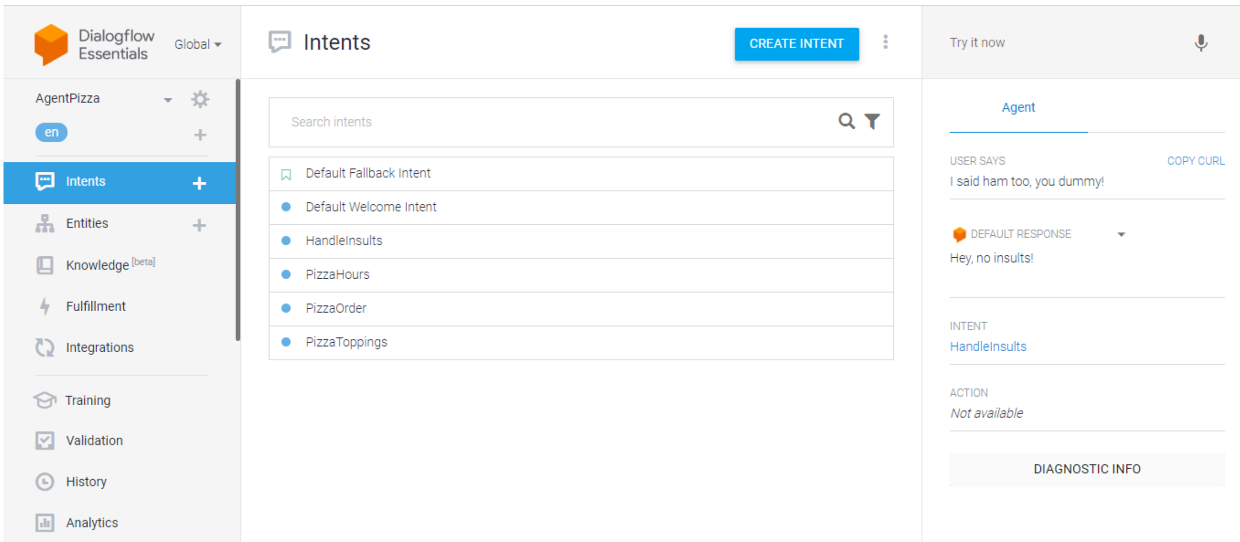
Samuel Peter May 18, 2024

Professor Rolando Coto-Solano COSC 72: Accelerated Computational Linguistics

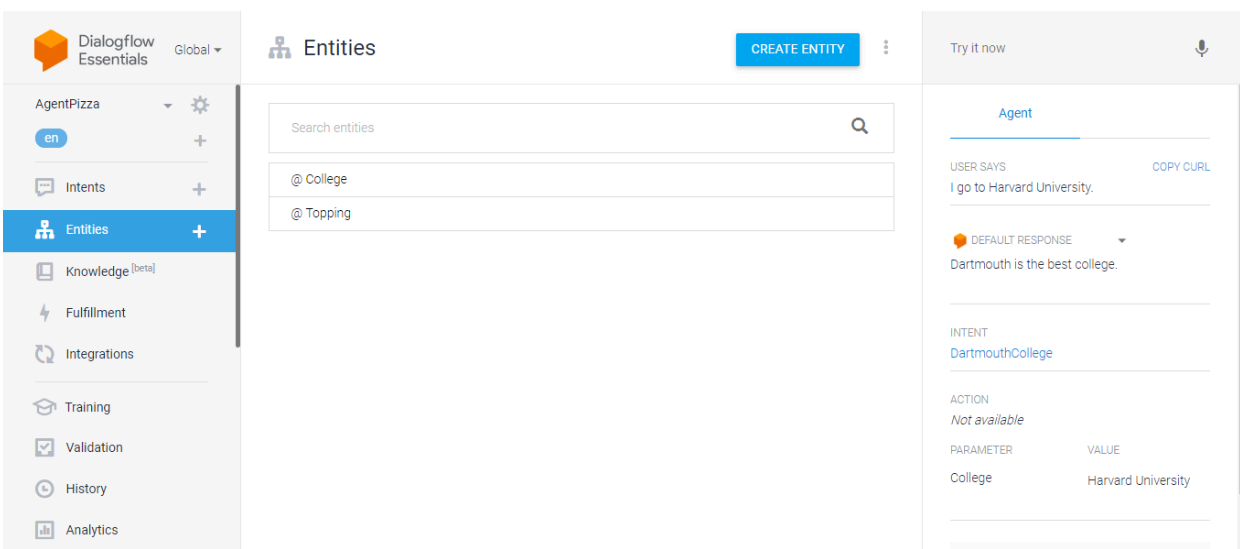
# Information Extraction and Speech Recognition

## 1. Pizza Chat (Dialogueflow Chat)

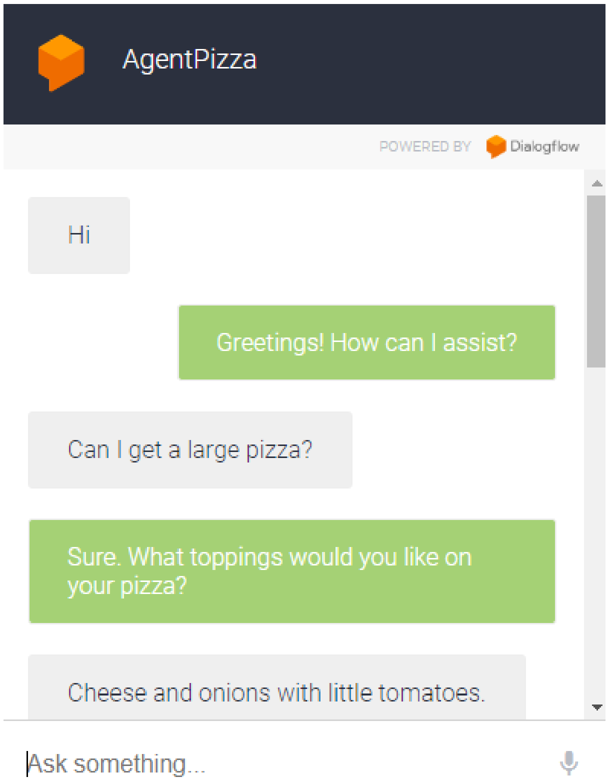
a. Screenshot of Intents:

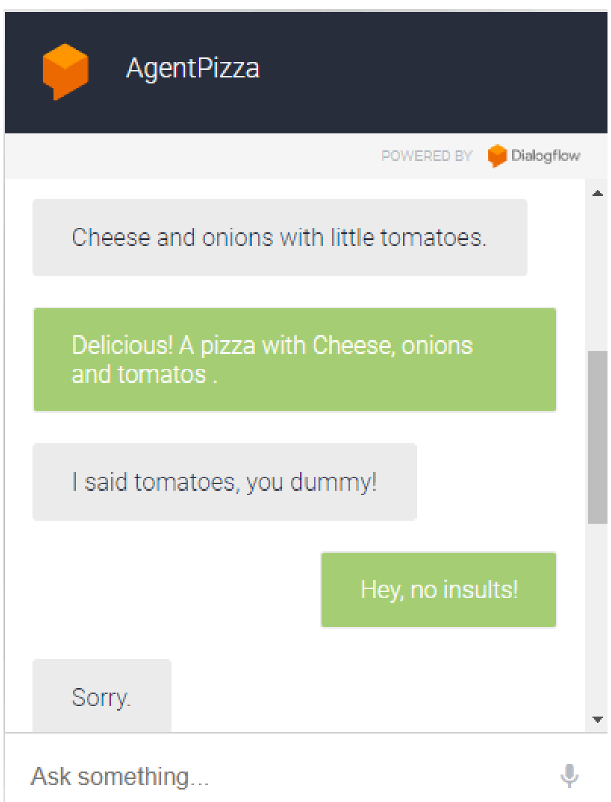


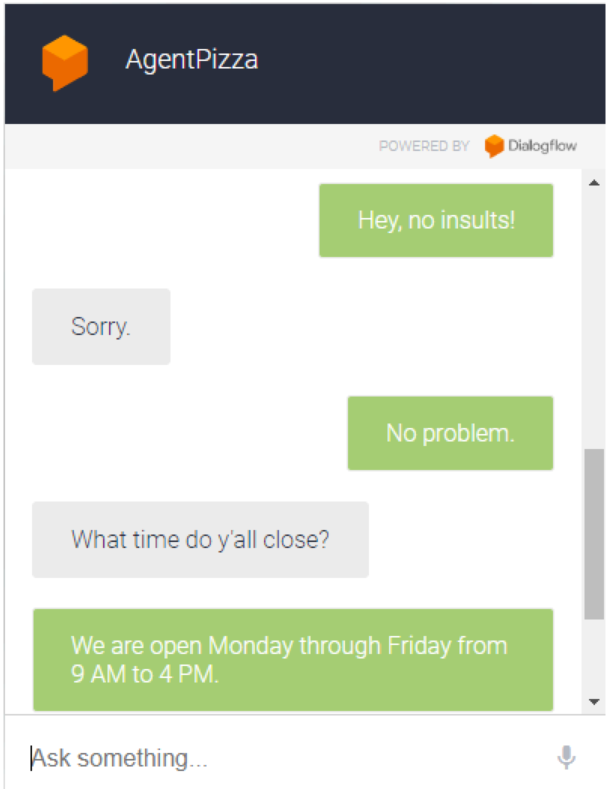

b. Screenshot of Entities:

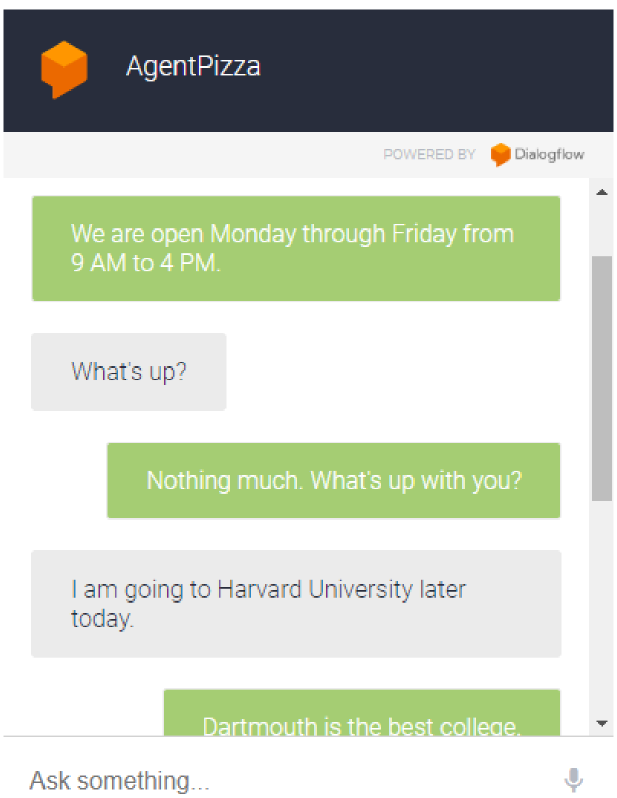


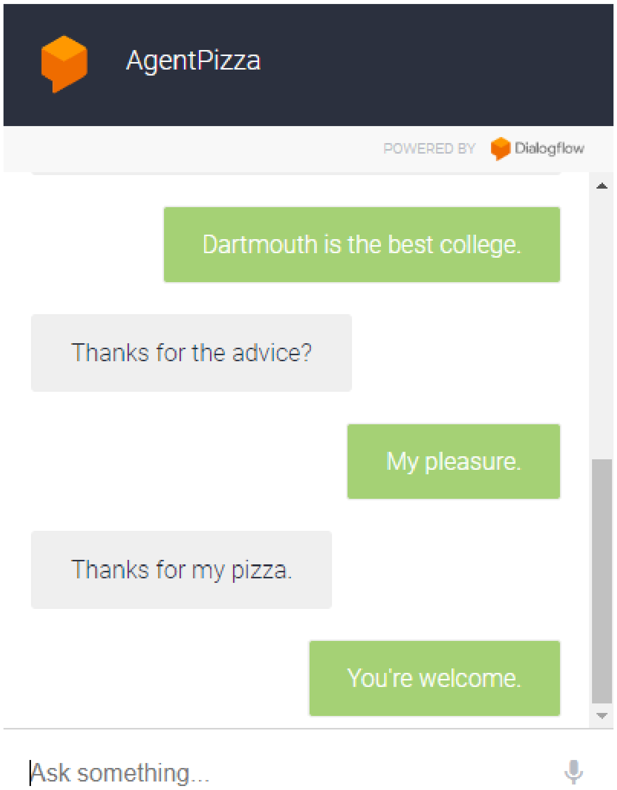

c. Screenshot of Assistant Preview:


