Crowdsourcing for language processing(CLAP)

Suman Swaroop
140050032

May 1, 2018

Guides : Prof. Kameswari Chebrolu and Prof. Preethi Jyothi

1 Introduction

This Project report is for Crowdsourcing for LAnguage Processing Android App, developed by 4th Year CSE IIT, Bombay undergraduate students, Suman Swaroop and Rajat Chaturvedi, under the guidance of Prof. Kameswari Chebrolu and Prof. Preethi Jyothi, as part of their B.Tech Project (BTP II) in the 8th semester.

Crowd-sourcing is a promising method for fast and cheap transcription of large volumes of speech data. It allows a researcher to get a lot of work done with significantly less effort required from an individual, and usually the simplicity of the task allows a layman to contribute in research too. This is especially true for regional Indian languages. All methods for Natural Language Processing require some form of speech along with corresponding text. The unavailability of this data for regional Indian languages makes them very less attractive to the research community. But the presence of large number of speakers of these languages in the country, presents us with an opportunity to make use of Crowd-sourcing to generate the kind of data which will make study of these languages easier.

This app acts as a transcribing tool for large volume of speech data. Apart from this, it has a verification section where users verify and correct speech transcribed by others to reduce errors to the extent possible.Based on the verification and transcribing accuracy of a user, they are given scores.

2 Android

2.1 Introduction

This android app supports min Sdk version 16 and runs flawlessly on the latest Nougat release.The screen scales well on different screen sizes.User interface has been designed for effective and enjoyable use.Backend architecture design helps developers to incorporate new features in an organized manner as well as write unit tests for each sub parts independently.

2.2 New features

- Offline Working: Unavailability of data connection doesn't stop the user.Once the user transcribes the speech data in memory and confirms submission, the request will be stored locally first and later on when data is available it will be synced with server.Every network task is persistent.
- Multiple Tasks: Addition of new types of tasks is easy. Currently there are 3 types of task: Speak, Label and Verify task.
- Leaderboard: This shows top 10 performers. Added to setup a competitive environment for the user and motivate him to do more tasks.
- Audio preprocessing: Real time calculation of Signal to Noise ratio of the audio to detect bad audio quality and warn user for the same.
- Introductory Flash cards: Catchy flash cards on app startup stating the purpose of this android app so that users have an idea about the purpose of this android app.
- **Notifications:** Regular notification to app users as to have more daily active users.
- **Offers:** Offers section to reward user based on the quality and quantity of the data provided by them. This section lists out all the current live offers.

2.3 Design

2.3.1 Home Screen

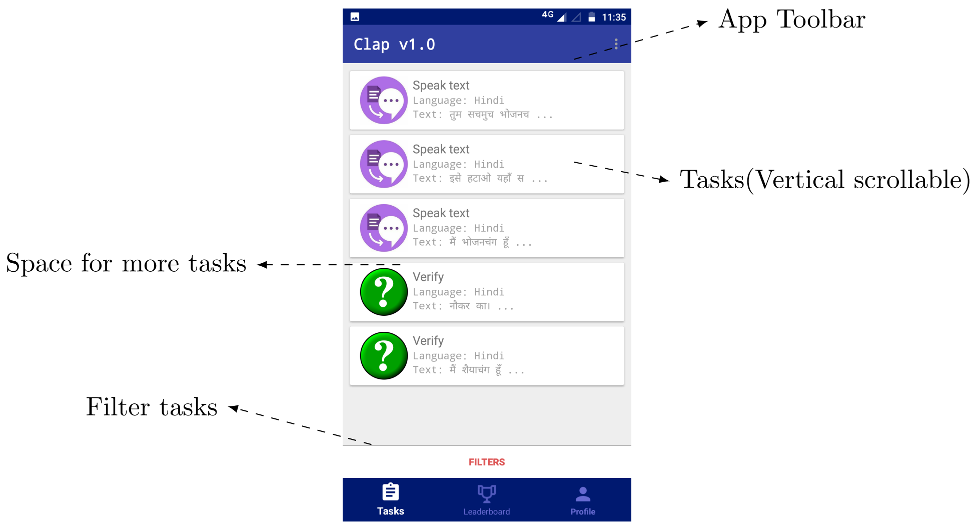

Figure 1: Home Screen

Home screen is divided into three parts: App toolbar(Consistent), Fragment(Inconsistent), Bottom Navigation Bar(Consistent) Consistent part means that the part doesn't change as the user navigates.The three dots on the right side of App toolbar popups menu for switching to Settings or Tutorial acrivity or to Logout.The middle fragment section switches between different User interfaces(Fragments) as the user navigates using the bottom navigation bar.Bottom navigation bar was choosen among other design choices because the three sections label, verify and profile are the most frequent visited section and hence they are one-click navigation.Other designs like side navigation bars required two click process.

2.3.2 Player

It is divided into two section.The upper section has all stuffs(buttons/edit areas) related to labelling of audio while the lower section comprises of media player functionality.The design choice of keeping large space between the two section eases access to both sections. When keyboard pops up, lower section shifts upward to help user type the conversion as well as control media playback within the half screen available to him.

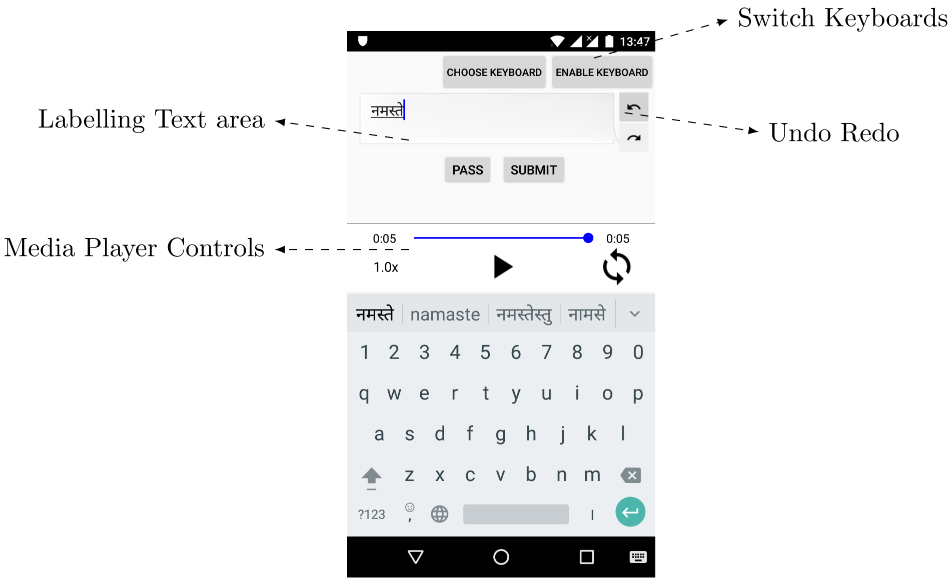

Figure 2: Player Screen

2.3.3 Profile

Profile Fragment at a high level has two sections.The upper section shows personal information of user.Design of this section is made such that the user can have its display picture as well as other personal information displayed in the first section.Latter section displays user performance reports in form of cards.This fragment is scrollable as to ease inclusion of more information about the user in near future.

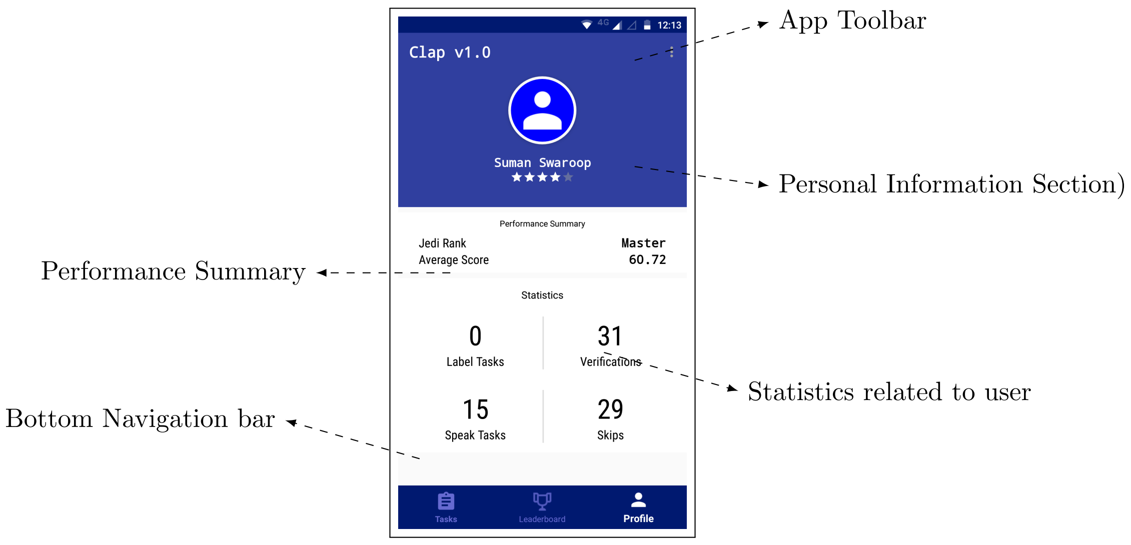

Figure 3: Profile

2.3.4 Speak task

Speak task requires users to speak the words written in text area box.To enhance user experience, user can playback recorded audio to check its quality. Warning pops up if user's recording is not good.

2.4 Architecture

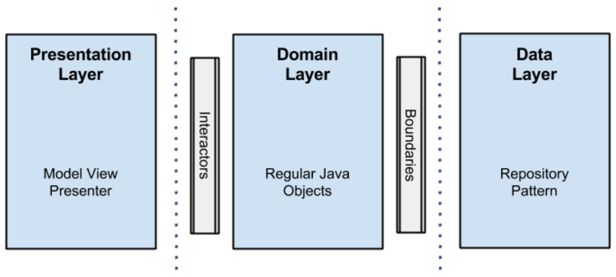

Figure 4: Clean Architecture Android

Clean Architecture used in this android app divides the complete work-flow of the app into 3 main layers.Higher layer depends upon the layer below it. They interact through well defined interfaces only.

2.4.1 Data Layer

This layer manages all tasks related to retrieval, storage and distribution of data to layer's above it.Data sources are Api, local database and shared preferences.ApiRepository in this layer makes the HTTP requests and the response is passed to layer above it via callbacks.

- Api: ApiInterface defines the retrofit interface to make HTTP calls. Api Endpoints have all the server endpoints.
- Model: It basically defines the models of different entities being used in app like App user, language list, Task etc.
- Repository: It is the interface through which the layer above it communicates the data layer. Divided into three parts: Api repository interface, Database interface, Shared Preferences interface.

2.4.2 Domain Layer

Pure logic are implemented here.This is a pure Java module without any android dependencies.Algorithmic tasks are carried out in this layer using the data obtained from the data repository.Currently no such heavy task exists in this app, so this layer is more or less used to pass data/instructions from layers above to below it and vice-versa using callbacks.

2.4.3 Presentation Layer

Model View Presenter architecture is implemented in this layer.Fragments and activities are only views, there is no logic inside them other than UI logic, and this is where all the rendering stuff takes place. Presenters in this layer are composed with interactors that perform the job in a new thread outside the main android UI thread, and come back using a callback with the data that will be rendered in the view.

2.5 To add new task:

Bottom up approach:

- Add all api endpoints in data layer. Add corresponding interfaces in Api Interface and Api Repository.Each task has a task model which is defined in model Task in data layer, define the task accordingly.
- Define Use cases of the task in domain layer
- Design the user interface in xml layout and code the corresponding behaviour in presentation layer. This layer is divided into three parts: adapters, fragments, activity.

3 Dashboard

The dashboard is designed for one-stop solution for admin to analyze data obtained by crowdsourcing. It allows admin to view data in well designed tabular and graphical manner. Current number of registrations, their contributions, tasks and their reviews and many more results are displayed in dashboard. It has following sections:

3.1 Overview

This sections plots the overall view of the current data collected. The admin can know about its audience i.e. in which language more people are contributing. It shows a plot of Number of users contributing in each language. It also displays a plot of how many users have contributed in each task type. This helps admin to boost some task, remove or redesign the others. Its equally important to know about the task distribution i.e. out of the total number of tasks completed till now, which task type is being done most by the users. To know about the most popular language, it is also important to know about the number of tasks completed in that language as well.

3.2 Tasks Section

3.2.1 Tasks table

This list out all tasks done by the users till now.Audio quality column reports cumulative review of that specific task. Each task is verified/reviewed by 3-4 users. Individual files can be downloaded on clicking audio id link. The admin can filter all tasks based on task language and audio quality review of the tasks. The table can be sorted based on any column by clicking on the top header row of the table.

