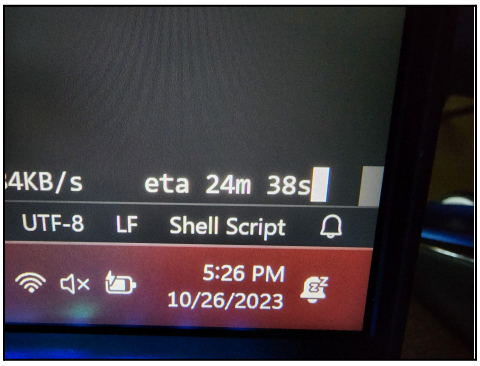
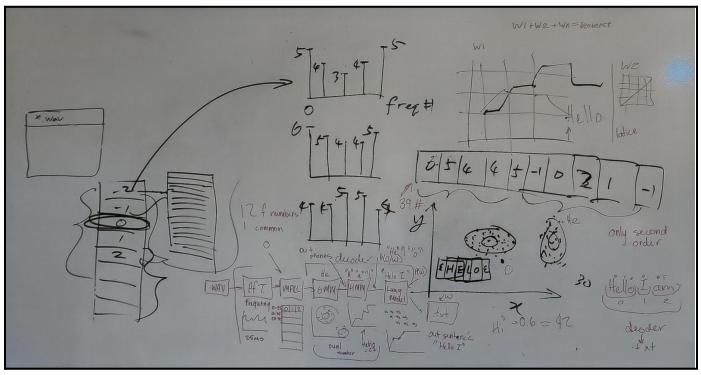
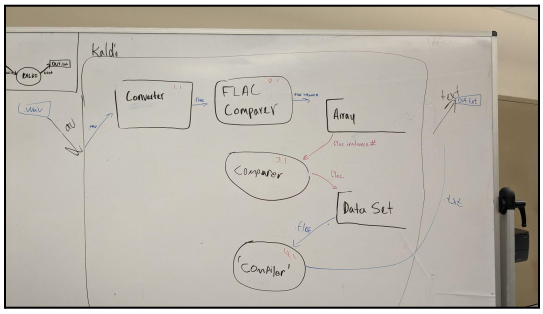
Milan Haruyama
CS 490
Dr. Rumia Sultana & Dr. M. Ilhan Akbas
07 December 2023

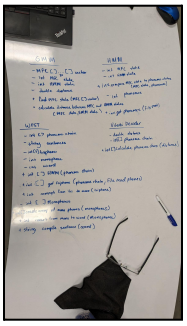
Engineering Notebook

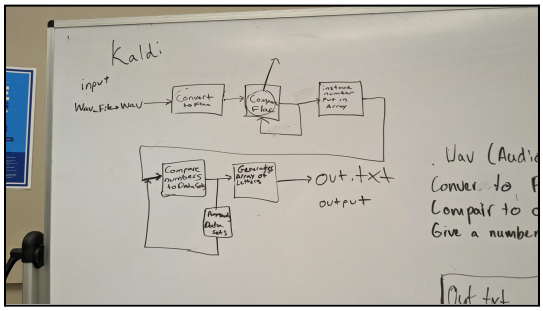
Photos throughout the semester:

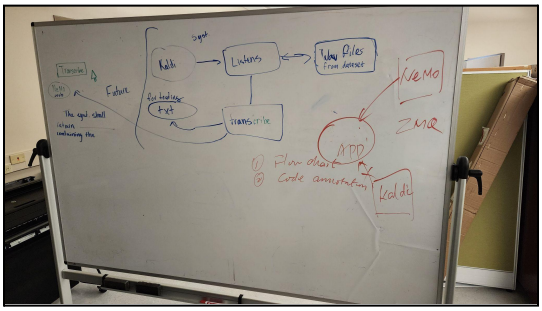


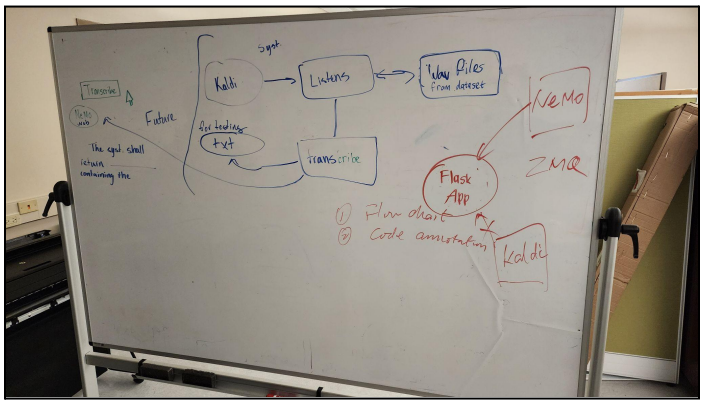


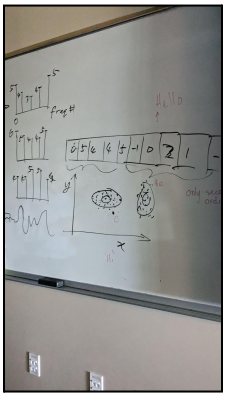


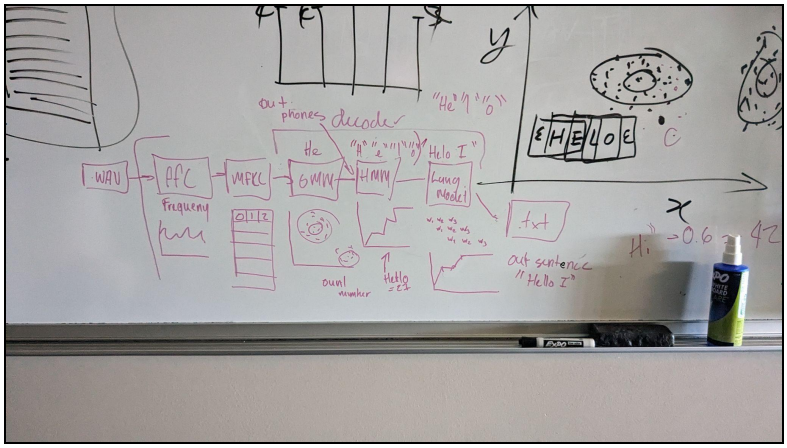


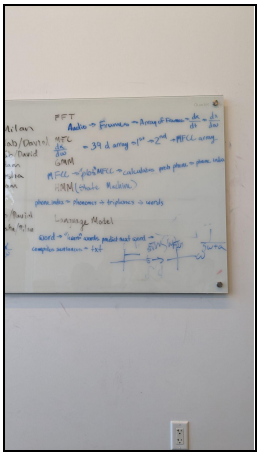








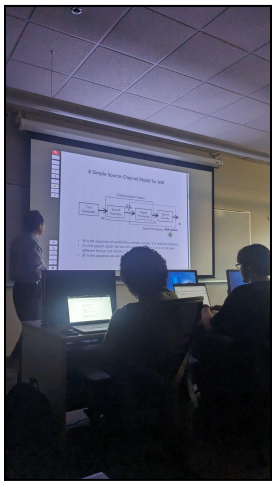


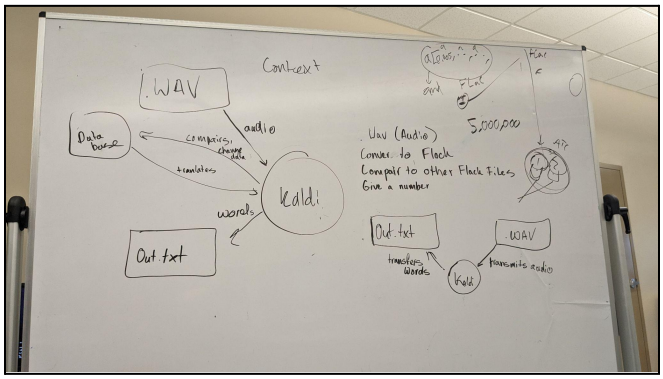


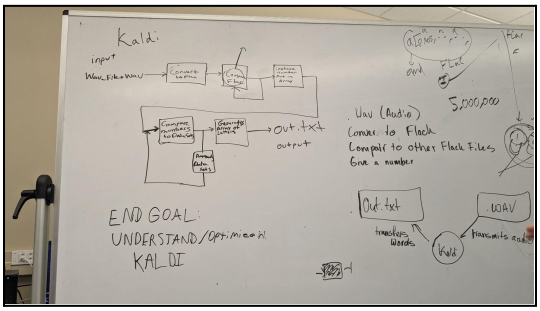
FFT

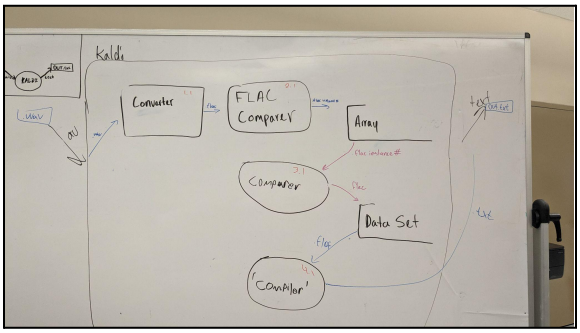
- Audio audioFile
- Frame frame
- Frame[] frames

