Watson Speech to Text Translator

Estimated time needed: **25** minutes

Objectives

After completing this lab you will be able to:

- Create Speech to Text Translator

Introduction

In this notebook, you will learn to convert an audio file of an English speaker to text using a Speech to Text API. Then you will translate the English version to a Spanish version using a Language Translator API. Note: You must obtain the API keys and enpoints to complete the lab.

Table of Contents

[ ]:
```
#you will need the following library
!pip install ibm_watson wget
```

Speech to Text

First we import `SpeechToTextV1` from `ibm_watson` .For more information on the API, please click on this link

[ ]:
```
from ibm_watson import SpeechToTextV1
import json
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

The service endpoint is based on the location of the service instance, we store the information in the variable URL. To find out which URL to use, view the service credentials.

[ ]:
```
url_s2t = "https://stream.watsonplatform.net/speech-to-text/api"
```

You require an API key, and you can obtain the key on the Dashboard .

[ ]:
```
iam_apikey_s2t = ""
```

You create a Speech To Text Adapter object the parameters are the endpoint and API key.

[ ]:
```
authenticator = IAMAuthenticator(iam_apikey_s2t)
s2t = SpeechToTextV1(authenticator=authenticator)
s2t.set_service_url(url_s2t)
s2t
```

Lets download the audio file that we will use to convert into text.

[ ]:
```
!wget -O PolynomialRegressionandPipelines.mp3  https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/PY0101EN/labs/PolynomialRegressionandPipelines.mp3
```

We have the path of the wav file we would like to convert to text

[ ]:
```
filename='PolynomialRegressionandPipelines.mp3'
```

We create the file object `wav` with the wav file using `open` ; we set the `mode` to "rb" , this is similar to read mode, but it ensures the file is in binary mode.We use the method `recognize` to return the recognized text. The parameter audio is the file object `wav` , the parameter `content_type` is the format of the audio file.

[ ]:
```
with open(filename, mode="rb") as wav:
    response = s2t.recognize(audio=wav, content_type='audio/mp3')
```

The attribute result contains a dictionary that includes the translation:

[ ]:
```
response.result
```

[ ]:
```
from pandas.io.json import json_normalize

json_normalize(response.result['results'],"alternatives")
```

[ ]:
```
response
```

We can obtain the recognized text and assign it to the variable `recognized_text` :

[ ]:
```
recognized_text=response.result['results'][0]["alternatives"][0]["transcript"]
type(recognized_text)
```

Language Translator

First we import `LanguageTranslatorV3` from ibm_watson. For more information on the API click here

```
[ ]: from ibm_watson import LanguageTranslatorV3
```

The service endpoint is based on the location of the service instance, we store the information in the variable URL. To find out which URL to use, view the service credentials.

```
[ ]: url_lt='https://gateway.watsonplatform.net/language-translator/api'
```

You require an API key, and you can obtain the key on the Dashboard.

```
[ ]: apikey_lt=''
```

API requests require a version parameter that takes a date in the format version=YYYY-MM-DD. This lab describes the current version of Language Translator, 2018-05-01

```
[ ]: version_lt='2018-05-01'
```

we create a Language Translator object `language_translator` :

```
[ ]: authenticator = IAMAuthenticator(apikey_lt)
language_translator = LanguageTranslatorV3(version=version_lt,authenticator=authenticator)
language_translator.set_service_url(url_lt)
language_translator
```

We can get a Lists the languages that the service can identify. The method Returns the language code. For example English (en) to Spanis (es) and name of each language.

```
[ ]: from pandas.io.json import json_normalize

json_normalize(language_translator.list_identifiable_languages().get_result(), "languages")
```

We can use the method `translate` this will translate the text. The parameter text is the text. Model_id is the type of model we would like to use use we use list the language . In this case, we set it to 'en-es' or English to Spanish. We get a Detailed Response object translation_response

```
[ ]: translation_response = language_translator.translate(\
    text=recognized_text, model_id='en-es')
translation_response
```

The result is a dictionary.

```
[ ]: translation=translation_response.get_result()
translation
```

We can obtain the actual translation as a string as follows:

```
[ ]: spanish_translation =translation['translations'][0]['translation']
spanish_translation
```

We can translate back to English

```
[ ]: translation_new = language_translator.translate(text=spanish_translation ,model_id='es-en').get_result()
```

We can obtain the actual translation as a string as follows:

```
[ ]: translation_eng=translation_new['translations'][0]['translation']
translation_eng
```

We can convert it to French as well:

```
[ ]: French_translation=language_translator.translate(
    text=translation_eng , model_id='en-fr').get_result()
```

```
[ ]: French_translation['translations'][0]['translation']
```

Language Translator

References

https://cloud.ibm.com/apidocs/speech-to-text?code=python

https://cloud.ibm.com/apidocs/language-translator?code=python

---

Authors:

Joseph Santarcangelo

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other Contributor(s)

Fan Jiang

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab