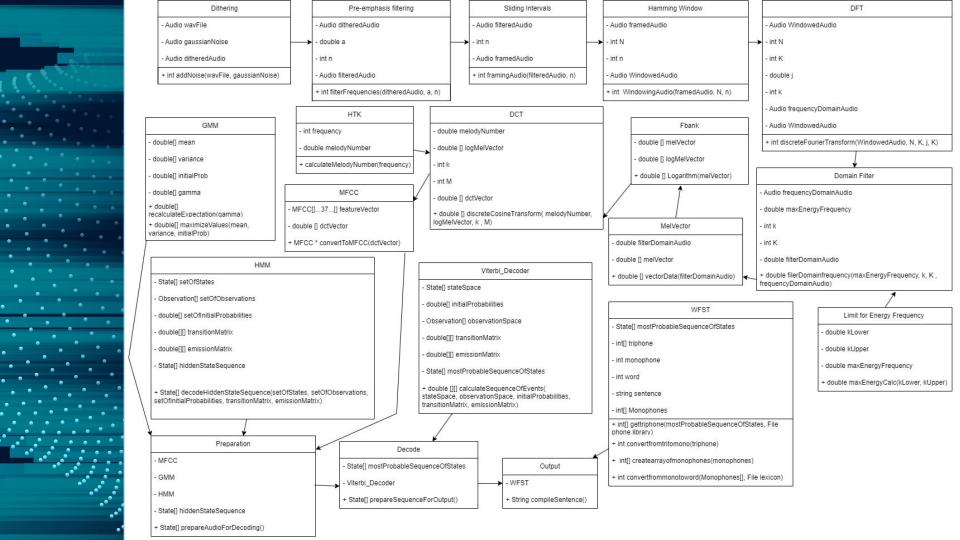
System Architecture (Output) Sequence

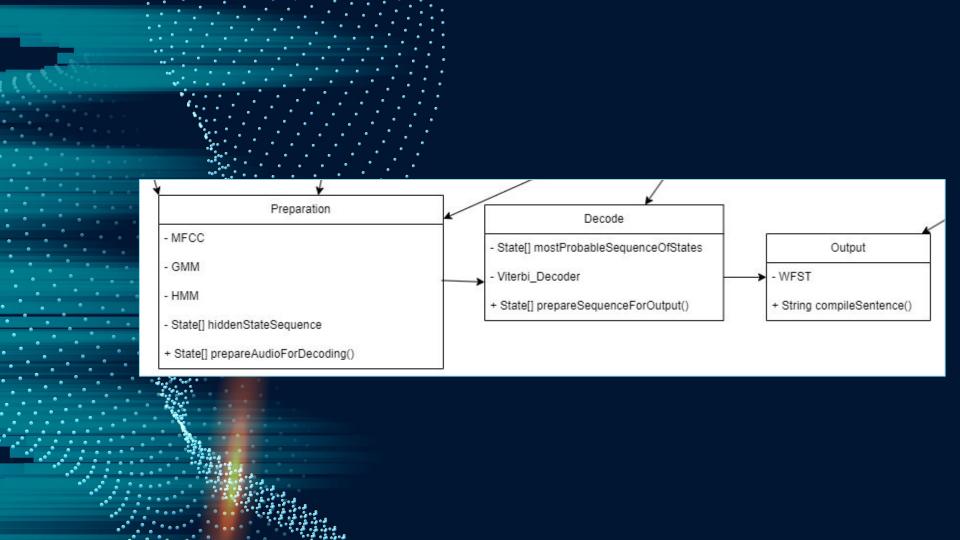
- Converting to sentences
- Weighted Finite State
 Transducers (WFSTs)
- HMM
- Context Dependence
- Lexicon
- Language Model