3.2.2 Audio duration plot

This section also shows a plot of how many hours of data has been collected up till now for each language. It helps admin to boost some languages tasks and run tests on others.

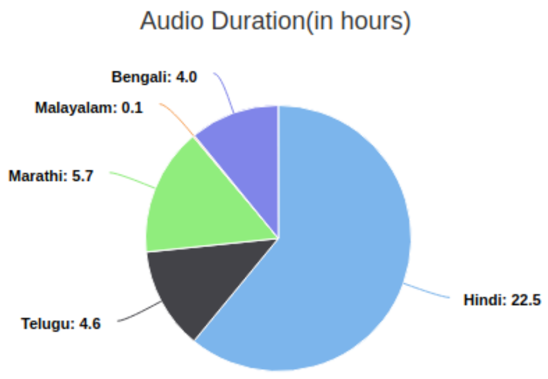

Figure 5: Audio durations

3.3 Users section

3.3.1 User Table

This section shows the tabular list of registered users with their count of different types of tasks completed by them. The table can be searched for specific users from their name and email id. The table can be sorted based on all columns by clicking on header of that column. Each user id is a clickable link that redirects to a personalised user page.

3.3.2 Personalised user section

In this the admin can know about a particular user's activity and performance. The pie chart plot shows the distribution of task types completed by the user. Admin can also analyze the number of tasks completed by the user in different languages. There is a speak task table section which lists out all the tasks done by the user. One important feature here to note is that the admin can calculate GOOGLE WER of all/few/random tasks done by the user. This is a cross verification feature by which admin can know about the quality of the user's tasks apart from the verification reviews alone. Similarly our trained machine can itself be used to cross verify new incoming tasks and filter out garbage data to good extent.

4 Pilot

After completion of our BTP I, a pilot was launched for our app. Using a form floated on GPO, around 141 users in the first phase, and 181 more users in the later phase signed up as volunteers. They were assigned tasks based on their language of preference, choosing from Hindi, Marathi, Telugu, Bengali and Malayalam. Initially, users were required to complete 60 Label and 60 Verify tasks in order to receive a reward - a Microsoft sponsored Tshirt. Later, we increased the number of tasks to 100 and 150 respectively. The increase was necessary because the amount of effort we predicted initially was less.

