iHear – Mobile application which can recognize Speech to generate summary text data

CS5560: Knowledge Discovery and Management

Guided By

Dr. Lee Yugyung

By

Manikanta Maddula (15)

Motivation:

There are approximately 70 million deaf people in the world. For many, hearing aids, sign language, cochlear implants and subtitles are useful. Some develop Lip reading skill. However, access is limited to above mentioned technology for many people. So, a mobile application which can recognize speech from user and display summary text about what user is speaking would be very helpful and cost effective. The idea for the project came from Hearing Dog. It is an assistance dog specially trained to help deaf people. Dog is trained to alert the owner to important sounds like smoke alarms, doorbells, telephone ringing etc. A mobile application can be cost effective since a large amount of users are already using smart mobile systems. And this application serves even more than a hearing dog by recognizing speaker's audio and generates summary data (Key words) instead of displaying the whole text which is not comfortable for the user.

Objective:

There are some mobile applications which can recognize important sounds for users but this application serves more by making use of Machine Learning developments in recent years. There are several speech to text recognition API's available. These can be used to convert audio to text data. There has been a lot of development in natural language processing and semantic applications. Converted text can be summarized by using NLP, machine learning algorithms. But since computing power that is available in mobile is limited, light weight machine learning model can be used in client side to improve the performance and recognition time. Self-Organizing maps and sigspace feature modelling can be used to achieve light weight machine learning. Sample data is used for training (or learning) and model building. Speech to text API's are used to generate text data. From the converted text data, word embedding algorithms can be used to extract keywords. Here Ontology models are generated based on training data. To work with large models or large streaming data, apache spark can be used. Apache Spark is an open source big data processing framework which provides high performance and sophisticated analytics. Dynamic recognition can be used to generate topic data or summary data. A particular domain is selected for the purpose of project and later can be extended it to make it as domain independent.

The overall objective is to build an application which can recognize speech and generate topic data or summary data in text format.

Mobile Application:

Android is the selected mobile platform because by 2015 the amount of android phones brought are 81.61% globally. Application will have simple interface. A start button is used to start the speech recognition and stop can be used to make application stop listening. Most of the screen area is used to present the summary data. As this application deals with streaming data, complete analysis may be presented with some delay for better output. User will have option to save the output.

Domain Definition:

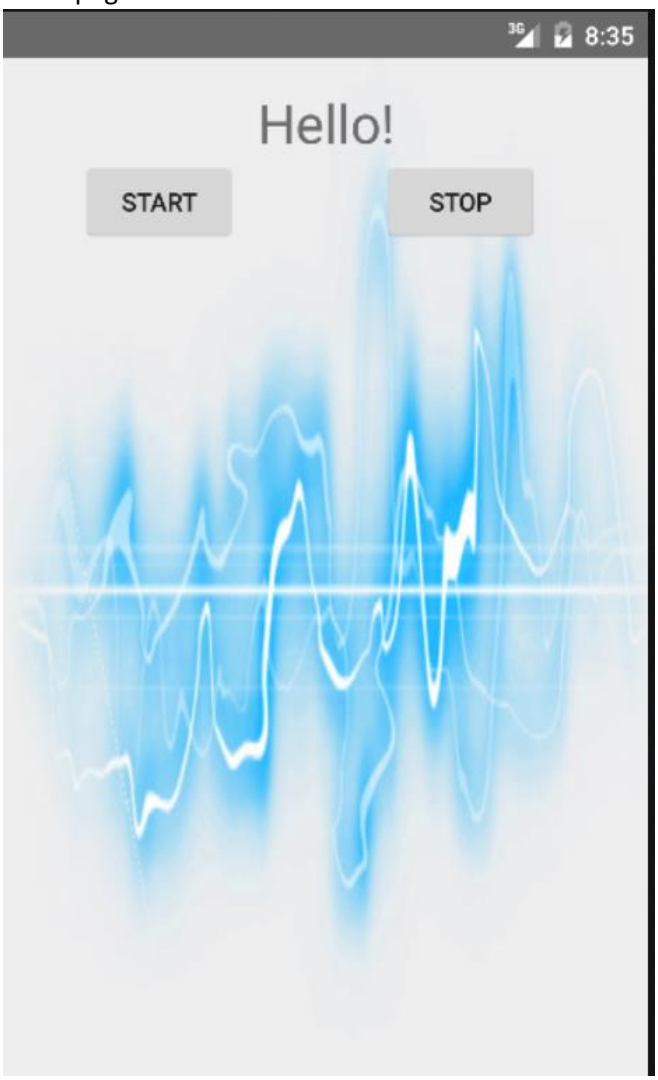
The application focuses on Sports domain. Application will be able to recognize the particular sport, players, event information and game commentary from the audio that the application is listening to. This could be from a speaker, a TV news, a recorded video etc.

