TED UNIVERSITY

CMPE 491

Senior Project I

PROJECT NAME

Development of a Turkish Language Processing System with Integrated Speech Recognition and Synthesis

High-Level Design Report

Submission Date: 27.12.2024

Team Members:

Doğa Paksoy	10078725538
Rana Çetinkaya	31726748334
Berrin Uzun	15982088236

Table of Contents

1 Introduction

1.1 Purpose of the system

The purpose of this system is to create a Turkish language processing system that combines speech synthesis and recognition. The goal of this solution is to enable smooth speech-to-text (STT) and text-to-speech (TTS) features so that users may communicate in Turkish naturally.

1.2 Design Goals

The following objectives guided the system's design:

- Enhance Speech Processing: Use machine learning models to process Turkish words accurately and naturally.
- Support Diverse Dialects: Use a large dataset for training in order to account for different Turkish dialects and accents.
- Real-Time Performance: Make sure that both TTS and STT functions respond with minimal latency.
- Seamless Integration: For further functionality, offer FNSS services API-based compatibility.
- Robust Noise Reduction: To improve clarity in difficult acoustic conditions, use sound isolation.

1.3 Definitions, acronyms, and abbreviations

- TTS: Text-to-Speech
- STT: Speech-to-Text
- NLP: Natural Language Processing
- FNSS: FNSS Savunma Sistemleri A.Ş.
- API: Application Programming Interface
- GDPR: General Data Protection Regulation
- KVKK: Personal Data Protection Law in Turkey

1.4 Overview

This report outlines the design and development of a system that offers higher-quality Turkish language processing. The system combines vast datasets and advanced machine learning techniques to provide accurate speech synthesis and recognition. Important design components include user-friendly interfaces, persistent data management, and subsystem decomposition. The following sections provide more information on the architecture, features, and implementation strategies to achieve the system's intended objectives.

2 Current software architecture (if any)

3 Proposed Software Architecture

The "Development of a Turkish Language Processing System with Integrated Speech Recognition and Synthesis" project's suggested software architecture is made to process Turkish text and speech effectively while guaranteeing the smooth integration of speech-to-text (STT) and text-to-speech (TTS) features. To provide precise, real-time language processing while meeting a range of user demands and system specifications, the architecture consists of several fundamental components.

Components:

- **Speech Recognition Module:** Uses machine learning models to translate spoken Turkish into written text while maintaining low latency and real-time performance.
- **Speech Synthesis Module**: Converts written Turkish text into genuine speech with programmable volume, speed, and pitch.
- **Noise Reduction Module**: Enhances the precision and clarity of speech inputs by utilizing adaptive noise reduction and sound isolation techniques.
- **Data Integration Module**: Optimizes model performance and cultural adaptability by including a variety of Turkish dialect datasets.
- **User Interface (UI)**: Enables establishing configuration, performance monitoring, and access to TTS and STT features through an intuitive and interactive interface.
- **API Integration Module**: Ensures that data interchange and communication with FNSS services and other external systems run well.

- **Admin Dashboard**: Enables system configuration management, error reporting, and real-time monitoring.

Modules interact to deliver robust functionality:

1. In real time, the Speech Recognition Module converts spoken input into text.
2. To improve input quality, the Noise Reduction Module separates background noise.
3. Using text inputs, the voice Synthesis Module produces natural-sounding, intelligible Turkish voice.
4. Results are displayed through the User Interface.
5. Admins use the dashboard for performance tracking and error resolution.

3.1 Overview

The system aims to create a robust Turkish language processing solution by integrating advanced machine learning and speech processing technologies. Its primary goals include:

1. Delivering top-notch TTS and STT features for Turkish while taking cultural and dialectal variances into account.
2. Ensuring low latency and real-time operation for efficient offline and online use.
3. Utilizing adaptive noise reduction techniques to improve the intelligibility of speech input.
4. Facilitating easy API interface with FNSS services.
5. Supplying an intuitive user interface for simple operation and accessibility.
6. Utilizing Python-based frameworks for effective model construction and training, such as TensorFlow and Keras.

3.2 Subsystem Decomposition

Interactions:

