Using Deep learning to save an endangered language

New Mexico
Supercomputing Challenge
Final Report
April 10, 2024

V. Sue Cleveland High

Team Members:
Sosa Alexandra
Jack Arnadavile

Teachers:
Ashli Knoell

Summary

Language is an expressive instrument for culture that connects us to the world and the ones around us spiritually and emotionally. Language is a tool that allows us to trade information, create friendships and share the past. Language and culture are inextricably related. And you can't learn one without first knowing the other. And because of this when we lose our language we also lose a part of our culture. Learning simple conversation in a dying language is enough to keep it alive and use it to expand and learn more about. In this project, we are teaching and refreshing the user on basic Navajo through simple conversation and using commands to get the computer to complete a task. A model such as the one we have created will help young Navajo children and adults learn their language and keep the Navajo language from dying.

Introduction

1.1 Loss of a language

Before 1492 there were over 300 Native languages spoken across North America and now there's around 170. Currently 75 native languages are nearing extinction with the Navajo language being one of them, despite being the most commonly spoken native language. If this continues 75 different tribes will lose part of their history, culture, religion, and connection to their past.

The Navajo language today has only 170,000 speakers and in 2050, it is estimated to have 20 to 0 speakers. Most indigenous languages have even fewer speakers like the Towa language with 3,000 speakers, the Ahtna language with 15 speakers and the Caddo with 2 speakers.

Using a deep language learning model that can understand parts of the Navajo language and perform an action according the word spoken, can help bring the language back because it can be used more and as a tool for anyone

1.2 The method

When we began working on our project we needed to understand how this project could help someone learn Navajo. We came upon the idea of making a model where you speak Navajo commands to your computer to act as an assistant. According to Joanne Lewsley “When your baby is about 12 months to 15 months old he'll understand simple requests, such as "stop", "give it to daddy", or "come here".” Some of the first words we learn in our language are basic commands. That gave us the idea of teaching navajo by teaching the user to say commands.

1.3 The Goal

With the increasing amount of native americans losing their ability to speak their language we hope that we can keep the navajo language alive by teaching it through our devices.

The Process

1.1 Errors

Upon building the model we encountered many issues. We thought the error was due to the exact features function because we set the wrong parameters but no matter how we changed this function we got the same issue. We eventually solved this issue later on.

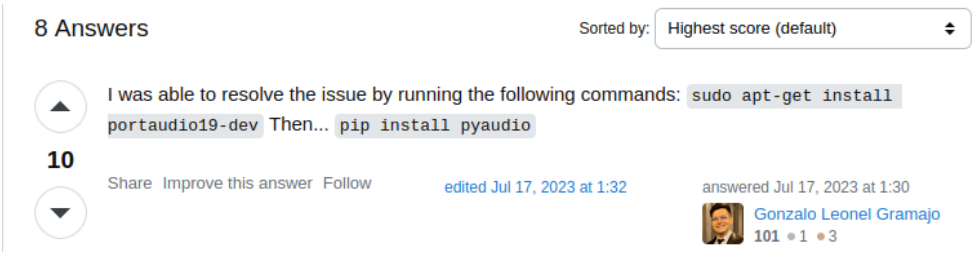
```
12
13 def extract_features(file_path, duration=3):
14     y, sr = librosa.load(file_path, duration=duration, sr=None)
15     n_mels = 128
16     mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=n_mels)
17     mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)
18
19     target_shape = (n_mels, int(sr * duration / 512) + 1)
20
21     mel_spec_resized = np.resize(mel_spec_db, target_shape)
22
23     return mel_spec_resized
24
```

Another issue we encountered was getting the program to recognize Tensorflow. In the terminal we would often get errors saying that it couldn't find Tensorflow. A lot of the issues we encountered were because we didn’t know how to install libraries and such, properly.

When we had issues with our code that we didn’t understand, we would use google and often relied on stack overflow and the people in that website to provide us with solutions. Sometimes we would use Chatgbt to explain why an issue is occurring if google couldnt provide a solution to our problem.