AgentPizza
POWERED BY Dialogflow
Hi
Greetings! How can I assist?
Can I get a large pizza?
Sure. What toppings would you like on your pizza?
Cheese and onions with little tomatoes.
Ask something...


AgentPizza
POWERED BY Dialogflow
Cheese and onions with little tomatoes.
Delicious! A pizza with Cheese, onions and tomatos .
I said tomatoes, you dummy!
Hey, no insults!
Sorry.
Ask something...


AgentPizza
POWERED BY Dialogflow
Hey, no insults!
Sorry.
No problem.
What time do y'all close?
We are open Monday through Friday from 9 AM to 4 PM.
Ask something...


AgentPizza
POWERED BY Dialogflow
We are open Monday through Friday from 9 AM to 4 PM.
What's up?
Nothing much. What's up with you?
I am going to Harvard University later today.
Dartmouth is the best college.
Ask something...




d. Agent Pizza URL for testing: https://bot.dialogflow.com/570d403f-0c70-41ff-9f47-914da89b6cfe

## 5. Errors in End-to-End Speech Recognition Model:

1. English speech recognition:
   The algorithm correctly identifies that the language spoken is English. Besides my name, the program was able to accurately transcribe the audio file. My name was transcribed as Samuara, instead of Samuel. This error could have come about due to the presence of background noise that my microphone was unable to cut out.

   Screenshot of English speech transcription:

```
[7]
_, probs = model.detect_language(mel)
print(f"Detected language: {max(probs, key=probs.get)}")

options = whisper.DecodingOptions(language="en", without_timestamps=True, fp16 = False)
result = whisper.decode(model, mel, options)
print(result.text)
```

```
Detected language: en
Hello, I am Samuara. In my free time, I love to play football. Messi is my favorite player and Manchester City is my favorite club. I am travelling to Prague in the spring and I cannot wait to
```

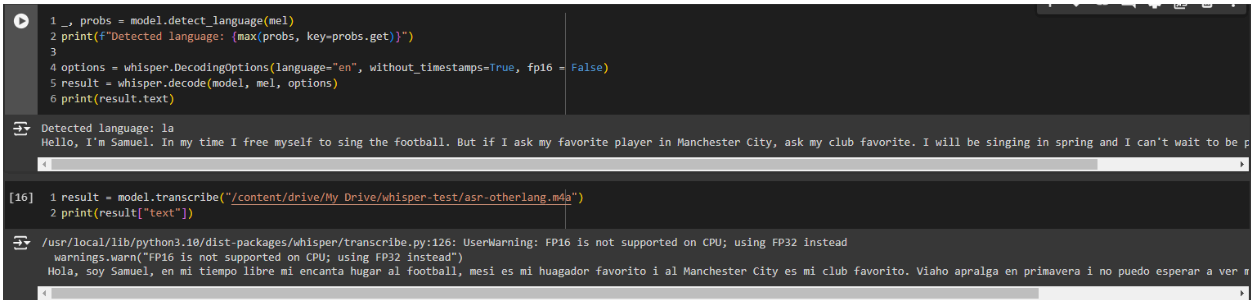
```
result = model.transcribe("/content/drive/My Drive/whisper-test/asr-test.m4a")
print(result["text"])
```

