System Design Document

For

RTube Kaldi Team

Team member: Tabitha O'Malley, Milan Haruyama, David Serfaty, Maxwell Moolchan, Tahmina Tisha, Adam Gallub

Version/Author	Date
V1/All	09/29/23

TABLE OF CONTENT

SYSTEM DESIGN DOCUMENT

Overview
The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

1 INTRODUCTION

1.1 Purpose and Scope

This document describes the system requirements, operating environment, system architecture, inputs and outputs, human-machine interface, detailed design, external interfaces, and system integrity controls for the Kaldi portion of the RTube project.

Commented [1]: same question as below

~~This section provides a brief description of the Systems Design Document's purpose and scope.~~

1.2 Project Executive Summary

This section provides an overview of the Kaldi portion of the RTube project from a management perspective, providing a view of the framework which the system design originates from.

~~This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.~~

1.2.1 System Overview

This section describes the system in narrative form using non-technical terms. It provides a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams show interfaces to external systems, if applicable. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements traceability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

Narrative Description:
High-Level System Architecture Diagram:
Subsystem Breakout:
Interfaces to External Systems:
High-Level Context Diagram:
Requirements Traceability Matrix (RTM):

Our project builds upon the foundation laid by previous teams who employed Kaldi to develop ASR models tailored for ATC applications. These predecessors generously shared their outcomes under an open-source license, fostering an environment of collaboration and innovation. Our team's mission is to extend this groundwork by creating novel variations of the existing ASR models, leveraging our

custom ATC dataset. The primary objective of our project is twofold. First, we aim to develop and assess these new permutations of ASR models meticulously. Second, we intend to employ the most suitable models from this set to construct an online speech recognition system capable of transcribing live ATC communications. By the culmination of this semester, our goal is to deliver a fully functional system that provides comprehensive transcriptions of live speech input originating from both aircraft and ATC towers. This system shall significantly enhance the efficiency and accuracy of ATC operations, ensuring that crucial communications are readily and precisely transcribed for analysis and decision-making. In essence, our project builds upon the collaborative efforts of previous teams, harnessing state-of-the-art ASR technology to advance the field of ATC communication by creating a robust and real-time transcription system.

1.2.2 Design Constraints

Speech recognition encounters numerous challenges, often characterized by the inherent limitations of our current technology. To begin, speech exhibits significant variability among speakers, encompassing differences in speed, accent, intonation, and pitch, thereby presenting formidable obstacles to achieving precise transcription. Furthermore, linguistic diversity means that individuals may articulate identical concepts using disparate vocabulary and phrasing, complicating the process of converting speech to text—particularly when multiple valid transcriptions exist. Another formidable limitation arises from the presence of background noise in recording environments, which can severely impede recognition accuracy, demanding a robust response adaptable to various noise levels. The utilization of different recording channels introduces further distortions, adding an additional layer of complexity to the recognition process.

While Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) constitute potent tools within the field of speech recognition, their limitations become apparent in the face of intricate speech patterns and underlying assumptions that may not consistently hold true in practice. The collection of ample labeled training data, an essential component of model development, presents a formidable challenge due to the substantial time and resources required, especially considering the vast array of speech patterns and accents. Additionally, the employment of triphones—combinations of three phonemes—holds the potential to enhance modeling, albeit at the expense of increased model complexity, computational demands, and data requirements. Coping with out-of-vocabulary words and phrases represents a pervasive limitation, as speech recognition systems often struggle to transcribe content that does not align with their predefined lexicons. The attainment of low-latency, real-time processing, while upholding accuracy, stands as a pivotal challenge, particularly in applications such as voice assistants. Furthermore, speech recognition predominantly focuses on the textual content of speech, neglecting the vital realm of non-verbal cues, encompassing aspects like facial expressions and gestures that bear significant relevance in understanding user intent. Implementing personalized speaker adaptation, where systems adapt to individual speakers, is a complex undertaking riddled with challenges. Lastly, contextual understanding remains a substantial limitation, as deciphering context and resolving word ambiguities within speech remains a multifaceted and occasionally error-prone endeavor for speech recognition systems.