4.1 Results from the pilot

Number of registered users: 322

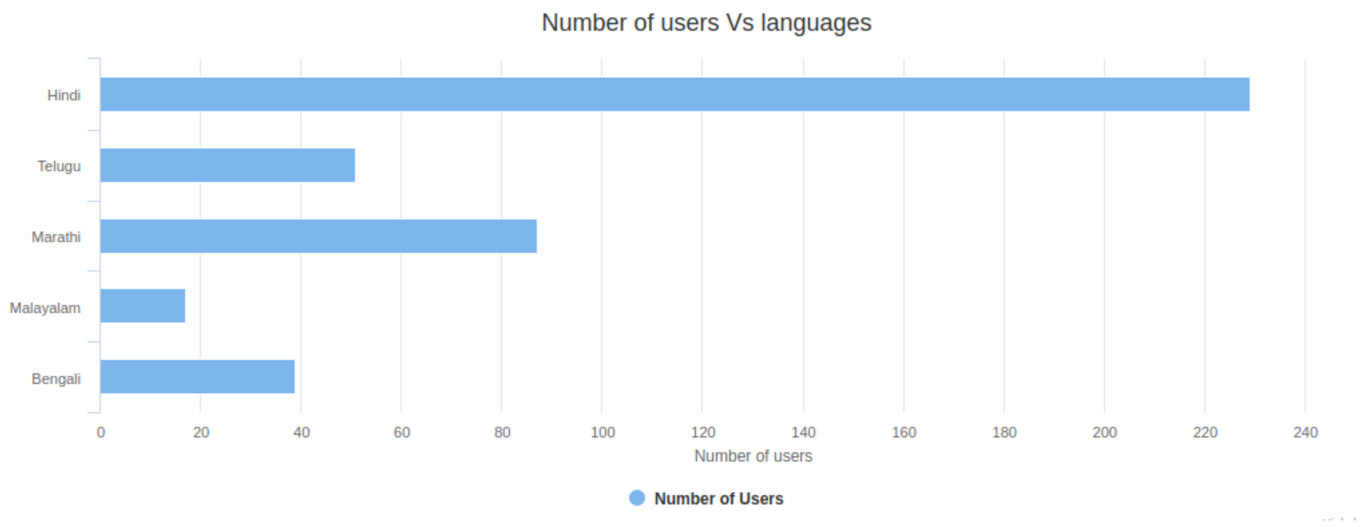

Figure 6: Number of users contributing in each language

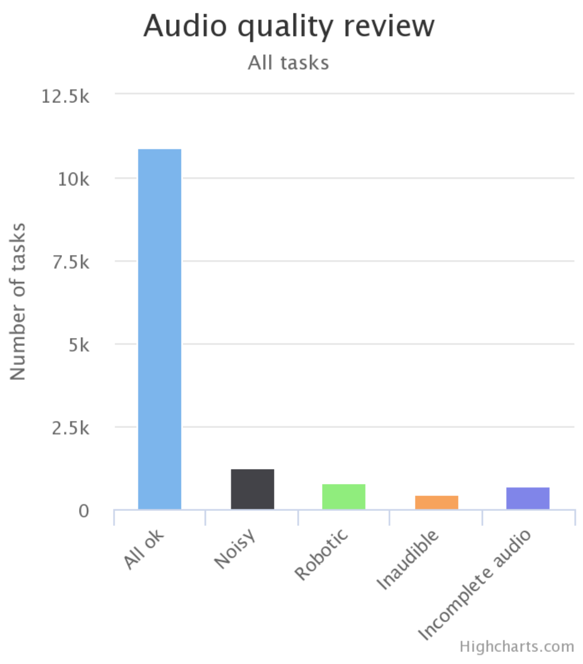

Figure 7: Audio Quality Review

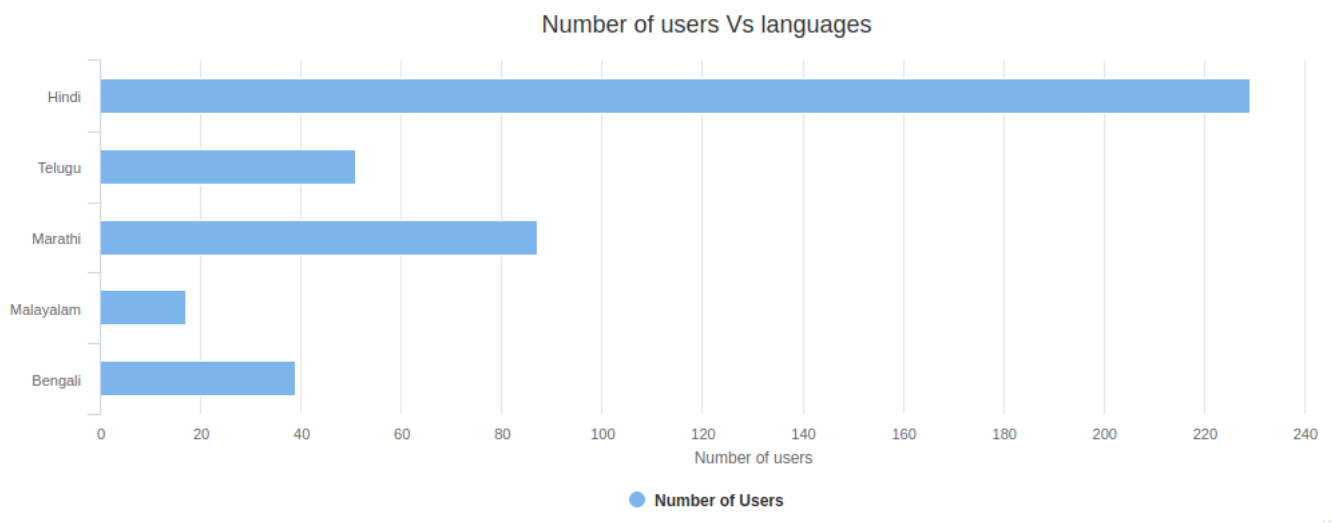

Figure 8: Number of users contributing in each language

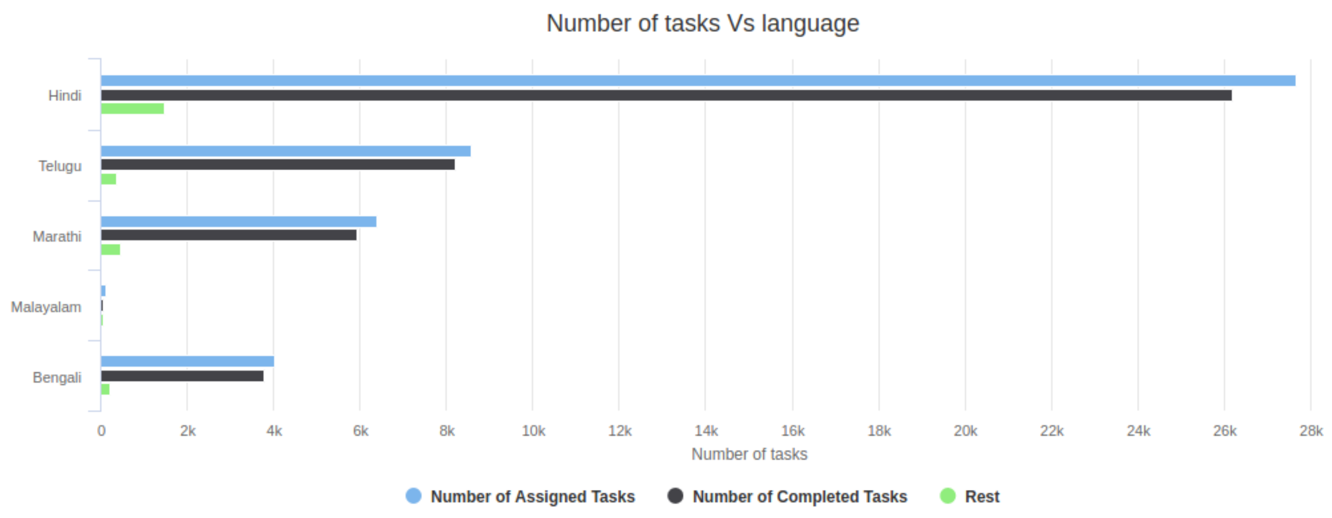

Figure 9: Number of tasks completed in each language

5 Evaluation

5.1 Kaldi

In order to evaluate the quality of our data, we decided to use the Kaldi-speech framework to train an Automatic Speech Recognition (ASR) model. We selected Hindi as the primary language, and divided over 18 hours of Hindi data into Train, Test and Validation sets. Kaldi ASR System gave a WER of 21

5.1.1 Comparison with Google's API

Google provides a Speech To Text API, we used the API to get all the transcribed text of the test data audio clips. We got a WER of 41% from the Google's data.Mainly there were substitution error with 4988 words wrongly substituted in Kaldi and 8506 words in Google. Results were better for kaldi because of the following reasons:

- The vocabulary of our model was small
- Test and train data were different but source was same
- Out of vocabulary word were only 599/5218 words
- Some transcriptions from our source, prathambooks.org were erroneous. So even if Google predicted those correctly, it will show up as wrong

