Scientific Journal Impact Factor (SJIF) : 5.278
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Speech Based Railway Query System

Abhijeet Chauhan[1], Narendra Kumar[2], Divyendu Shekhar[3], Abhinandan[4]

[1,2,4]*Computer Science & Engineering, The National Institute of Engineering Mysore*
[3]*Information Science & Engineering, The National Institute of Engineering Mysore,*

Abstract—Speech based railway query system is the railway enquiry system which acts based on the inputs provided by the user. No form of communication is better than voice. This autonomous system also uses our voice and provides the fastest and most authentic response. The best part of this system is that it can be easily installed at any of the railway stations. The convenience of using this system is that it works in bilateral and interactive manner in the form of speech. This system needs little to no human intervention and it requires very less maintenance and operating cost.

Keywords—Model-View-Controller Architecture, Speech-to-text Engine;

I. INTRODUCTION

In today's competitive era of rat race, importance of speech is paramount. Data can be transmitted at the rate of knots through the digital medium and can travel with supersonic speeds. Hence, there is a need for information inflow and outflow at a comparable pace too. Here is one such instance when we long for information flow at such unsurmountable speed. Most of us have experienced while waiting at railway station in long queues for the human controllers to get the information about the train timings. We have also encountered so many times that there will be no person for providing this information which significantly wastes our time. Here is one solution for such problems which minimizes human intervention in providing information at the railway station. Speech based railway query system is the enquiry system which operates based on the voice input given by the user.

II. SOLUTION DETAILS

2.1. Hardware Requirements

- Processor : i3 and newer.
- Monitor : SVGA and newer.
- RAM : 512 MB or more.
- Speed : 1.5 GHz or better.
- Secondary Device : Microphone

2.2. Software Requirements

- Rails 6.1
- Ruby 3.0
- PostgreSQL
- Custom Speech-to-text engine.

2.3. Related Concepts and Technologies

2.3.1. Ruby

Ruby is an interpreted, high-level, dynamic, reflexive, object-oriented, general purpose language. it had been designed and developed in mid - 1990s by Yukhiro "Matz" Matsumoto in Japan. Ruby was mostly influenced by Perl, Smalltalk, Eiffel, Ada and Lisp. Ruby supports functional, object-oriented and imperative paradigms. Also, Ruby has an automatic management of memory and is also a dynamic type system.

2.3.2. Ruby on Rails

Ruby on rails or just Rails, is a framework developed and designed under the MIT License and uses Ruby as its primary language. Rails may be a model–view–controller (MVC) framework, providing default structures for a database, a web service and sites. It encourages and facilitates the employment of web standards like JSON or XML for data transfer and HTML, CSS and JavaScript for display and user interfacing. additionally, to MVC, Rails emphasizes the employment of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and also the active record pattern.

2.3.3. Custom Speech to Text Engine

The Custom Speech-to-text engine is used to convert speech to text. The engine uses browser's ability to use to microphone to capture the voice. After the voice has been captured, it is fed to this engine which has a neural network at its core which uses various methodologies such as sampling and converts the speech to text.

2.3.4. MVC Architecture

Model–view–controller (MVC) is an architectural pattern used for deploying user interfaces on machines. It shards an application into three intertwined parts, so as to separate the representations of information from the ways it is accepted from the user. Normally in tradition, it was used for desktop graphical user interfaces (GUIs) but as days have passed, this architecture has become extremely popular for designing web applications.

III. DEEP DIVE

One may ask that speech recognition has been around for decades, so why is it just now hitting the main world? One of the possible arguments is that deep learning is now making speech recognition accurate enough to be used outside of carefully monitored environments. Andrew Ng already predicted that as and when speech recognition bumps from 95% to 99% accurate, it shall become a primary way that humans start interacting with the computers.

Since deep learning is involved, this system can reduce outside noise which the machine might encounter when it is being used by a person. There would be a lot of data going inside the system and hence this kind of processing is required to filter out white noise. There is a neural network present in between the machine when it takes the speech and converts it to text. There is also a problem related to speed. Some human being speaks faster and some slower. The work of neural network exemplifies here. In general, Sound waves are one dimensional and at any moment of time, it only has a single value based on the sound wave. This concept is called sampling and will be used to convert the waves into bits. This system uses Nyquist Theorem to reconstruct the original sound from the samples. The pre-processing of sampled sound data is then converted from the complex sound waves to simple sound waves using the well-known Fourier Transformation.

After the work of Neural Engine, text is produced from the speech. This text is then used into the system to interact with several railway-based APIs to return the result. The result is then parsed and provided to the user in a tabular format.

3.1 RESULTS

JSON parse while the speech is recorded and parsed to text via the custom engine:

Sent to Engine:

```
{
 "content-type": "audio/l16;rate=16000",
 "current_action": "start",
 "current_state": "listen"
}
```

Received from Engine:

```
{
 "current_index": 0,
 "final_results_output": [
  {
   "is_final_flag": true,
   "speech_converted": [
    {
     "converted_text": "Bangalore to Mysore"
    }
    ],
   }
  ]
}
```

converted_text from JSON is parsed and the two station names are separated.

Separated station names are then used and an API call is made to indianrailapi.com using the controller written in Ruby.

Controller:

RailwayController.rb

```
class RailwayController < ApplicationController
require ‘open-uri’
skip_before_filter :verify_authenticity_token
def index
end
```

Model:

StationName.rb

```
class StationName < ApplicationRecord
end
```

View:

Index.html.erb

```
<h2>Speech Based Query System</h2>
<h3>Text Area</h3>
<textarea id=”transcription” class=”materialize-textarea” readonly=”readonly”>
< /textarea>
```


Sample Response from API:

{
"ResponseCode":"200",
"Status":"SUCCESS",
"TotalTrains":"1",
"Trains":[
{
"TrainNo":"13507",
"TrainName":"ASN GKP EXPRESS",
"Source":"HJP",
"ArrivalTime":"02:55",
"Destination":"SV",
"DepartureTime":"00:25",
"TravelTime":"02:30H",
"TrainType":"EXP"
},
],
"Message":""
}

This Response is then parsed to provide the result in a tabular fashion.

IV. CONCLUSION

Speech based railway query system allows user to search train between specific time intervals by simply saying the query to the system. The result will be displayed according to the search in sorted order according to the departure time of the various trains from the start time till end time of the query. Speech based railway query system can also be installed with the same specifications in airport terminals or bus terminals. It solves the problem of waiting of passengers for enquiry and also less time consuming then waiting for the reply from the available enquiry booth.

REFERENCES

[1] http://www.speech-topics-help.com/self-introduction-speech.html

[2] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[3] http://rubyonrails.org/

[4] Alex Graves, Santiago Fern´andez, Faustino Gomez and J¨urgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," icml 2006, http://www.cs.toronto.edu/~graves/icml_2006.pdf

[5] Adam Coates, Deep Learning for Speech Recognition at the Bay Area Deep Learning School, https://www.youtube.com/watch?t=13874&v=9dXiAecyJrY&feature=youtu.be

[6] https://indianrailapi.com/api-collection/trains-between-station