University of North Texas

CSCE 5280 AI for wearables

Dialog act aware conversation response generator

Project 1

Shou-Jen Wang

Eric Hedrick

Ashwin Thota

Setup

speech_recognition tensorflow nltk

The probablistic model developed from https://github.com/NathanDuran/Probabilistic-RNN-DA-Classifier (https://github.com/NathanDuran/Probabilistic-RNN-DA-Classifier) will be used to make prediction

In [1]:

```
#Speech to text
import speech_recognition as sr
print(sr.__version__)

#Dialog act classification
import os
import tensorflow as tf
#from tensorflow import keras
from tensorflow.keras.models import load_model

model_dir = 'models/'
model_name = 'Probabilistic Model'

num_epoch = 10
hidden_layer = 128

model_name = model_name + " -" + \
             " Epochs=" + str(num_epoch) + \
             " Hidden Layers=" + str(hidden_layer)

p_model = load_model(model_dir + model_name + '.hdf5')

#to preprocess data the same way, the utilities.py file from the prior project need
s to be imported
from utilities import *

resource_dir = 'data/'
embeddings_dir = "embeddings/"
embedding_filename = 'word2vec_swda'
model_dir = 'models/'
model_name = "Embeddings Model"

# Load metadata
metadata = load_data(resource_dir + "metadata.pkl")
word_frequency = 2
frequency_data = load_data(embeddings_dir + 'probabilistic_freq_' + str(word_freque
ncy) + '.pkl')


#to tokenize the input
import nltk

# compile the model
from tensorflow.keras.optimizers import RMSprop
learning_rate = 0.001
optimizer = RMSprop(lr=learning_rate, decay=0.001)

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

# response - did not have time to develop fully
import pandas as pd

df = pd.read_csv('dialog_act_table.csv', skip_blank_lines = True)
da_tbl = df.dropna()
```

```
3.8.1
Loaded data from file data/metadata.pkl.
Loaded data from file embeddings/probabilistic_freq_2.pkl.
```

Run as many times from here

In [2]:

```python
r = sr.Recognizer()
#define microphone object for microphone input
mic = sr.Microphone()

#microphone records
#after a period of time with no input, it will stop listening automatically
try:
    print("Listening")
    with mic as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
except KeyboardInterrupt:
    pass
```

```
Listening
```

In [3]:

```python
#application of speech recognition using google speech api
input_text = r.recognize_google(audio)

print(input_text)
```

```
I am a graduate student at the University of North Texas
```

Preprocess input data

In [4]:

```python
print(input_text)

#Convert to same format for input
utterances = []
labels = []
utterances.append(nltk.word_tokenize(input_text))
#set a default label for processing
labels.append('%')

# Save input to same data structure
data = dict(
    utterances=utterances,
    labels=labels)

#save_data(resource_dir + "input" + "_data.pkl", data)
print(data)
```

```
I am a graduate student at the University of North Texas
{'utterances': [['I', 'am', 'a', 'graduate', 'student', 'at', 'the', 'University
', 'of', 'North', 'Texas']], 'labels': ['%']}
```

In [5]:

```python
# generating probabilistic embeddings in the same format as training
ins_x, ins_y = generate_probabilistic_embeddings(data, frequency_data, metadata)
```

In [6]:

```
result=p_model.predict(ins_x,batch_size=100, verbose=1)

# generating predictions.
index_to_label = metadata["index_to_label"]
prediction=index_to_label[np.argmax(result)]
print(prediction)
```

```
1/1 [==============================] - 0s 1ms/step
sd
```

In [7]:

```
# determine if the dialog act should generate a response

row = da_tbl[da_tbl.columns[1]]==prediction
response = da_tbl.loc[row]['Response']
#print(response.dtype)
reply = response.iloc[0]
print(reply)
```

```
no
```

In [8]:

```
print("Input: " + input_text)
print("Prediction: " + prediction)
print("Generate Reply:" + reply)
```

```
Input: I am a graduate student at the University of North Texas
Prediction: sd
Generate Reply:no
```

In [ ]: