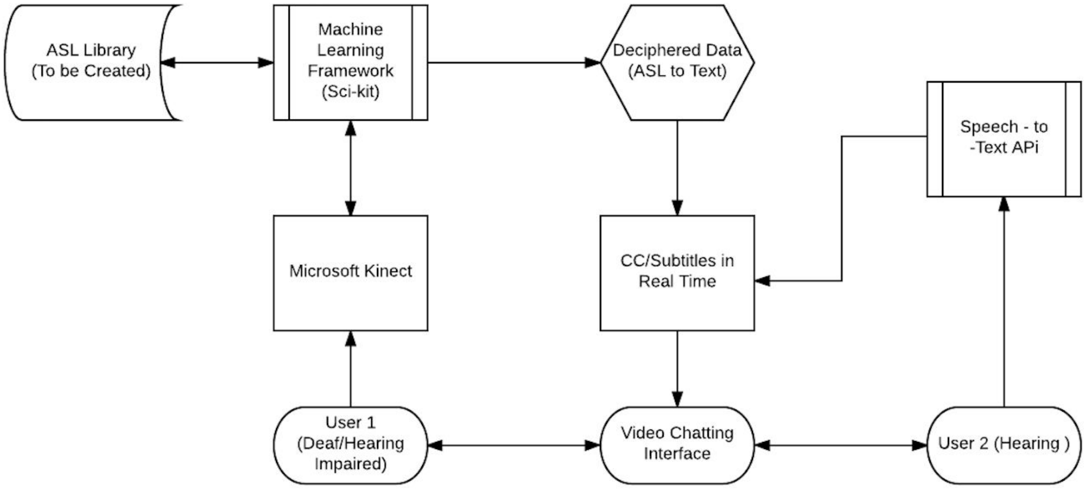
Overall project

The overall project involves creating a video chat interface that will translate American Sign Language (ASL) into text on one side, and it will convert speech to text on the other side. This will allow deaf and hearing-impaired individuals to communicate with hearing individuals that do not understand sign language. This project is broken down into three main modules: converting ASL to text, converting speech to text, and integrating the previous modules into a video chat interface.

The figure below shows the overview of how this system works. User 1, which is the deaf/ hearing impaired individual, will perform sign language into the Microsoft Kinect Sensor. The Kinect sensor will send raw data into the machine learning framework. The machine learning technique will use the ASL library, which will be generated by Elizabeth and Chelsea, to interpret what the user is trying to say. The results from the framework will be sent to a cs (C-sharp) file, where a script will read the words and show it on the screen for the other user. Similarly, User 2, the hearing individual, will talk normally into the video interface. The speech to text API will convert the spoken word into text in real time, and save the text to the cs file for the script to read from.



Responsibilities

Chelsea is responsible for the speech to text module and with integrating that into the video chat interface. She will also create an ASL library that will be used in conjunction with the machine learning aspect. Elizabeth is responsible for implementing the machine learning that is part of the ASL to text module. She will also work with Chelsea to help integrate the three modules into one application.

Speech To Text - Chelsea

For this module, the Google Speech API will convert a person's spoken word into text. As part of this API, a few methods that will be used are SpeechClientCreate(), StreamingRecognize(), and StreamingRecognitionConfig(), and SpeechClientClose().

SpeechClientCreate() - creates an instance of the SpeechClient object with default settings, set by Google.
StreamingRecognize() - analyzes the speech as it is being detected.
StreamingRecognitionConfig() - transforms the speech into text.
SpeechClientClose() - deletes the instance of the SpeechClient Object.

SciKit Learn SVM/ASL To Text/ASL Library API - Elizabeth

EC-Chat will contain an ASL library that scikit learn can use to classify the data. This library will be an open source library as it will need to continue growing as the application is more used. The beta version of EC-Chat will include an ASL library that contains thirty signs, each composed of ten different users. This library and software is built under the assumption of further growth.

The EC-Chat team will maintain their application certified set of ASL signs for the library on their open source page. Any individual or group who wishes to contribute may add to the github repository, however only the EC-Chat team may modify the EC-Chat certified signs. This is an attempt to ensure that the content is accurate.

Users may add to this library by recording the signing of a certain vocabulary word. The video will have specifications on format and type accepted. The English translation must accompany the ASL sign. The EC-Chat software will have the ability to convert the video into the csv format required for the scikit learn to read. As more data is added to the library, English translation of the word will have more signs to add to the correlating word. Due to scikit learn's supervised learning method of using support vector machines, the more videos that are added, the more accurate the estimations will be for further use of the software.

There will be an implemented control to users uploading translations to the library to ensure there are not false or incorrect data that may influence the translations. This will be done by having auditors review and implemented training into our software.

Video Transmission / Integration - Chelsea

This module will require the use an open source video chat interface. The text that is received from the text to speech module and the ASL to text module are written to a cs (c-sharp) file. An XAML script will read from this constantly changing variable to relay the text onto the screen.