Software Requirements Specification

for

Speaker Diarization

Version 1.0 approved

Prepared by Apoorva Iyer

Vinita Khushalani

Deepika Kini

Simran Makhija

Thadomal Shahani Engineering College

05-09-2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Speaker diarisation (or diarization) is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. It can enhance the readability of an automatic speech transcription by structuring the audio stream into speaker turns and, when used together with speaker recognition systems, by providing the speaker's true identity.It is used to answer the question "who spoke when?".

1.2 Document Conventions

The titles have been specified in bold and in Times New Roman Font.The font size used here is 18.The subtitles are specified in bold but with font size 14. The description are specified in Times New Roman in 12 font size.

1.3 Intended Audience and Reading Suggestions

The different types of reader that the document is intended for are developers, project managers, marketing staff, users, testers, and documentation writers. The rest of this SRS contains the description, features, scope and applications of the product.

1.4 Product Scope

The main objective of the software is to separate speech of individuals and identify a voice to be of a particular individual. This can be used for medical transcription as well as in speech recognition.

Transcription is where the audio files is converted to text files such as databases and documents.

1.5 References

The following sites were referred

http://www1.icsi.berkeley.edu/~vinyals/Files/taslp2011a.pdf

https://arxiv.org/pdf/1708.02840.pdf

https://www.researchgate.net/publication/317724530_Speaker_diarization_using_deep_neural_network_embeddings

2. Overall Description

2.1 Product Perspective

Speaker Diarization aims to automate the process of identifying individuals. The applications of diarization- The current systems used for medical transcriptions are humans. This may result into various errors. The current identification methods are fingerprint biometrics, palm biometrics. Voice Biometrics is much easier to implement.

Inspired by recent advancements in the speaker diarization domain achieved with convolutional neural networks (CNNs) and successful applications of recurrent convolutional neural networks (R-CNNs) to other problems (such as image classification or birds' sound classification) we propose employing a recurrent neural network architecture to speaker diarization problem. Our motivation to use this approach is based on the observation that the temporal patterns present in audio streams can be better aggregated and interpreted with recurrent architectures, due to the specific feedback embedded in their design

2.2 Product Functions

Speaker Diarization aims to automate the process of identifying individuals. It aims to answer the question of "who spoke when". From a set of speech corpus it should be able to identify the individuals.

2.3 User Classes and Characteristics

Speaker Diarization when put to use in the medical transcription will be used by individuals in billing and accounting field. For voice biometrics, the system will be utilised by individuals within an organisation to identify themselves.

2.4 Operating Environment

The software should be operable on common Operating systems like Windows and also on open source operating Systems such as Linux,Unix. Packages of python would be required to run the code.

2.5 Design and Implementation Constraints

Developers would require python packages to create the code. Anaconda python distribution is preferred tp create .py files. The usual standards of the programming language is considered.

2.6 User Documentation

User manuals and tutorials will be provided along with the product.

2.7 Assumptions and Dependencies

It is assumed that the noise is completely removed in the preprocessing step. Hence the diarisation process depends on the data preprocessing process. The software would require the python packages too.

3. External Interface Requirements

3.1 User Interfaces

A GUI (Graphical User Interface) for easy loading of audio files. The interface would consist buttons to start stop the recording and load in the software.

3.2 Hardware Interfaces

A microphone would be required. As well as a recorder to record the data set. Microphones are easily available on all computers and mobiles.

3.3 Software Interfaces

The software will be designed to run on operating systems like Windows, Linux. It would require the packages of python like numpy to build the codes and run it too.

3.4 Communications Interfaces

The software may require FTP and HTTP for updation of software. Otherwise as the software is not based on a network, it does not require communication protocols for functioning.