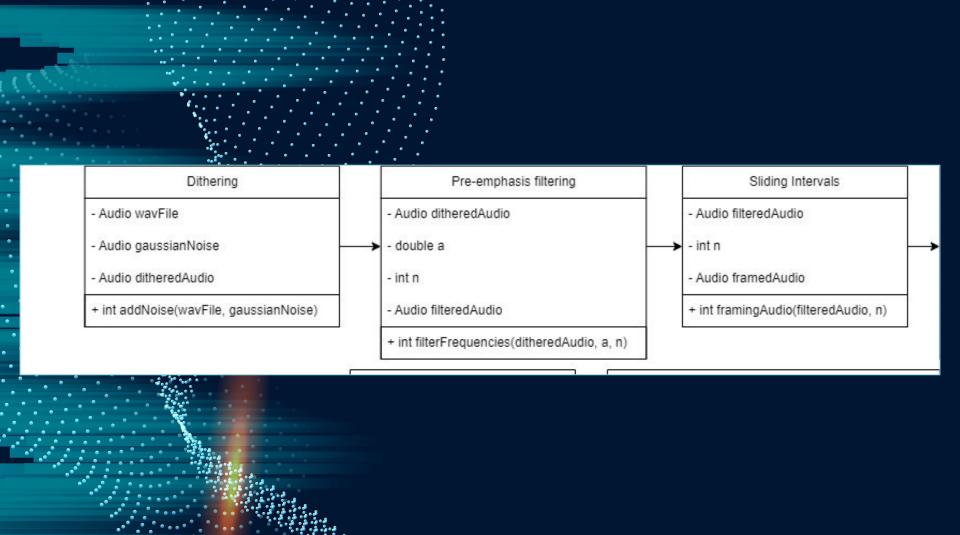


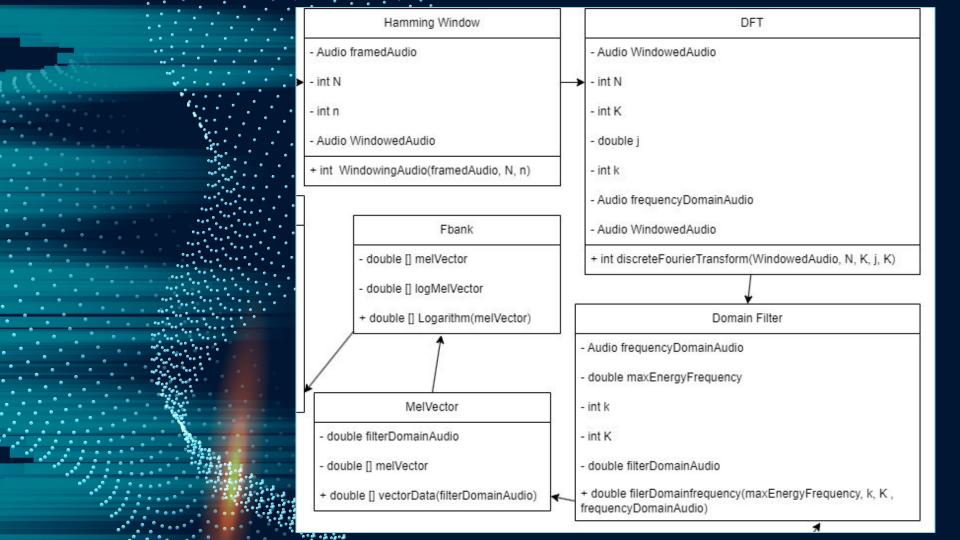
Out.txt

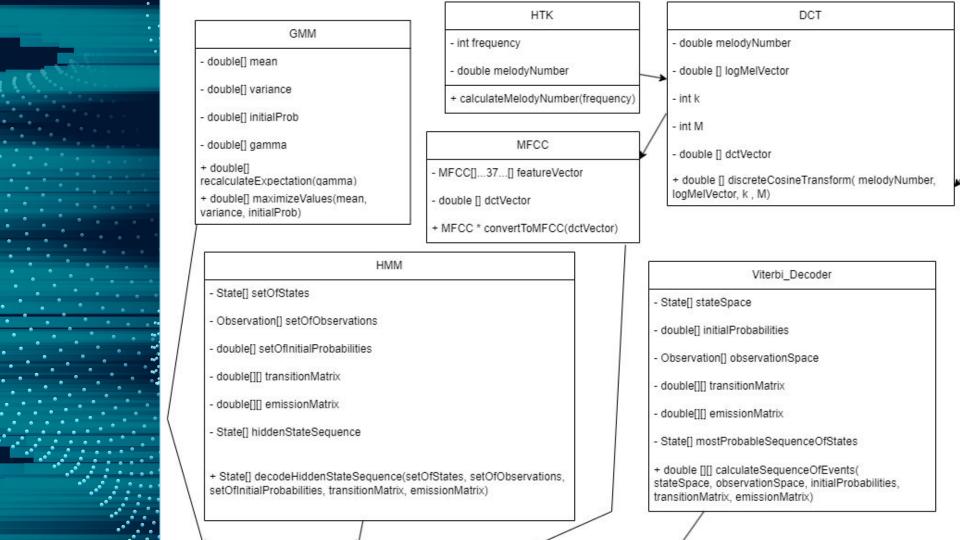








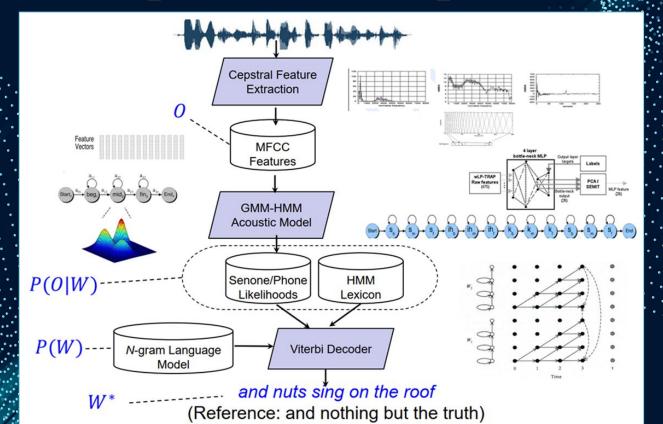




Viterbi_Decoder State[] stateSpace double[] initialProbabilities Observation[] observationSpace double[][] transitionMatrix double[][] emissionMatrix State[] mostProbableSequenceOfStates + double [][] calculateSequenceOfEvents(stateSpace, observationSpace, initialProbabilities, transitionMatrix, emissionMatrix)

	WFST
	- State[] mostProbableSequenceOfStates
	- int[] triphone
	- int monophone
	- int word
	- string sentence
	- int[] Monophones
	+ int[] gettriphone(mostProbableSequenceOfStates, File phone library)
	+ int convertfromtritomono(triphone)
	+ int[] createarrayofmonophones(monophones)
	+ int convertfrommonotoword(Monophones[], File lexicon)

Sub-System Design



Iteration Iteration Iteration Iteration 7.0.0 13.1.0 17.0.5 17.1.0 **WER = 18.06% WER = 13.50% WER = 13.21% WER = 13.23**% **SER = 78.88% SER = 72.36% SER = 71.26% SER = 72.08%**

Demo