POS Tag	Kaldi	Google
NN(Noun, Singular)	1387	3488
VM(Main Verb)	1113	1921
PSP(Post position)	614	723
PRP(Pronoun)	673	855
VAUX(Auxiliary Verb	644	825

Table 1: Number of words having error

To get a better understanding of the error, we tagged the data with parts of speech using the POS tagger and evaluated the number of words that are wrongly substituted for each POS tag. Both Kaldi and Google did not perform well for transcribing nouns.

6 Future Ideas

6.0.1 Android

- Story tasks: Continuous series of text to be spoken from a particular story targeted to a particular user.
- In-built Keyboard: Label tasks and editing in verify task are difficult due to non-availability of a language keyboard inbuilt in app. The app completely rely on external keyboard for such purpose.

6.0.2 Dashboard

- Kaldi WER for cross verification of speak tasks done by user
- Paytm wallet integration to reward users for quality work. Paytm API files are already included in project but further work is required after subscribing for a paytm merchant account.

References

[1] *Kaldi ASR*.

[2] *Highcharts for creating graphs*.

[3] *Stackoverflow*.

[4] *Official Django Developer Guide*.

[5] *Django REST Framework*.

[6] IIT Bombay. *POS Tagger*.

[7] Google Developers. *Google Cloud Speech Api*.