4. System Features

4.1 Speech Separation

4.1.1 Description and Priority

Speaker diarization is the task of identifying “who spoke when?” in an audio stream. The system does not assume any prior knowledge about the speakers or the number of speakers in a given audio . Speaker diarization has many applications, especially for tagging the audio in telephone

conversations, broadcast news and meetings. Conversational meetings are spontaneous and therefore challenging. Diarization of meetings is a task that has received significant attention. The approaches to speaker diarization includes top-down, bottom-up, parametric and non-parametric clustering. Bottom-up agglomerative clustering is the most popular approach. The state of the art speaker diarization systems include Hidden Markov Model/ Gaussian Mixture Model (HMM/GMM) and Information Bottleneck (IB) systems.

1. Identifying the number of speakers: For correctly identifying the speakers, it is necessary to find the number of speakers in the dataset. The number should be accurate or else the whole concept of the application for speaker diarization will fail.
2. Identifying the speakers: For speaker recognition, an accurate dataset of each speaker is needed before-hand for correctly labeling speakers.

4.1.2 Functional Requirements

REQ 1:- Segmentation and Clustering:

Segmentation and clustering modules are part of most LVCSR systems. A segmentation module is responsible for segmenting speech input in smaller chunks that can be processed by the recognizer directly. Often the segmenter also filters out non-speech such as silence, lip-smacks, laughter or even tunes or sound effects. The clustering module is used to group together segments with similar characteristics. Obvious characteristics to cluster on are audio channel (broadband/telephony) or gender. Some clustering systems, called speaker diarization systems, are able to cluster speech fragments from individual speakers. Using the clustering information the recognizer can process each cluster optimally. For example, special gender dependent models (see section 1.5.6) can be applied when gender information is available or model adaptation techniques can be applied for each separate speaker when a speaker diarization system has been used.

REQ 2:- Agglomerative Clustering:

Agglomerative clustering consists of four iterative steps. First, initial clusters need to be defined. Often, each speech segment that is found during segmentation is considered a single cluster. Next, the distance between these clusters needs to be determined. Often this is done pairwise and the distance between each pair of clusters is stored in a matrix. This matrix is used in the third step to determine if there are any clusters that can be merged into one cluster or if the optimum number of clusters is reached. If this stopping criterion decides that the optimum is not yet reached, in the fourth step the clusters with the smallest distance are merged and the process is iterated starting at the second step. The distance metrics that are used to determine the distance matrix, are often the same metrics as used during segmentation. This means that for each cluster a model is created and during the merging phase a new model is created for each pair of clusters that are merged.

REQ 3:- Assessment of Speaker Diarization:Speaker diarization systems need to segment and cluster audio recordings on speakers. The metric used to evaluate the performance of these systems is called Diarization Error Rate (DER) [NIS07]. It is computed by first finding an

optimal one-to-one mapping of the reference speaker segments to system output and then obtaining the error as the fraction of time that the system did not attribute correctly to a speaker or to non-speech. Finding the optimal mapping is needed because the system does not need to identify speakers by name and therefore its speaker labels will differ from the labels in the reference transcript.

REQ 4:- Acquisition of Speech Signal :

The acoustic wave speech signal generated by humans can be converted into an analog signal using a microphone. An antialiasing filter is thereafter used to condition this signal and additional filtering is used to compensate for the channel impairments. The antialiasing filter band limits the speech signal to approximately the Nyquist rate (half the sampling rate) before sampling. The conditioned analog signal is then sampled by an analog ñ to ñ digital (A/D) converter in order to obtain a digital signal.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. Response Time: For such an application of speaker diarization, the response time should be minimal. This can be done with the help of high-speed computational units. For systems that have to support significant numbers of users the cost of response times delays can actually be measured in monetary terms and therefore can form part of trade-off studies between different architectures providing different levels of performance.For systems that have to support significant numbers of users the cost of response times delays can actually be measured in monetary terms and therefore can form part of trade-off studies between different architectures providing different levels of performance.

2. Scalability: In one respect scalability is simply specified as the increase in the system's workload that the system should be able to process. The scalability required is often driven by the lifespan and the maturity of the system. For example, a new (and hence immature) system could suffer an unexpected growth in popularity and suffer from a significant increase in workload as it becomes popular with new users. In our software, audio datasets will have to be analyzed and added for the purpose of testing purposes to make the model an efficient classifier. Hence, scalability is a necessity.