1.2 Learning

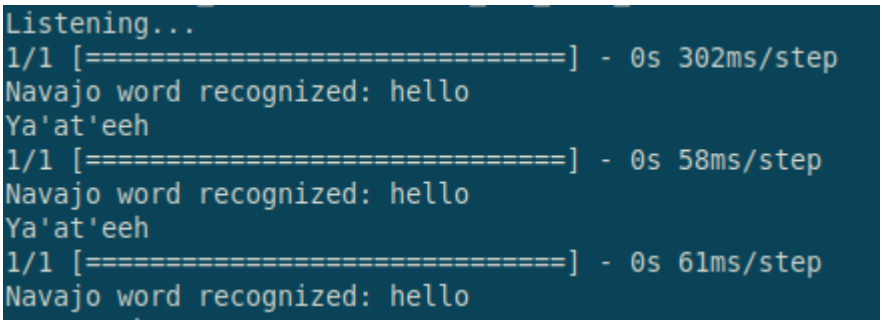
Having very little knowledge on deep learning, this project took a lot of searches and informative videos to get us to where we are today. Our mentor was able to give us some information and resources to help us start out the project but unfortunately we lost the ability to contact him so we relied on the internet to be our teacher. We relied heavily on google and youtube to teach us how to code.



Conclusion

1.1 The Results

In conclusion our model was not as successful as we wanted because it understood only one of our Navajo words. We believe that there might be a data imbalance or not enough data that's causings the model to be trained insufficiently.



1.2 Future plans

We intend to continue working on this project and expand our data. We hope that this project can be shared with everyone and inspire other people to help keep the Navajo language alive.

We would also want to include more spoken commands and have our program speak back to us. We currently have seven commands now but we would like to expand on our commands and even include a way to have a simple conversation with the computer.

Lastly, we want to include other indigenous languages that are going extinct and help keep their language and culture alive. We hope we can share this project online and have other people contribute their data so that we can help save other indigenous languages.

1.2 Products of our work

This project was coded in python and we used several python libraries for audio processing and machine learning tasks such as Librosa for audio analysis, Pandas for data manipulation, Scikit-learn for machine learning and data analysis, Tensorflow for its open source machine learning framework, and Keras for API.