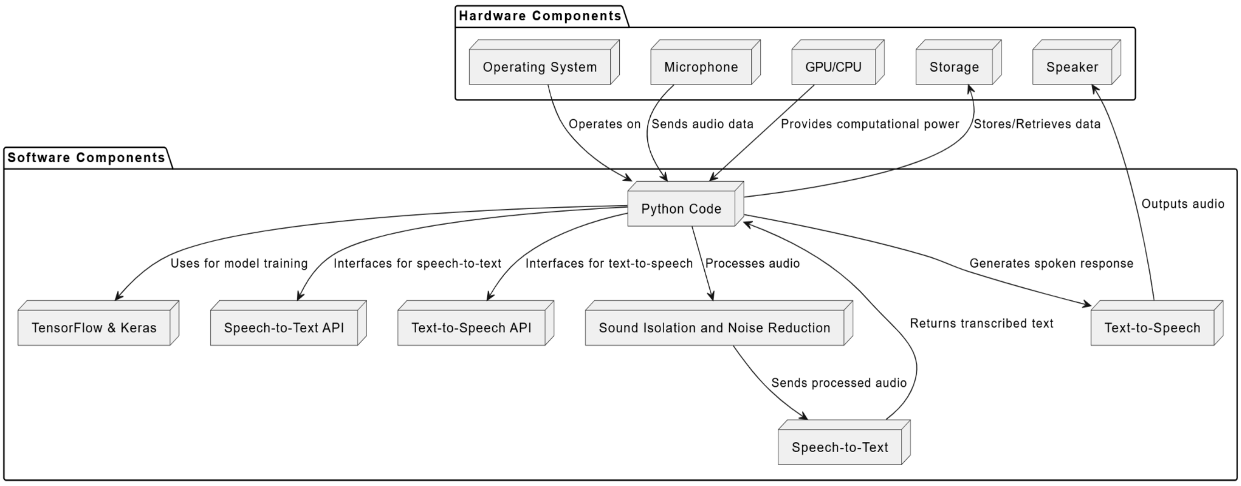
- Text that has been transcribed is sent to the user interface by the speech recognition module.

- The Speech Synthesis Module produces speech after receiving text input from the user interface.
- Prior to the Speech Recognition Module processing audio input, the Noise Reduction Module enhances its quality.
- Models are trained using datasets of Turkish accents and dialects by the Data Integration Module.
- For seamless data interchange, external services are connected with the API Integration Module.

Dependencies:

- For clear input, the Speech Recognition Module relies on the Noise Reduction Module.
- The Speech Synthesis Module depends on the user interface providing high-quality text input.
- To train models using a variety of datasets, the Data Integration Module is required.
- Interoperability with FNSS services is guaranteed by the API Integration Module.

3.3 Hardware/Software Mapping



3.4 Persistent Data Management

Local Filesystem

Training Data: Sets of datasets arranged for efficient access and versions that are used to train models to handle Turkish language nuances effectively.

User Interface Design: A simple user interface that makes TTS/STT features easily accessible.
Machine Learning Frameworks: TensorFlow and Keras are used to create models effectively.

Model Checkpoints: Model states saved during training that enable performance analysis and recovery at different periods in time.

Evaluation Results: Locally stored performance indicators, like accuracy and latency, are used to evaluate the efficacy of the model and guide future developments.

Backup: Consistent backups of important data, such as logs and configurations, to guarantee system recovery in the event of an incident.

Cloud Storage

Interface with FNSS Services: Facilitates seamless data interchange with FNSS services through API-based interface, which frequently entails cloud-based interactions for data processing and storage.

Data Accessibility: Provides team members with remote access to training data and assessment outcomes.

Scalability: Allows for expanding data requirements without necessitating modifications to local infrastructure.

Data Sharing: By enabling simple file sharing between team members and stakeholders, Google Drive eliminates the need for time-consuming file transfers and guarantees that everyone has access to the most recent information.

Version control: Helps manage datasets and model versions by tracking file changes over time.

3.5 Access control and security

1. Encrypting Data: Sensitive information will be encrypted both in transit and at rest to prevent unwanted access and guarantee privacy laws are followed.

2. Secure APIs: To guarantee data confidentiality and integrity during exchanges, FNSS service integration APIs will make use of HTTPS and authentication tokens.

3. Compliance with Regulations: The system will comply with relevant data protection regulations, guaranteeing that all security protocols fulfill the legal requirements for managing personal information.

4. User Awareness and Training: Users will be trained in security best practices, which will raise awareness of possible risks and improve system security overall.

3.6 Global Software Control

1. User Authentication:
   The platform is only accessible by authorized users due to a strong user authentication method built into the system. Secure user credential management preserves data integrity and guards against unwanted access.
2. Speech Processing Execution:
   The global control initiates the TTS and STT features after a user authenticates. It ensures smooth functioning by controlling the execution and termination of speech-processing tasks.

3. Error Handling:

   The global control system manages exceptions and faults throughout the speech processing operations by integrating with Python-based frameworks such as TensorFlow and Keras. This guarantees that any problems are recorded and addressed without interfering with user interactions.
4. Coordination:

   The interactions between the FNSS services, backend processing models, and user interface are synchronized via global control, which serves as a central coordinator. It guarantees that information moves smoothly and that tasks are carried out in the right order.
5. Security:

   The worldwide control system ensures that data is secure and that KVKK and GDPR requirements are followed. It ensures that information is processed and kept safely by protecting sensitive user data using access control and encryption.

