Project Report: Speech to Text Order Assist

Rutu Nanavati, Viral Patel and Rishabh Agrawal

1 Summary

Natural language processing (NLP) is a form of artificial intelligence that focuses on analyzing human languages to draw insights, create advertisements, aid you in texting and more. The most difficult part of NLP is understanding or providing meaning to the natural language that the computer receives.

First, the computer must take natural language (humans speaking English/Spanish etc.) and convert it into artificial language. This is what is called speech recognition, or speech-to-text. Once the information is in text form, Next step is Natural Language Understanding(NLU) where you try to understand the meaning of that text.

In recent years, we have witnessed a revolution in the ability of computers to understand and to convert natural speech, especially with the application of deep neural networks (e.g., Google Home Mini, Alexa etc). But in particular these automated speech systems are still struggling to recognize simple words and commands. They don't engage in a conversation flow and force the user to adjust to the system instead of the system adjusting to the user.

The key focus of this project is to speed up the process flow of ordering at restaurants. This is especially useful in a drive-thru and take-out service to get through the queue quickly. Currently, a lot of resources and man-power are wasted by a person taking manual orders. Eventually, this could replace the person taking orders and save the money and time for both user and business end.

The Quick Service Restaurant (QSR) Assist aims to accomplish the above. It would start out by hearing the order, transcribe it to text, map it to the items in the menu and finally create an invoice. From the converted text we would be able to classify the intent of that part of the speech such as adding, deleting, modifying and canceling an order. The extracted information is then used for keyword extraction and invoice generation.

We would be using the google dataset Taskmaster-1 [7]. This dataset consists of 13,215 task-based dialogues in English. We will be using the conversation from only 2 of the 6 domains (ordering pizza and ordering coffee drinks) which are relevant to Quick Service Restaurants. Dialog annotations are based on the API calls associated with each type of task-based dialog. The full JSON description of the ontology can be found in ontology.json. Each conversation was annotated by two workers. Both annotations are included in this collection. The ontology.json can act as a reference to validate the invoice generated in the end. However, this does not validate intent classification. We are looking at various APIs that would help us test the intent classifications in further steps.

2 Methods

The Quick Service Restaurant (QSR) Assist will have the following key functionalities:

1. Speech Recognition

- The first step is to receive the audio input (16000 Hz) and store it in (.flac) file format. This will be achieved by creating a JavaScript audio recorder and save audio files using Django [2]. Currently we have used the audio files recorded through Audacity [3] software.

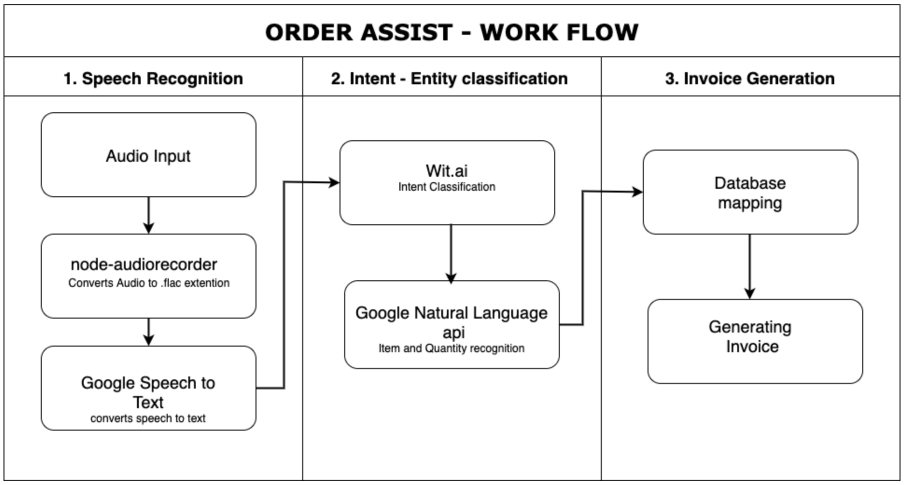

Figure 1: Work Flow Order Assist

Content limit	Audio Length
Synchronous Requests	∼ 1 Minute
Asynchronous Requests	∼ 480 Minutes
Streaming Requests	∼ 1 Minute

Table 1: API Requests

- Audio files are converted to text using Google Cloud Speech-to-text API client. Speech-to-Text API enables developers to convert audio to text by applying powerful neural network models in an easy-to-use API. The API recognizes 120 languages and variants that can support ones global user base. Speech-to-Text has ability to transcribe text from streaming audio or prerecorded audio in real-time.

  As described in Table 1, Google Speech to text has three types of API requests based on audio content. We are currently using Asynchronous request. However, we will be use Streaming Requests which is suitable for streaming data where the user is talking to microphone directly and needs to get it transcribed. This type of request is apt for building chatbots. As mentioned, the streaming data should be approximately a minute for this type of request.

  Input speech: ”Can I please get a Veggie Pizza and a coke bottle.”

  Transcribed to: ”Can I please get a Veggie Pizza and a coke bottle.”

- Cloud Speech-to-Text can return recognized text from audio stored in a file. It’s capable of analyzing short-form and long-form audio. Cloud Speech-to-Text is tailored to work well with real-life speech and can accurately transcribe proper nouns (such as, Viral Patel or Rutu Nanavati) and appropriately format language (such as, dates, phones numbers).

2. Intent-Entity Classification

- Intent-Entity Classification part of the pipeline is for understanding the intentions of humans and extract any relevant information to take some action.

- It uses the text transcribed to recognize the entity(object) and intent(action). eg: ”Please add olives to the pizza”.

  Entities: Pizza.

  Sub-Entities: Olives.

Intent: Customize

- For our scenario there are three main tasks for taking the order:
  - Item Recognition
  - Intent Classification
  - Quantity Recognition
- The object entity recognition is done to identify and classify items. This can be done by tokenizing and classifying the words. We used wit.ai [6] API to classify the intent (add, modify or update) of the conversation. Refer results from wit.ai [6] API from Appendix, Table 3: Intent Classification.
- Item and Quantity recognition is achieved by using natural-language [5] API by Google cloud. The Items are identified correctly from the statement along with the value. We are using entity analysis from the language module to perform this task. It identifies the value that is the entity and the type of entity. It can also recognize the quantity for the order.

3. Invoice Generation

- For invoice generation we are currently using a local database that stores the list of items associated with the price. The database can serve as a menu for offerings a restaurant has to offer. Entries from this database can be added, deleted or modified.
- The items recognized through the API from the conversation is queried in the local database to create the order and invoice. At this point, the order is mapped to the items in the menu by the unique identification code which would fetch prices and generate invoices in the end. The fetched prices are further passed through a simple python function that would calculate item-wise prices and total price overall. This can be given out in any format.Refer to Table 2, for results after the table has been mapped to the database.

3 Results

We ran our model for various different conversations.The results of one such conversation is explained below:

```
sentence = ['Can i order five Burger and fries',
            'I would like to order one large pizza and one soda',
            'Please remove the burger from my order',
            'I Would like to get a coffee too',
            'hi I would like to order one cheese burger,
            5 fries and 3 coke and 7 sandwich',
            'Remove one coke too']
```

On referring to Table 3, we observe most of the intents have been classified with a minimum of 0.70 confidence most of the intents have been classified over 0.95 confidence score using wit.ai [6]. Similarly item and quantity have been identified using these intents from natural-language API. The key-value pair of item and quantity at every step is given below as per the intent in each iteration.

```
{'burger': 5, 'fries': 1}
{'burger': 5, 'fries': 1, 'pizza': 1, 'soda': 1}
{'burger': 4, 'fries': 1, 'pizza': 1, 'soda': 1}
{'burger': 4, 'fries': 1, 'pizza': 1, 'soda': 1, 'coffee': 1}
{'burger': 4, 'fries': 6, 'pizza': 1, 'soda': 1, 'coffee': 1, 'coke': 3, 'sandwich': 7}
```

Item	Quantity	Price
burger	4	10.0
fries	6	4.5
pizza	1	10.0
soda	1	3.0
coffee	1	5.0
coke	2	3.0
sandwich	7	5.0

Table 2: Invoice

In Table 2, we observe the final list of item with their quantity and price associated with it. This could be sent to the person preparing the order. This table is further aggregated with price column aggregated to total price for that item accounting for quantity and total invoice price.

4 Discussion and Conclusion

Currently we are able to take the order end to end and generate an invoice. However, in the iteration steps there are few issues we need to keep in mind in order to obtain accurate values. We are able to correctly classify our main entities for most of the part however identifying sub entities is still an issues.

Also we can observe in the second last statement of sentence that from 'cheese burger' only 'burger' is identified but cheese isn't. So our biggest challenge for next steps is to identify sub-entities.Another challenge is that for multiple intent recognition, the client API is unable to identify multiple intents in a statement. We are planning to work on this part in the next phase of our project.