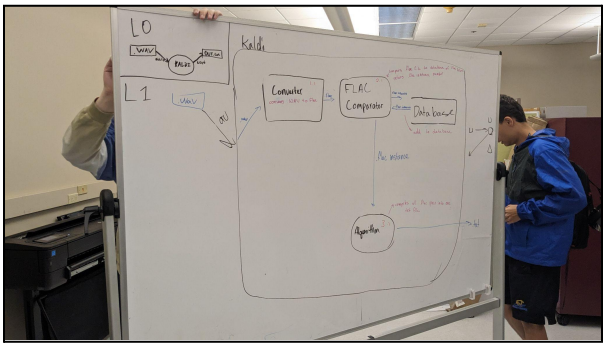
+ Frame cvt2frm(Audio)
+ Frame[] frameArr(Frame)
+ Frame cvt2freq(Frame[])

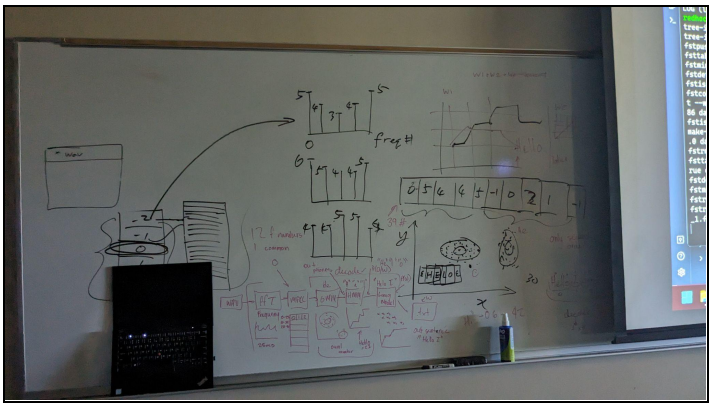


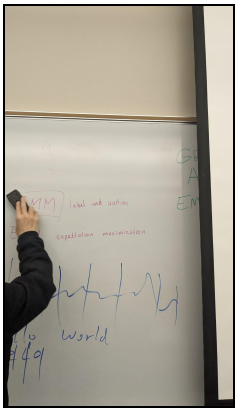


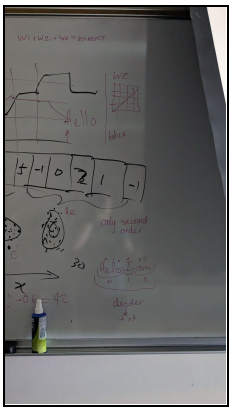


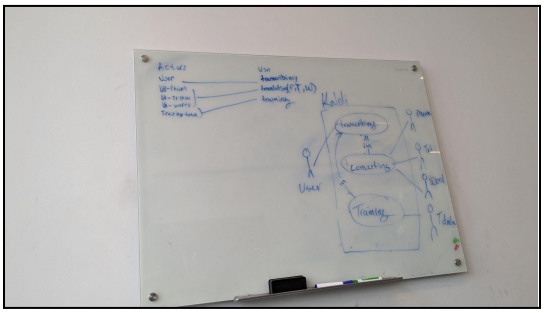


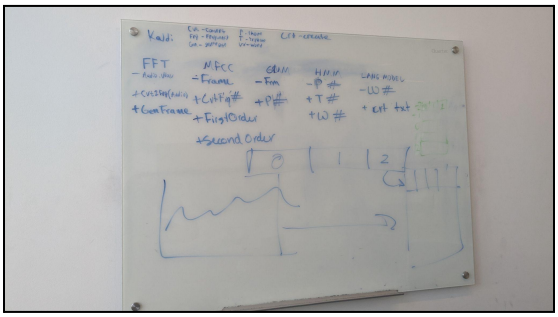


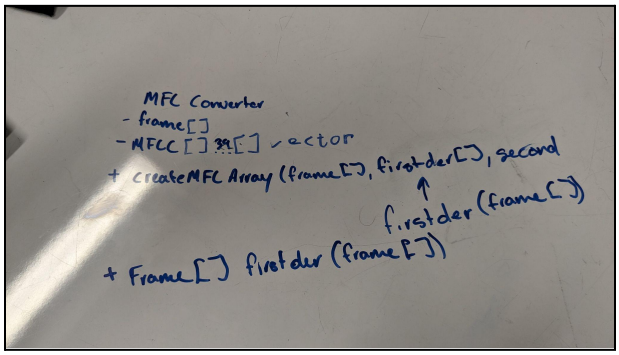


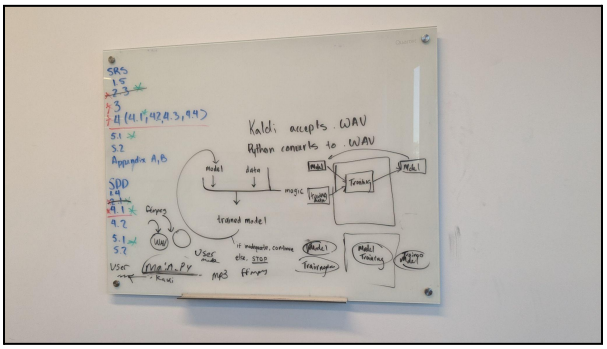


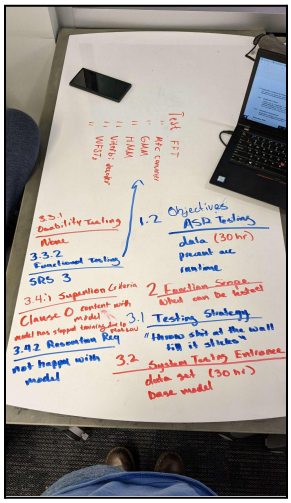


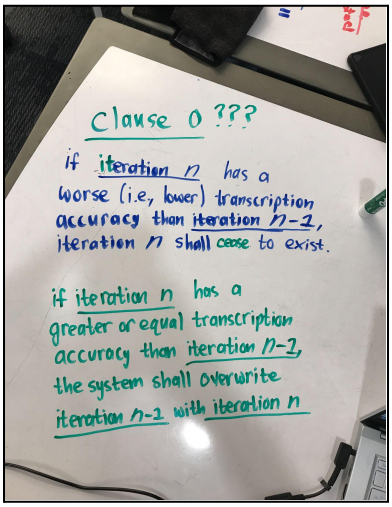


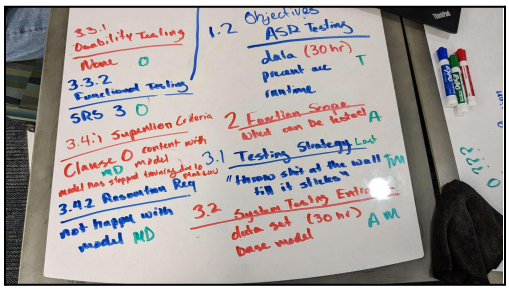