5.2 Safety Requirements

The application can be misused by using the datasets from tapped phones and used for illegal activities. In such cases, to stay out of legal trouble, such loopholes should be filled in the licensing agreement.

5.3 Security Requirements

The dataset used for testing is from various sites and they need to be verified.Some security industry experts point to vulnerabilities in the current crop of voice-related technologies that make its use on any device questionable. In a voice recognition attack, typical security controls are evaded with fraudulent voice samples.
Even in the user's device, the security should be in order to see that data is not manipulated and integrity is maintained.

5.4 Software Quality Attributes

1. Correctness: Accuracy for labeling the speakers is an important component for this software. For this, the number of speakers should also be identified.

2. Robustness: The algorithm has to be adapted according to the differences in the nature of the data and the environment in which they are recorded. For example, Broadcast News speech data is usually acquired using boom or lapel microphones with some recordings being made in the studio and others in the field. Conversely, meetings are usually recorded using a desktop or far-field microphones (single microphones or microphone arrays) which are more convenient for users than head-mounted or lapel microphones. As a result, the signal-to-noise ratio is generally better for BN data than it is for meeting recordings. Additionally, differences between meeting room configurations and microphone placement lead to variations in recording quality, including background noise, reverberation and variable speech levels (depending on the distance between speakers and microphones). Hence, adaptability is also a major feature of this application.

3.Flexibility: The software should be able to operate on any platform and any device. Also, the audio file format(WAV, AIFF, MPEG, etc.) shouldn't pose a problem, or the format should be specified in advance.

5.5 Business Rules

Since our model uses RNN method, the model should be created by using training datasets. These datasets need to be provided with the number of users and the speakers. Also, the system needs to be trained for understanding and finding patterns in audio using pitch contours and formant frequencies.
The user will have to do both training and testing for his/her specific environment.
Testing should be iterative and incremental(increase the number of users) to build the system.

Appendix A: Glossary

Convolutional Neural Network(CNN) :

Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs. From Hubel and Wiesel's early work on the cat's visual cortex , we know the visual cortex contains a complex arrangement of cells. These cells are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. These cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in natural images.

Additionally, two basic cell types have been identified: Simple cells respond maximally to specific edge-like patterns within their receptive field. Complex cells have larger receptive fields and are locally invariant to the exact position of the pattern.

The animal visual cortex being the most powerful visual processing system in existence, it seems natural to emulate its behavior. Hence, many neurally-inspired models can be found in the literature.

Recurrent Neural Network(RNN) :

A **recurrent neural network** (**RNN**) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that can not be unrolled.

Both finite impulse and infinite impulse recurrent networks can have additional stored state, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory.

Anaconda in Python :

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

Nyquist Frequency:

The Nyquist frequency should not be confused with the Nyquist rate, which is the minimum sampling rate that satisfies the Nyquist sampling criterion for a given signal or family of signals. The Nyquist rate is twice the maximum component frequency of the function being sampled. For example, the Nyquist rate for the sinusoid at 0.6 f_s is 1.2 f_s, which means that at the f_s rate, it is being undersampled. Thus, Nyquist rate is a property of a continuous-time signal, whereas Nyquist frequency is a property of a discrete-time system.