Dataset:

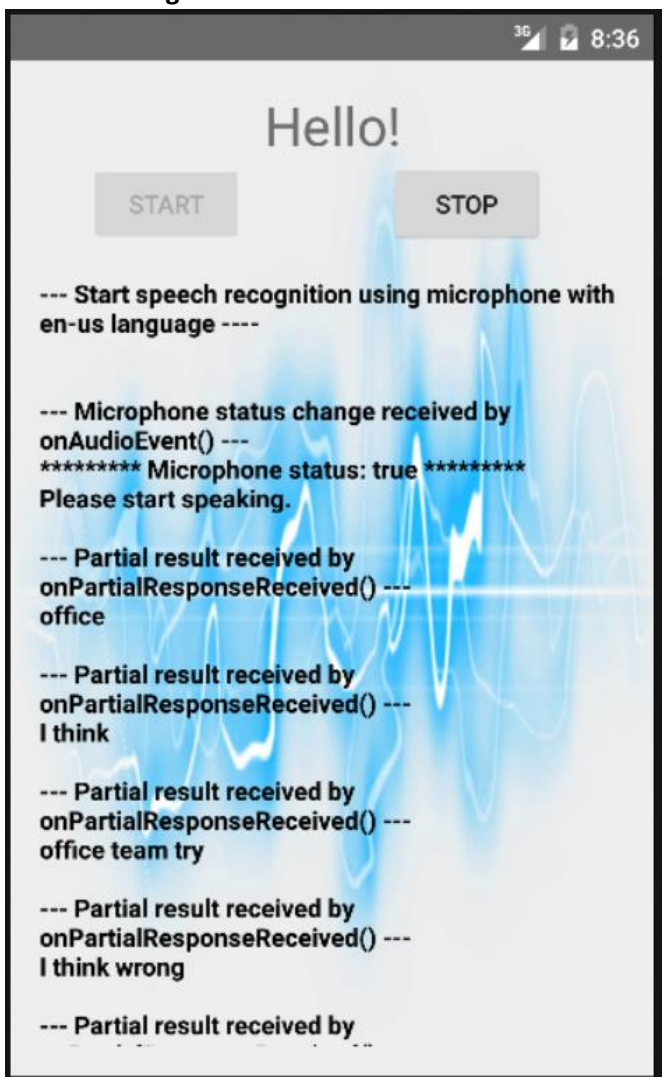
1. BBC Sport website corresponding to sports news in five topical areas from 2004-2005 with 737 documents and 5 natural classes. Source: http://mlg.ucd.ie/datasets/bbc.html
2. 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Source: http://qwone.com/~jason/20Newsgroups/
3. Wikipedia and DBpedia data API.