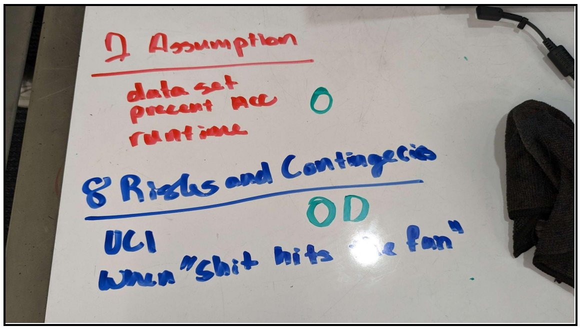

MFC Converter
- frame[]
- MFCC[][] ~actor
+ createMFC Array (frame[], firstder[], second
firstder (frame[])
+ Frame[] firstder (frame[])


Kaldi accepts .WAV
Python converts to .WAV


3.5.1 Durability Testing
None
3.3.2 Functional Testing
SRS 3
3.4.1 Suspension Criteria
Clause 0 content with
3.4.2 Resumption Req
not happy with
1.2 Objectives
ASD Testing
data (30 hr)
percent acc
runtime
2 Function Scope
What can be tested
3.1 Testing Strategy
"throw shit at the wall
till it sticks"
3.2 System Testing Entrance
data set (30 hr)
base model


Clause 0 ???
if iteration n has a
worse (i.e, lower) transcription
accuracy than iteration n-1,
iteration n shall cease to exist.
if iteration n has a
greater or equal transcription
accuracy than iteration n-1,
the system shall overwrite
iteration n-1 with iteration n


3.5.1 Durability Testing
None
3.3.2 Functional Testing
SRS 3
3.4.1 Suspension Criteria
Clause 0 content with
3.4.2 Resumption Req
not happy with
1.2 Objectives
ASD Testing
data (30 hr)
percent acc
runtime
2 Function Scope
What can be tested
3.1 Testing Strategy
"throw shit at the wall
till it sticks"
3.2 System Testing Entrance
data set (30 hr)
base model


7 Assumption
data set
precent acc
runtime
O
8 Risks and Contingecis
OD
UCI
When "shit hits

SRS Revision History:

Name	Date	Reason For Changes	Version
Tabitha, Milan, Tisha, David, Adam, Max	09/29/23	Starting the document	V1.0
Milan	10/27/23	Writing and Editing Requirements: 3.1	V2.4
Milan	10/30/23	Editing all Sections Writing Section: 5.1, 5.2, 2.1	V2.9
Milan	10/31/23	Editing all sections	V2.12
Milan	11/05/23	Editing all sections	V3.2
Milan	11/07/2023	Adding/Editing: 3	V3.7
Milan	11/11/2023	Editing: 3	V3.11
Tisha	11/18/2023	Rewriting section 5.2	V3.18
Tisha	11/18/2023	Edited section 5.2	V3.19
Milan	11/18/2023	Editing all sections	V3.20
Milan	11/19/2023	Editing all sections	V3.23
Milan	11/20/2023	Reviewing/editing all sections	V3.24

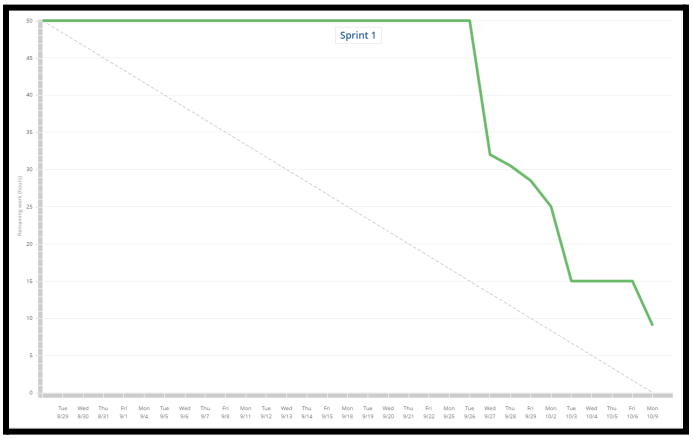
SDD Revision History:

Name	Date	Reasons For Change	Version
Tabitha, Milan, David, Max, Tisha, Adam	09/29/2023		V1.0
Tisha, Tabitha, Milan	10/24/2023	Rewriting the section: 2.2	V2.2
Tabitha, Tisha, Milan	10/24/2023	Writing the section, Rewriting, and editing: 1.2	V2.3
Milan	10/26/2023	Writing Sections: 1.1, 1.2, 1.5	V2.7
Milan	10/29/2023	Editing All Sections	V2.13
Milan	10/30/2023	Editing All Sections	V2.14
Milan	10/31/2023	Editing all sections	V2.18
Milan	11/07/2023	Editing all Sections, reformatting Table of Contents	V3.6
Milan	11/11/2023	Editing all sections	V3.10
Milan	11/11/2023	Editing: 2.2	V3.11
Milan	11/18/2023	Editing all sections	V3.19
Milan	11/19/2023	Editing all sections	V3.22
Milan	11/20/2023	Reviewing/editing all sections	V3.23

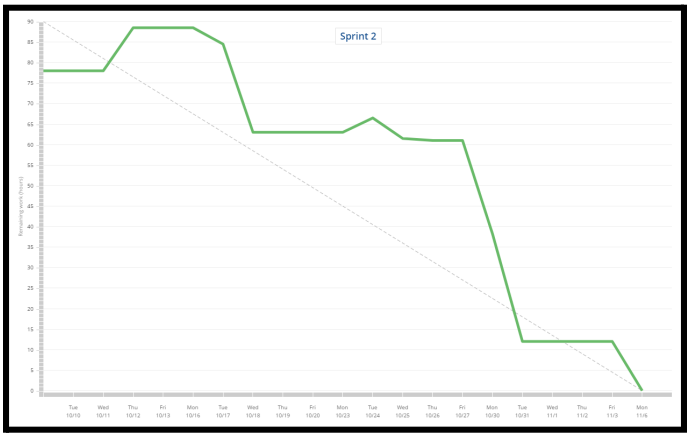
System Test Plan Revision History:

Name	Date	Reason For Changes	Version
All	10/18/2023	Write wrong information	V1.0
Milan	11/27/2023	Section 3.4; Editing all sections	V2.4
Milan	11/28/2023	Editing all sections	V2.9
Milan	12/02/2023	Editing all sections	V2.12

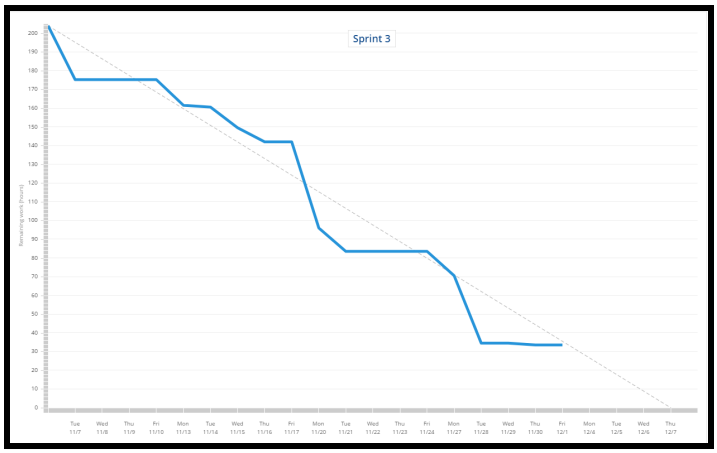
Sprint 1 Burndown Chart:



Sprint 2 Burndown Chart:



Sprint 3 Burndown Chart:



Notes taken throughout the semester:

21 September 2023

Diversity of speech creates problem with speech recognition

- Different length
- Different pauses
- Different intonation
- Different pitch
- Different stress

Background noise and channel responses also create problems with speech recognition.

W is the sequence of words from a certain speaker; aka utterance
X is the speech signal in the frequency domain
O (a vector made up of 39 frequency components) is extracted from X in the frequency domain
Kaldi performs all the complex math operations (thank god lol)

GMM & HMM

- Gaussian Mixture Model (GMM) for approximating complex probability density functions (PDFS)
    - E.g., weight of cats (domestic, big cats, etc.)
    - Conditional probability (given previous text, the probability of next)
        - E.g., P("Happy" | "I", "am"): Given the text "I" and "am", what is the probability that the next text is "happy"?
- Markov Chain:
    - P(s2 | s1, s0) = P(s2 | s1); implies s0 is irrelevant; only the immediately preceding word is relevant.
- Hidden Markov Model (HMM)
    - System described by states
    - State cannot be observed
    - State transitions in a no-backward manner
    - Each state transmits observable RV
    - Use observable RV to infer the state of the HMM
- Each phone has their own set of states
    - E.g., "Hello"
        - "H" has its own set of states
        - starting from either silence/end of previous word/sound
        - the start of the phone
        - the middle of the phone
        - end of the phone
        - transition to next phone

Phones, Triphones, etc.

10 October 2023

How Kaldi sample ASR model works

1. Input .wav (.wav OR .FLAC converted to .wav through ffmpeg)
2. .wav is split into 25ms frames every 10ms [0-25, 10-35, 20-45, n-n+25] (the frames overlap to assist in finding the beginning and end of words and to eliminate noise)
3. Frame is put into FFT to convert to frequency graph
4. MFC lines up frames into an array and compares the current frame with the frames before and after it by 2 orders [-2|-1|0|1|2].
5. Each element of the array is put into the GMM to determine the most likely phone, and returns a numerical value equivalent to one of the phones in the phone lexicon
6. The numbers from the GMM are then input into the HMM to assemble triphones out of the phones
7. The triphones are then input into another HMM to become the most likely possible word by comparing the result of the HMM to the word library
8. The word returned from the HMM is input into the language model and using the two previous words the model tries to predict the next most likely word [-2|-1|0]
9. The HMM outputs the converted sentence to the out.txt file