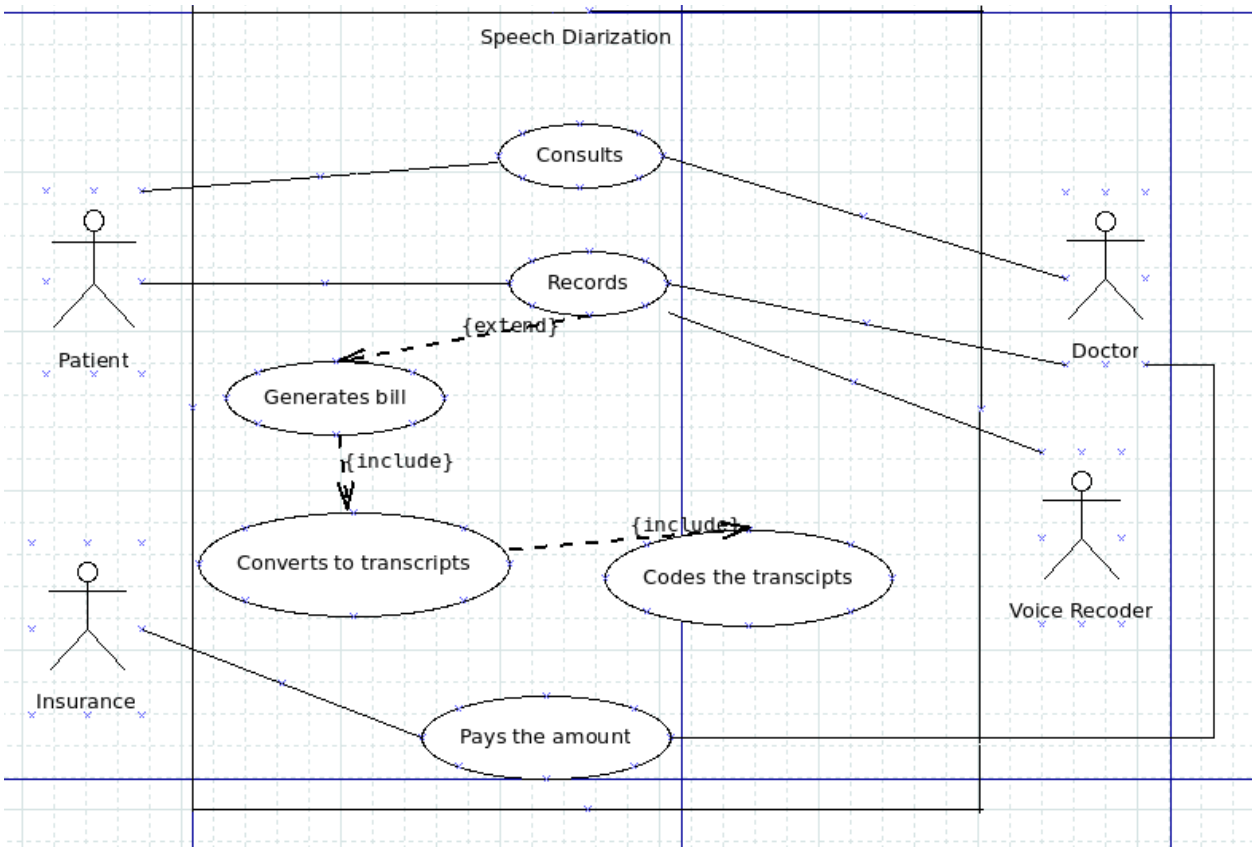
When the function domain is time, sample rates are usually expressed in samples per second, and the unit of Nyquist frequency is cycles per second (hertz). When the function domain is distance, as in an image sampling system, the sample rate might be dots per inch and the corresponding Nyquist frequency would be in cycles/inch.

Large Vocabulary Continuous Speech Recognition(LVCSR) :

Large Vocabulary Continuous Speech Recognition(LVCSR) begins by recognizing phonemes much like a phonetic system,but then applies a dictionary or language model of potentially 50,000 – 1,00,000 words and phrases to produce a full transcript. In LVCSR every word is recognized and nothing is thrown away or skipped.

Appendix B: Analysis Model

Use case in medical-field application:



Appendix C: To Be Determined List

http://www1.icsi.berkeley.edu/~vinyals/Files/taslp2011a.pdf

https://arxiv.org/pdf/1708.02840.pdf

https://www.researchgate.net/publication/317724530_Speaker_diarization_using_deep_neural_network_embeddings

https://www.ibm.com

https://ieeeexplore.ieee.org