3.7 Boundary Conditions

The limitations, presumptions, and restrictions that govern the "Development of a Turkish Language Processing System with Integrated Speech Recognition and Synthesis" are outlined in boundary conditions. These guarantee that the system continues to operate and function at its best under the given circumstances.

Assumptions:

- **Stable Turkish Dialects:** The system requires few model updates since it considers that Turkish dialects and linguistic subtleties stay mostly constant throughout time.
- **Operational Environment:** To guarantee precise STT processing, the system must operate in controlled settings with little background noise.
- **Data Availability:** For model training and fine-tuning, it is assumed that a varied and high-quality dataset representing different Turkish dialects and accents would be available.
- **API Reliability:** It is expected that the external APIs used to integrate with FNSS services would continue to be reliable and appropriate for the needs of the system.
- **User Expertise:** To utilize the system and comprehend error messages, users are assumed to possess rudimentary digital abilities.

Limitations:

- **Performance in High Noise:** Even with noise isolation strategies, the system's performance may suffer in settings with a lot of background noise.
- **Device Compatibility:** Some functions could be platform-specific even though the device is made to work on multiple platforms (Windows, macOS, iOS, and Android).
- **Real-Time Processing Latency:** Due to hardware constraints or heavy computational demands, real-time TTS and STT processing may experience minor delays.
- **Dialect-Specific Variances:** When processing rare or not well documented Turkish dialects, the system may show decreased accuracy.

Constraints:

- **Regulatory Compliance:** GDPR, KVKK, and other pertinent data privacy laws must be complied with by the system; these laws may place limitations on the processing and retention of data.
- **Hardware Resources:** The system's ability to process data in real time is limited by the availability of enough GPU and RAM resources.
- **Network Dependency:** Only predefined functionality will be available offline; cloud-based features require consistent internet connectivity.
- **Third-Party Dependencies:** To prevent incompatibilities, the system depends on third-party libraries and tools (such as TensorFlow and Keras), which need to be updated and maintained.

4 Subsystem Services

1. Text-to-Speech (TTS) Subsystem: Translates written Turkish text into speech that sounds natural. To accommodate different dialects and user preferences, it enables speech aspects like pitch, speed, and volume to be changed.

2. STT Subsystem (Speech-to-Text):
Various dialects and colloquial idioms are accommodated in the precise transcription of spoken Turkish into written text by the STT subsystem. It is designed with real-time processing with the goal to ensure great accuracy and minimal latency.

3. Sound Isolation and Noise Reduction: Uses adaptive noise reduction techniques to improve voice clarity during TTS and STT procedures. The main audio input is made clear and understandable by filtering out background noise.

4. Reporting Subsystem: Users can filter results by different metrics and time periods using the comprehensive reports on TTS and STT performance that are generated by this subsystem. It facilitates monitoring system performance and highlighting areas that are require development.

5. Interface Subsystem: This subsystem enables seamless data interchange between the Turkish Language Processing System and external systems by facilitating API-based interface with FNSS services.

5 Glossary

TTS (Text-to-Speech): A technology that produces audio output that sounds natural by translating written text into spoken language.
STT (Speech-to-Text): A technology capable of accurately translating spoken conversation into written text.
NLP (Natural Language Processing): A branch of artificial intelligence that aims to give computers the ability to meaningfully comprehend, interpret, and react to human languages.
FNSS (FNSS Savunma Sistemleri A.Ş.): FNSS was founded as the first private defense industry business in Turkey. It is owned entirely by Nurol Holding Inc. and focuses on producing armored combat vehicles. It is also a pioneer in defense industry initiatives.[1]
API (Application Programming Interface): A collection of protocols and tools that facilitate effective communication and data sharing between various software programs.
GDPR (General Data Protection Regulation): A European Union law that controls the gathering, storing, and processing of personal information in order to protect privacy and security.
KVKK (Personal Data Protection Law in Turkey): A Turkish law that protects and processes personal information while guaranteeing people's right to privacy.
TensorFlow: Google created an open-source machine learning framework that is used to create and train deep learning and machine learning models.
Keras: TensorFlow-based high-level neural network API for efficiently building and training machine learning models.
Noise Reduction: Improving the clarity of audio inputs or outputs by separating and reducing background noise.
Machine Learning: A subfield of artificial intelligence that allows computers to gradually improve their performance without explicit programming by learning from data.

Speech Synthesis: TTS technology is used to create artificial human speech in order to produce audio that sounds natural.

Speech Recognition: Utilizing STT technology to translate spoken words into text while accounting for various dialects and accents.

Dataset: A set of data, frequently customized for particular activities or languages, that is used to train and evaluate machine learning models.

Real-Time Processing: The capacity to analyze information or carry out calculations quickly or efficiently is essential for TTS and STT features.

Sound Isolation: Techniques for improving speech input quality in noisy settings by removing background noise.

6 References

[1]: FNSS History