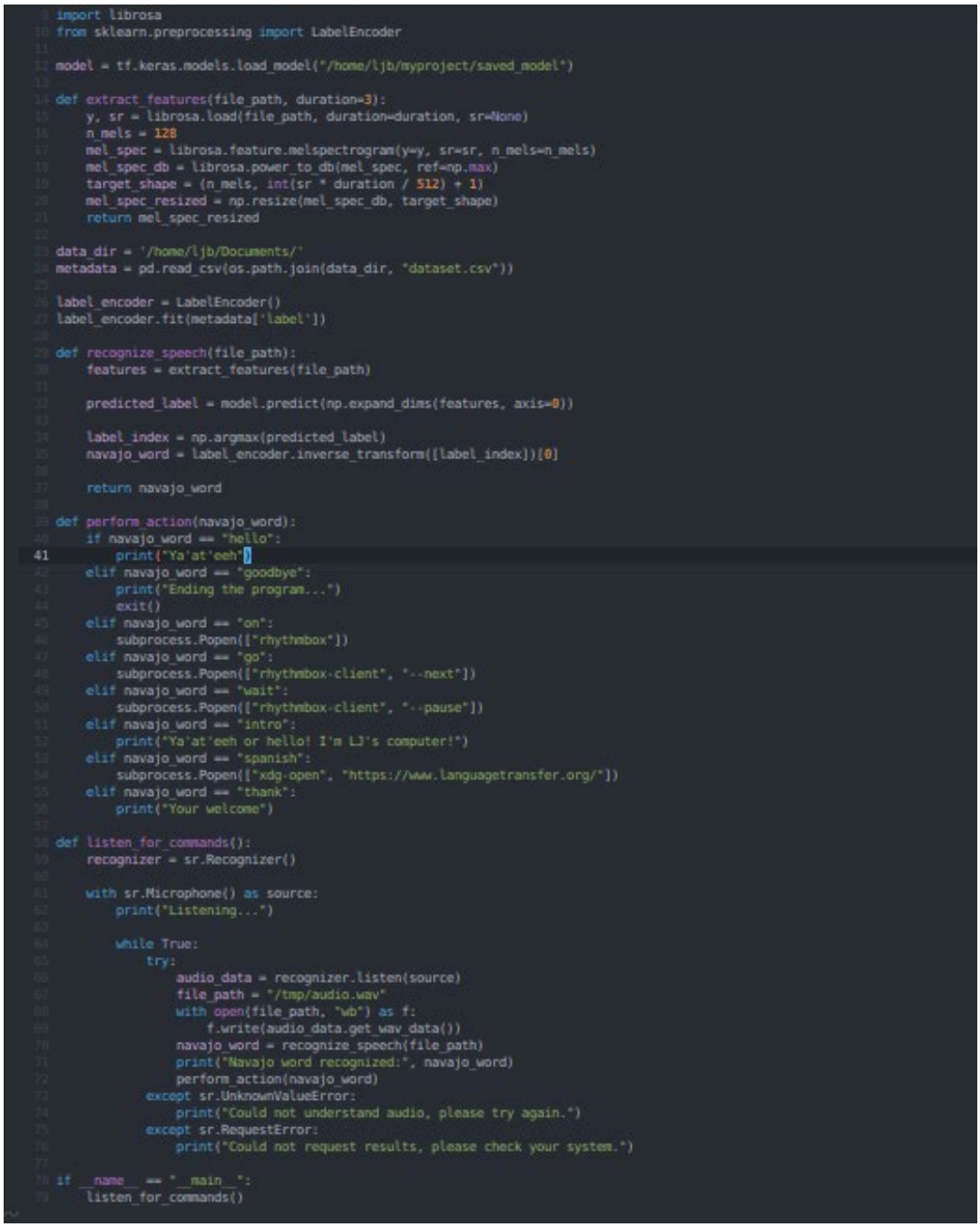
```python
#!/usr/bin/env python3

import os
import numpy as np
import pandas as pd
import librosa
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D

def extract_features(file_path, duration=3):
    y, sr = librosa.load(file_path, duration=duration, sr=None)
    n_mels = 128
    mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=n_mels)
    mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)

    target_shape = (n_mels, int(sr * duration / 512) + 1)

    mel_spec_resized = np.resize(mel_spec_db, target_shape)

    return mel_spec_resized

data_dir = '/home/ljb/Documents/'
metadata = pd.read_csv(os.path.join(data_dir, "dataset.csv"))

metadata = metadata.dropna(subset=['file_path'])

X = np.array([extract_features(os.path.join(data_dir, file), duration=3) for file in metadata['file_path']])

label_encoder = LabelEncoder()
y = to_categorical(label_encoder.fit_transform(metadata['label']))

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(X.shape[1], X.shape[2], 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(y.shape[1], activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=32)

model.save("/home/ljb/myproject/saved_model")
```

This is the code used to train the model

1.3 Acknowledgments

Even though this project wasn't very successful we got to learn a lot about python, frameworks, and machine learning. Though our model was only able to recognize one word we are very grateful to the people who helped us such as our

teacher Ashli Knoell and mentor who we unfortunately lost contact with. Thank you judges for giving us feedback and even helpful info when we were stuck.



```
import librosa
from sklearn.preprocessing import LabelEncoder

model = tf.keras.models.load_model("/home/ljb/myproject/saved_model")

def extract_features(file_path, duration=3):
    y, sr = librosa.load(file_path, duration=duration, sr=None)
    n_mels = 128
    mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=n_mels)
    mel_spec_db = librosa.power_to_db(mel_spec, ref=np.max)
    target_shape = (n_mels, int(sr * duration / 512) + 1)
    mel_spec_resized = np.resize(mel_spec_db, target_shape)
    return mel_spec_resized

data_dir = '/home/ljb/Documents/'
metadata = pd.read_csv(os.path.join(data_dir, "dataset.csv"))

label_encoder = LabelEncoder()
label_encoder.fit(metadata['label'])

def recognize_speech(file_path):
    features = extract_features(file_path)

    predicted_label = model.predict(np.expand_dims(features, axis=0))

    label_index = np.argmax(predicted_label)
    navajo_word = label_encoder.inverse_transform([label_index])[0]

    return navajo_word

def perform_action(navajo_word):
    if navajo_word == "hello":
        print("Ya'at'eeh")
    elif navajo_word == "goodbye":
        print("Ending the program...")
        exit()
    elif navajo_word == "on":
        subprocess.Popen(["rhythmbox"])
    elif navajo_word == "go":
        subprocess.Popen(["rhythmbox-client", "--next"])
    elif navajo_word == "wait":
        subprocess.Popen(["rhythmbox-client", "--pause"])
    elif navajo_word == "intro":
        print("Ya'at'eeh or hello! I'm LJ's computer!")
    elif navajo_word == "spanish":
        subprocess.Popen(["xdg-open", "https://www.languagetransfer.org/"])
    elif navajo_word == "thank":
        print("Your welcome")

def listen_for_commands():
    recognizer = sr.Recognizer()

    with sr.Microphone() as source:
        print("Listening...")

        while True:
            try:
                audio_data = recognizer.listen(source)
                file_path = "/tmp/audio.wav"
                with open(file_path, "wb") as f:
                    f.write(audio_data.get_wav_data())
                navajo_word = recognize_speech(file_path)
                print("Navajo word recognized:", navajo_word)
                perform_action(navajo_word)
            except sr.UnknownValueError:
                print("Could not understand audio, please try again.")
            except sr.RequestError:
                print("Could not request results, please check your system.")

if __name__ == "__main__":
    listen_for_commands()
```

This is the

code that uses that performs commands based on the commands spoken

Citations

babycentre. “Developmental milestones: your baby's understanding.” *BabyCentre*, https://www.babycentre.co.uk/a6575/developmental-milestones-your-babys-understanding. Accessed 11 April 2024.

Wikipedia, https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiUgbTj6L2EAxVFLEQIHb3UCvkQFnoECBMQAw&url=https%3A%2F%2Fwww.acf.hhs.gov%2Farchive%2Fana%2Fpreserving-native-languages-article&usg=AOvVaw0-8ptgfFjzhPXGgJ3SPKmg&opi=89978. Accessed 22 February 2024.

Wikipedia, https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwji-8-h7b2EAxXxLkQIHc1VAO4QFnoECBEQAw&url=https%3A%2F%2Fgreenheart.org%2Fblog%2Fgreenheart-international%2Flanguage-the-essence-of-culture%2F&usg=AOvVaw2THAMEQPWhlR-nZ3. Accessed 23 February 2024.

Wikipedia, https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwioyujL6r2EAxW3L0QIHcWfDooQFnoECA8QAw&url=https%3A%2F%2Fwww.worldatlas.com%2Farticles%2Fendangered-languages-of-the-united-states.html&usg=AOvVaw2fNQIcgwwy_SdXjGNunPw2&. Accessed 23 February 2024.

“Ahtna language.” Wikipedia, https://en.wikipedia.org/wiki/Ahtna_language. Accessed 23 February 2024.

“Caddo language.” Wikipedia, https://en.wikipedia.org/wiki/Caddo_language. Accessed 23 February 2024.

“Language And Culture Relationship - A Detailed Guide - TLD.” The Language Doctors, 2 April 2021, https://thelanguagedoctors.org/blog/what-is-the-language-and-culture-relationship/. Accessed 22 February 2024.