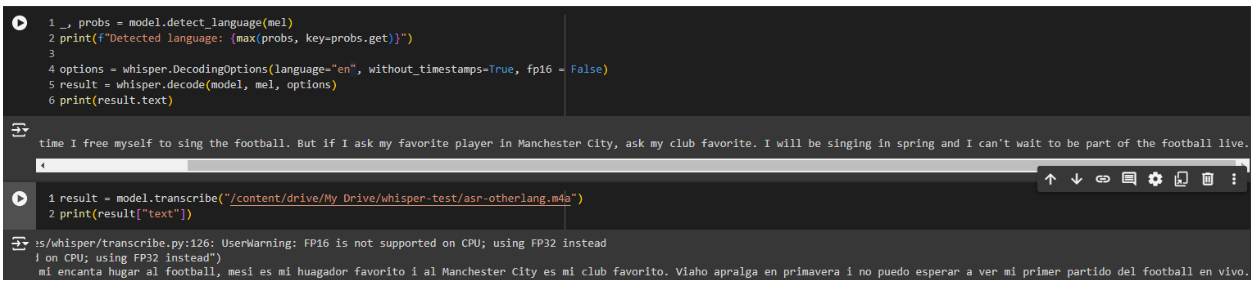
```
/usr/local/lib/python3.10/dist-packages/whisper/transcribe.py:126: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
 Hello, I am Samuara. In my free time I love to play football. Massey's my favorite player and Manchester City is my favorite club. I am travelling to Prague in the spring and I cannot wait to
```

2. Spanish speech recognition:
   The program accurately identifies the other language spoken as Spanish. There were a couple of errors in the transcription process. I do have to admit that my Spanish pronunciation is not the best, so I might partly have contributed to the transcription errors. Here is a list of errors I observed:
   a. Jugar vs Hugar: Jugar(play) is identified as hugar, which is not a word in the Spanish language. The English transcription of the sentence says I like to sing about football, despite there being no occurrence of singing in the Spanish transcription of the sentence. This issue shows that the algorithm is primarily created for the English language instead of Spanish because the h sound in Spanish is represented by the letter j. The absence of the letter j could have led to an incorrect recognition of the word, which would then lead to an incorrect transcription and translation.
   b. Similarly, my sentence about traveling to Prague in the spring was transcripted to: I will be singing in spring and I can't wait to be part of the football live. The Spanish transcription of the audio(Viaho apralga en primavera i no puedo esperar a ver mi primer partido del football en vivo), though slightly off, is accurately translated meaning-wise by Google Translate to ‘It's spring time and I can't wait to see my first live football game’. This could be attributed to Google’s superior translation algorithm and its ability to identify the user’s intent despite the presence of typos.
   c. My sentence ‘Messi es mi juego favorito y el Manchester City es mi club favorito.’ was identified fairly accurately and transcripted as ‘mesi es mi huagador favorito i al Manchester City es mi club favorito.’ The English translation of this was hard to understand. It read ‘But if I ask my favorite player in Manchester City, ask my club favorite.’ This issue could be attributed to the fact that most speech

recognition/translation algorithms are catered toward English, hence struggling to correctly translate/transcribe non-English languages.

Screenshots of Spanish speech transcription:




3. CTC Algorithm Summary:

The Connectionist Temporal Classification (CTC) algorithm is designed for sequence-to-sequence tasks, such as speech recognition, in which the input and output sequences have varying lengths and alignments. It is composed of several key components and algorithmic steps.

The CTC algorithm utilizes a blank token to facilitate non-output steps, aiding in aligning the input and output sequences. It calculates the probabilities of all potential input-output alignments, allowing for flexibility in sequence mapping. The collapse function within the CTC algorithm transforms paths into the final output sequence by eliminating consecutive duplicates and blank tokens, resulting in a more refined output sequence.

The algorithm proceeds through several essential steps, including the forward-backward algorithm, which computes the probabilities of partially and fully producing the output sequence at each time step. Following this, the loss calculation aggregates the probabilities of all valid paths for the correct output sequence. Finally, during the training phase, the algorithm employs the gradients from the probability calculations to update model parameters.

CTC offers several advantages, such as the ability to train on raw, unaligned data without requiring pre-alignment. Additionally, it effectively handles variable lengths, making it suitable for tasks with differing input and output lengths.

This algorithm finds applications in speech recognition, where it converts audio signals to text, as well as in handwriting recognition, where it translates pen strokes or images to text. In summary, CTC enables neural networks to learn the correct sequence mapping by considering all potential alignments, making it particularly well-suited for tasks such as speech and handwriting recognition.