Tasks Completed in Phase1:

1. Android Application UI:
   Homepage:



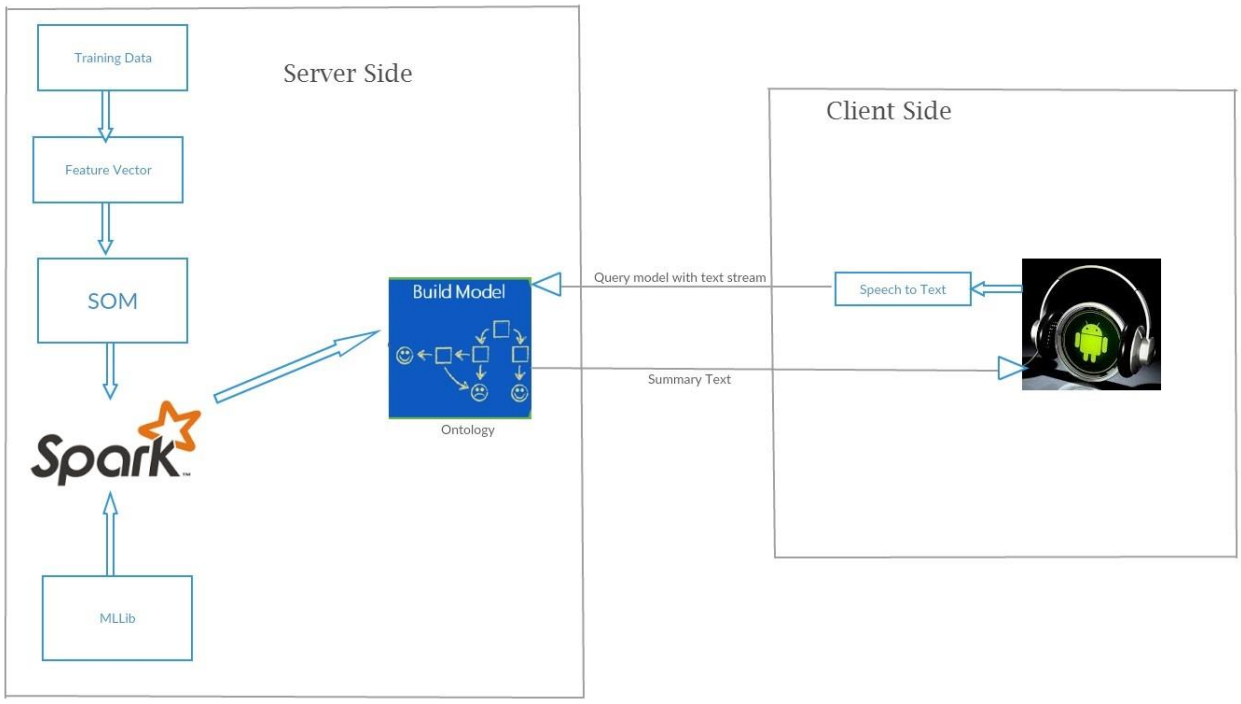
Start Listening:



2. Microsoft Speech to text API implementation in android.
3. System Architecture Design.
4. Stanford Core NLP for NLP operations in Intellij using java.
5. TFIDF for some of the BBC dataset (Cricket category files) is done. Term document frequency is constructed.
   Example input: “Hayden sets up Australia win”
   Output for document1 (stripped):
   (1048576,[45,65,66],[0.9364934391916745,2.3814551551518304,7.751767917624546])

Implementation:

Architecture:

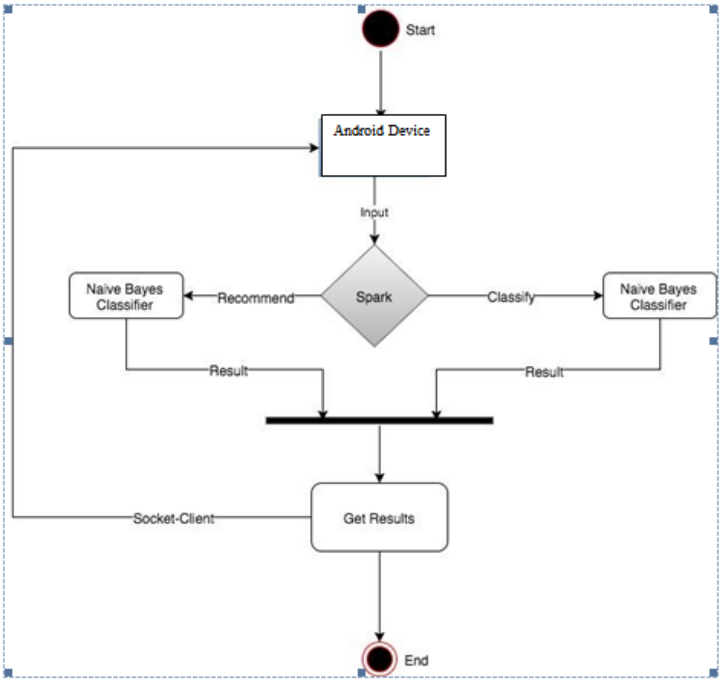


Raw data set is used to get feature vector by using NLP, TF-IDF, Word2Vec and LDA. By applying Machine learning algorithms (using Spark MLLib) to this feature vector a model is trained. Model is an ontology or knowledge graph related to domain. This model can be queried to get desired summary data by providing text stream as input. Features may need to be extracted from the input text data to query model for appropriate key words from input data.

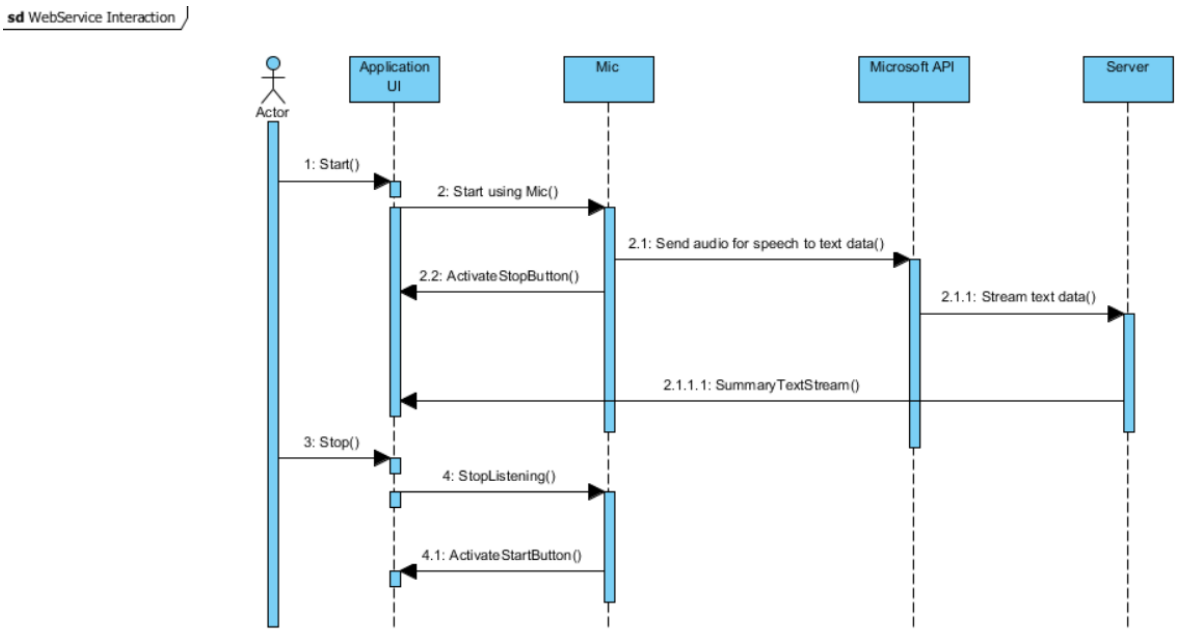
Audio data from android is converted to text using Microsoft Speech to Text conversion. Microsoft provides “Bing” (It’s search engine) API and gives about 5000 transactions per month for free. This text stream is sent to server and summary text data is returned as result from server to android which is displayed in Android View.

Class Diagram:

This figure demonstrates the high level design architecture of the system.



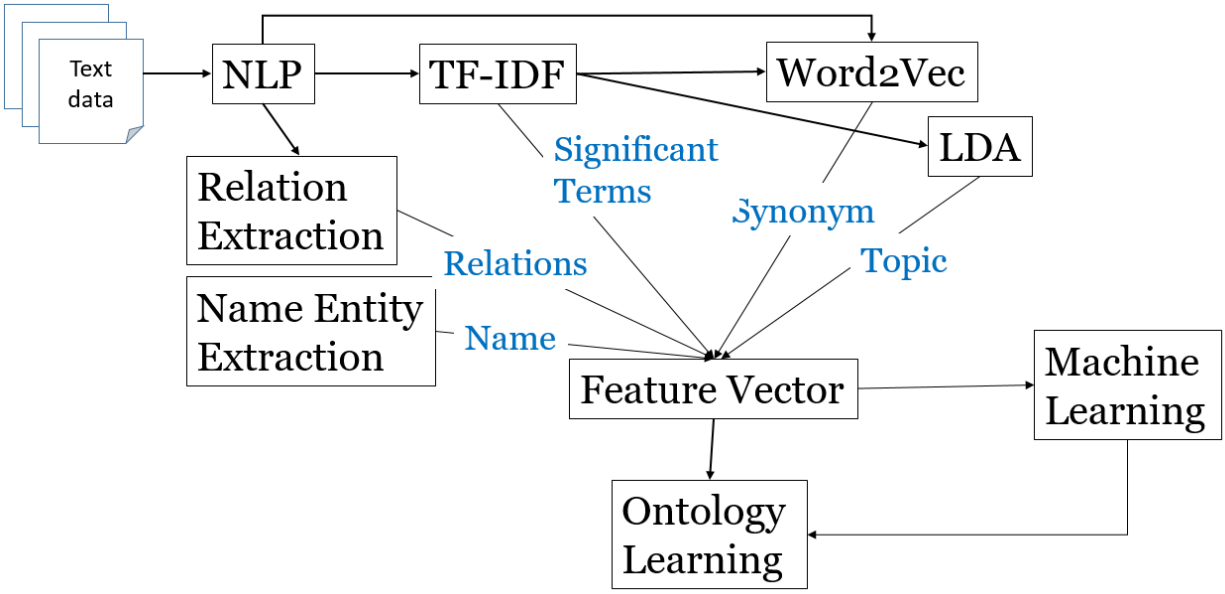
Sequence Diagram:



The above figure is the sequence diagram for application interaction.

Workflow:

First phase of project is to develop Android UI. Then using API to convert speech from text.



Existing Services:

Microsoft speech to text API is used to convert Speech to text.

Stanford Core NLP API is used for NLP operations required for generating feature vectors. (Future)

Spark ML libraries are used for TF-IDF, Word2Vec operations. (Future)

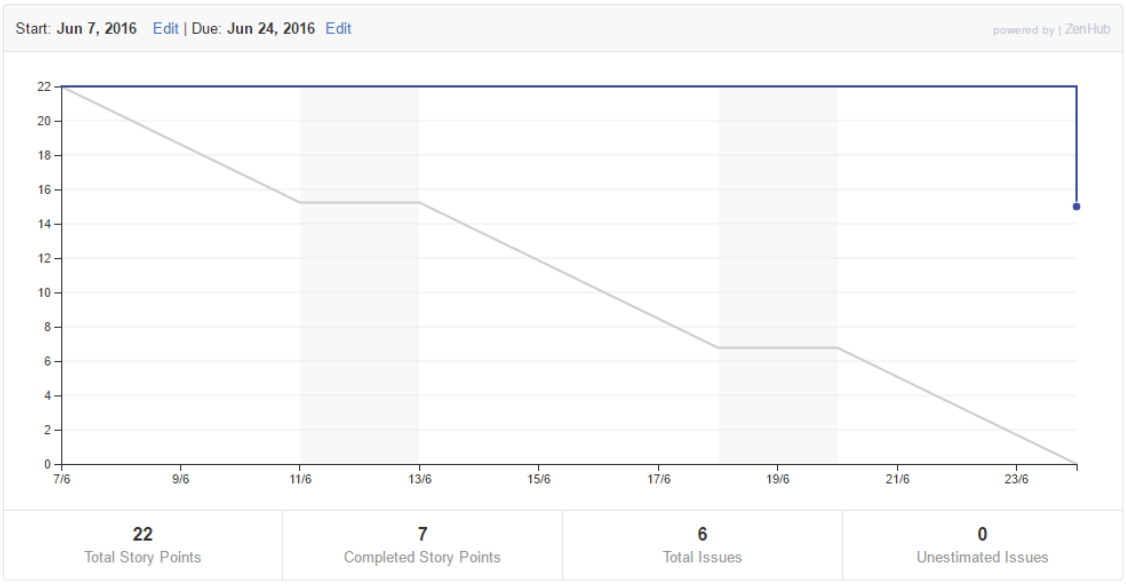
Spark ML Libraries are used for machine learning algorithms. (Future)

SparQL, OWL API, Ontology are used for model construction and querying. (Future)

Android services for audio data generation.

Project Management:

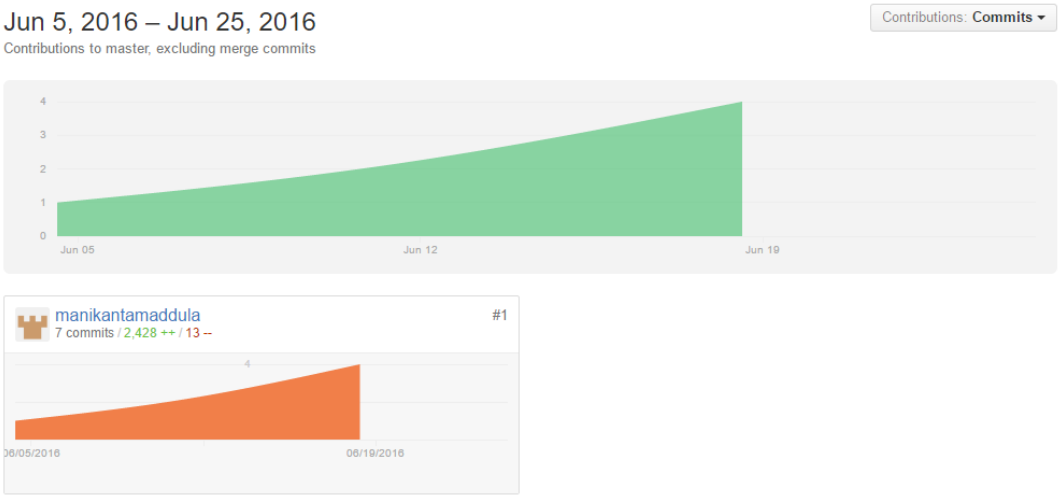
Burndown Chart:



Phase1 Submission

Repository	Issues	Story Points
KDM-Summer-2016-iHear	#6 TFIDF and NLP ccalculations for Dataset collected	13
KDM-Summer-2016-iHear	#1 Training Data	2
KDM-Summer-2016-iHear	#5 Architecture, UML diagram, class diagram etc	2
KDM-Summer-2016-iHear	#4 Application UI	2
KDM-Summer-2016-iHear	#3 Speech to text conversion	2
KDM-Summer-2016-iHear	#2 Documentation	1

Contributions:



Concerns/Issues:

1. Need to collect more datasets. Integrating datasets.
2. Need to figure out working with stream text data for querying server on summarization.
3. Need to know if feature extraction can be done in mobile client instead of server.
4. Can querying of model for summarization be implemented in mobile client.
5. Machine learning for categorization and classification.

Future Work:

Next phase of the project should complete until building model from datasets. This involves generating feature vectors, applying machine learning algorithms to build model.

Third phase of the project implements summarization from the model and audio data in application.