We also, observed that there were some speech to text errors while converting by Google API speech to text like misplaced punctuation or spelling errors. We plan to work on these results in the second phase of the project. As our intent classification is almost perfect we dont feel the need to train it more, so except that this are the issues we need to address going forward:

- Multiple Intent Classification
- Punctuations Handling in Speech to Text
- Identifying entities as multiple words together
- Identifying sub-entities
- Breaking down text received from Google API to exact sentences

5 Statement of Contributions

Rishabh Agrawal: Created 'MakeMenu' class, Intent Classification, Order generation, Report, Proposal, Presentation, SQL mapping, preliminary analysis.

Rutu Nanavati: Entity Analysis, Researched on APIs from Google and wit, Visualizations and flow diagram presentation for report, order aggregation presentation, proposal, report.

Viral Patel: Preliminary Exploration, Market Analysis, Entity recognition, Visualizations and work flow visualization for project, Report, presentation, proposal, researched on rasa NLU Pipeline.

References

[1] https://www.seelevelhx.com/2018-drive-thru-study-key-findings-full/

[2] https://github.com/voxy/django-audio-recorder/blob/master/README.md

[3] https://www.audacityteam.org

[4] https://ai.google/research/pubs/pub44631

[5] https://cloud.google.com/natural-language/

[6] https://wit.ai/

[7] https://ai.google/tools/datasets/taskmaster-1

[8] https://cloud.google.com/text-to-speech/

[9] https://www.topbots.com/ai-nlp-research-papers-acl2019/

6 Appendix

Github link for supporting codes: https://github.com/nanavatirutu/SpeechToTextOrderAssist

Sentence	Confidence Score	Intent
Can i order five Burger and a fries	0.7741	Add
I would like to order one large pizza and one soda	0.989	Add
Please remove the burger from my order	0.995	Remove
Can you swap the burger for a cheese pizza	0.987	Update
I would like to get a coffee too	0.9744	Add
That would be all	0.70	End

Table 3: Intent Classification

The GOOGLE NLP API was tested to get the intent and objects from some text orders. Here are those examples:

- **”One Burger and Two Fries”**

```
{
  "entities": [
    {
      "mentions": [
        {
          "text": {
            "beginOffset": 12,
            "content": "burgers"
          },
          "type": "COMMON"
        }
      ],
      "metadata": {},
      "name": "burgers",
      "salience": 0.58150464,
      "type": "OTHER"
    },
    {
      "mentions": [
        {
          "text": {
            "beginOffset": 24,
            "content": "fries"
          },
          "type": "COMMON"
        }
      ],
      "metadata": {},
      "name": "fries",
      "salience": 0.4184954,
      "type": "CONSUMER_GOOD"
    },
    {
      "mentions": [
        {
          "text": {
            "beginOffset": 7,
            "content": "five"
          },
          "type": "TYPE_UNKNOWN"
        }
      ],
      "metadata": {
        "value": "5"
      },
      "name": "five",
      "salience": 0.0,
      "type": "NUMBER"
    }
  ],
  "language": "en"
}
```

- **I want to order a burger, with coke and fries at the side**

```
        {
  "entities": [
    {
```

```
  "mentions": [
    {
      "text": {
        "beginOffset": 18,
        "content": "burger"
      },
      "type": "COMMON"
    }
  ],
  "metadata": {},
  "name": "burger",
  "salience": 0.36076498,
  "type": "OTHER"
},
{
  "mentions": [
    {
      "text": {
        "beginOffset": 31,
        "content": "coke"
      },
      "type": "COMMON"
    }
  ],
  "metadata": {},
  "name": "coke",
  "salience": 0.33710873,
  "type": "OTHER"
},
{
  "mentions": [
    {
      "text": {
        "beginOffset": 53,
        "content": "side"
      },
      "type": "COMMON"
    }
  ],
  "metadata": {},
  "name": "side",
  "salience": 0.18460932,
  "type": "OTHER"
},
{
  "mentions": [
    {
      "text": {
        "beginOffset": 40,
        "content": "fries"
```

```
          },
          "type": "COMMON"
        }
      ],
      "metadata": {},
      "name": "fries",
      "salience": 0.11751698,
      "type": "CONSUMER_GOOD"
    }
  ],
  "language": "en"
}
```