1.2.3 Future Contingencies

In the development of our speech recognition system, we anticipate several potential contingencies that could influence our project's direction. These contingencies encompass challenges such as the lack of interface agreements with external agencies, the possibility of unstable architectural elements, concerns regarding data availability and quality, evolving regulatory landscapes, resource constraints, personnel changes, security vulnerabilities, shifting market and user demands, technological

advancements, and unexpected events like natural disasters. To address these contingencies, we are committed to maintaining flexibility in our architecture, staying informed about industry developments, establishing backup plans, and continually monitoring and adapting to the changing landscape. Our proactive approach includes risk assessments, contingency planning, and diligent monitoring, ensuring that our system remains robust and adaptable to the dynamic factors that may arise during its development and deployment.

1.3 Document Organization

This document is designed to provide an idea of the system design to the reader. The following sections provide information on the operations of the product, limitations, interactions, interfaces, software and hardware designs, and security.

~~This section describes the organization of the Systems Design Document.~~

1.4 Project References

1.4.1 Speech Recognition for Air Traffic Control (pdf), from Professor Jianhua Liu
1.4.2 Principle of Kaldi Based ASR (pptx) from Professor Jianhua Liu
1.4.3 Bash Scripting Cheat Sheet (pdf) from Professor Jianhua Liu
1.4.4 Mini-LibriSpeech with DNN-HMM Based on Wan's Blog (pdf) from Professor Jianhua Liu
1.4.5 Mini-LibriSpeech with GMM-HMM Based on Wan's Blog (pdf) from Professor Jianhua Liu

~~This section provides a bibliography of key project references and deliverables that have been produced before this point.~~

1.5 Glossary

ERAU - Embry Riddle Aeronautical University
ASR - Automatic Speech Recognition, …
ATC - Air Traffic Control, …
HMM - Hidden Markov Model, used to model transitions of tri-phones.
GMM - Gaussian Mixture Model, …
DNN -
MFCC - Mel-frequency cepstral coefficients, …
WER - Word error rate, the rate at which error in words occurs.
IPA - International Phonetic Alphabet, …
NLP - Natural Language Processing, …
Phone - *pertaining to speech*, a distinct speech sound or gesture.
Triphone - *pertaining to speech*, a sequence of three consecutive phonemes.

~~Supply a glossary of all terms and abbreviations used in this document. If the glossary is several pages in length, it may be included as an appendix.~~

Commented [2]: Maxwell Moolchan

Commented [3]: Maxwell Moolchan

Commented [4]: how detailed does this need to be, ex. should I include documents and presentations that Liu gave us?

Commented [5]: as detailed as we can make it anything that we use for the project we should have in here

Commented [6]: started adding terms, they need explanation beside terms

2 SYSTEM ARCHITECTURE

<In this section, describe the system and/or subsystem(s) architecture for the project. References to external entities should be minimal, as they shall be described in detail in Section 6, External Interfaces.>

2.1 System Hardware Architecture

~~In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components. If appropriate, use subsections to address each subsystem.~~

This product does not have a hardware component.

2.2 System Software Architecture

In this section, describe the overall system software and organization. Include a list of software modules (this could include functions, subroutines, or classes), computer languages, and programming computer-aided software engineering tools (with a brief description of the function of each item). Use structured organization diagrams/object-oriented diagrams that show the various segmentation levels down to the lowest level. All features on the diagrams should have reference numbers and names. Include a narrative that expands on and enhances the understanding of the functional breakdown. If appropriate, use subsections to address each module.

2.3 Internal Communications Architecture

In this section, describe the overall communications within the system; for example, LANs, buses, etc. Include the communications architecture(s) being implemented, such as X.25, Token Ring, etc. Provide a diagram depicting the communications path(s) between the system and subsystem modules. If appropriate, use subsections to address each architecture being employed.

The internal communications architecture of our ASR (Automatic Speech Recognition) system designed for ATC (Air Traffic Control) applications is centered around the efficient flow of data and interactions among various software modules. As our project focuses on speech recognition and signal processing, our communication framework prioritizes the seamless transfer of information and data transformations within the system. Key components include the Audio Input Module, responsible for capturing audio input from ATC communication channels and communicating it to the Audio Processing Module, which applies signal processing techniques. The ASR Engine then transcribes the processed audio data into text. Coordination and control are ensured by the Control Module, which orchestrates interactions between modules. Data processing and transformation stages, such as pre-emphasis, framing, and Mel Frequency wrapping, prepare the audio data for analysis. Probability models, including Gaussian

Commented [7]: Tahmina Tisha

Commented [8]: which section?, this is subdivided into 3 sections

Commented [9]: do we have a hardware component?

Commented [10]: we shouldn't

Commented [11]: maxwell moolchan

Commented [12]: Tahmina Tisha

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Commented [13]: Milan

Formatted: Highlight

Mixture Models (GMMs) and Hidden Markov Models (HMMs), play a pivotal role in recognizing spoken words, aided by decision trees to reduce complexity. Finally, the Weighted Finite State Transducer (WFST) facilitates the mapping of phone sequences to recognizable words. This architecture, while unconventional in terms of networking technologies, is fundamental to transforming raw audio data into transcribed text, serving as a valuable tool for enhancing ATC operations.

Commented [14]: @tabitha@elementalscience.com

Commented [15]: I got the information from the notes that we took on 9/21

Note: The diagrams should map to the FRD context diagrams.

3 HUMAN-MACHINE INTERFACE

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs.

3.1 Inputs

This section is a description of the input media used by the operator for providing information to the system; show a mapping to the high-level data flows described in Section 1 .2.1, System Overview. For example, data entry screens, optical character readers, bar scanners, etc. If appropriate, the input record types, file structures, and database structures provided in Section 3, File and Database Design, may be referenced. Include data element definitions, or refer to the data dictionary.

Provide the layout of all input data screens or graphical user interfaces (GUTs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary.

This section should contain edit criteria for the data elements, including specific values, range of values, mandatory/optional, alphanumeric values, and length. Also address data entry controls to prevent edit bypassing.

Discuss the miscellaneous messages associated with operator inputs, including the following:

- Copies of form(s) if the input data are keyed or scanned for data entry from printed forms
- Description of any access restrictions or security considerations
- Each transaction name, code, and definition, if the system is a transaction-based processing system

For this project, most user and system input UI elements and graphic representations shall be handled by the NeMo team, not the Kaldi team. The only input to the system shall be an audio transmission from selected aircraft or ATC towers. The selection of the aircraft or ATC tower shall be managed by the NeMo app and not Kaldi. The input and processing of the audio shall be managed by Kaldi.

3.2 Outputs

This section describes the system output design relative to the user/operator; show a mapping to the high-level data flows described in Section 1.2.1. System outputs include reports, data display screens and GUIs, query results, etc. The output files are described in Section 3 and may be referenced in this section. The following should be provided, if appropriate:

- Identification of codes and names for reports and data display screens
- Description of report and screen contents (provide a graphic representation of each layout and define all data elements associated with the layout or reference the data dictionary)
- Description of the purpose of the output, including identification of the primary users
- Report distribution requirements, if any (include frequency for periodic reports)
- Description of any access restrictions or security considerations

For this project, there shall be little to no user output or display for the user, as that shall be handled by the NeMo team. The only output of the system shall be a speech transcription displayed on the NeMo app. There shall not be a UI element created by the Kaldi team to display text output.

4 DETAILED DESIGN

Formatted: Highlight

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

4.1 Hardware Detailed Design

Formatted: Pattern: Clear, Highlight

A hardware component is the lowest level of design granularity in the system. Depending on the design requirements, there may be one or more components per system. This section should provide detailed information about individual component requirements to correctly build and/or procure all the hardware for the system (or integrate COTS items).

If there are many components or if the component documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each component and its functions, adequately. Industry-standard component specification practices should be followed. For COTS procurements, if a specific vendor has been identified, include appropriate item names. Include the following information in the detailed component designs (as applicable):

- Power input requirements for each component
- Signal impedances and logic states
- Connector specifications (serial/parallel, 11-pin, male/female, etc.)
- Memory and/or storage space requirements
- Processor requirements (speed and functionality)
- Graphical representation depicting the number of hardware items (for example, monitors, printers, servers, I/O devices), and the relative positioning of the components to each other
- Cable type(s) and length(s)
- User interfaces (buttons, toggle switches, etc.)
- Hard drive/floppy drive/CD-ROM requirements
- Monitor resolution

This project has no hardware requirements and therefore, has no hardware design requirements either.

4.2 Software Detailed Design

Commented [16]: Milan

Formatted: Highlight

Formatted: Highlight

A software module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs).

If there are many modules or if the module documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Industry-standard module specification practices should be followed. Include the following information in the detailed module designs:

- A narrative description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.; explain any algorithms used by the module in detail
- For COTS packages, specify any call routines or bridging programs to integrate the package with the system and/or other COTS packages (for example, Dynamic Link Libraries)
- Data elements, record structures, and file structures associated with module input and output
- Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.)
- Data entry and data output graphics; define or reference associated data elements; if the project is large and complex or if the detailed module designs shall be incorporated into a separate document, then it may be appropriate to repeat the screen information in this section
- Report layout

4.3 Internal Communications Detailed Design

Commented [17]: Milan

Formatted: Highlight

Formatted: Highlight

If the system includes more than one component there may be a requirement for internal communications to exchange information, provide commands, or support input/output functions. This section should provide enough detailed information about the communication requirements to correctly build and/or procure the communications components for the system. Include the following information in the detailed designs (as appropriate):

- The number of servers and clients to be included on each area network
- Specifications for bus timing requirements and bus control
- Format(s) for data being exchanged between components
- Graphical representation of the connectivity between components, showing the direction of data flow (if applicable), and approximate distances between components; information should provide enough detail to support the procurement of hardware to complete the installation at a given location
- LAN topology

5 EXTERNAL INTERFACES

External systems are any systems that are not within the scope of the system under development, regardless whether the other systems are managed by the State or another agency. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

5.1 Interface Architecture

In this section, describe the interface(s) between the system being developed and other systems; for example, batch transfers, queries, etc. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide a diagram depicting the communications path(s) between this system and each of the other systems, which should map to the context diagrams in Section 1.2.1. If appropriate, use subsections to address each interface being implemented.

5.2 Interface Detailed Design

For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. This section should provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate):

- The data format requirements; if there is a need to reformat data before they are transmitted or after incoming data is received, tools and/or methods for the reformat process should be defined
- Specifications for hand-shaking protocols between the two systems; include the content and format of the information to be included in the hand-shake messages, the timing for exchanging these messages, and the steps to be taken when errors are identified
- Format(s) for error reports exchanged between the systems; should address the disposition of error reports; for example, retained in a file, sent to a printer, flag/alarm sent to the operator, etc.
- Graphical representation of the connectivity between systems, showing the direction of data flow
- Query and response descriptions

If a formal Interface Control Document (ICD) exists for a given interface, the information can be copied, or the ICD can be referenced in this section.

Formatted: Highlight

Commented [18]: Milan

Formatted: Highlight

Formatted: Highlight

Commented [19]: Milan

Formatted: Highlight

Formatted: Highlight

6 SYSTEM INTEGRITY CONTROLS

Commented [20]: Milan

Formatted: Highlight

Formatted: Highlight

Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled.

Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:

- Internal security to restrict access of critical data items to only those access types required by users
- Audit procedures to meet control, reporting, and retention period requirements for operational and management reports
- Application audit trails to dynamically audit retrieval access to designated critical data
- Standard Tables to be used or requested for validating data fields
- Verification processes for additions, deletions, or updates of critical data

Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.