WatsonSTT

January 30, 2023

0.1 Install and Import Dependencies

Instructions YT Tutorial https://www.youtube.com/watch?v=A9_0OgW1LZU

use pip install if you don't have watson libraries installed

[ ]:
```python
import os
#!pip install ibm_watson
from ibm_watson import SpeechToTextV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import matplotlib.pyplot as plt
import json
from ibm_watson import SpeechToTextV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibm_watson import SpeechToTextV1
from ibm_watson.speech_to_text_v1 import CustomWord
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibm_watson.speech_to_text_v1 import CustomWord
from ibm_watson import LanguageTranslatorV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

from dotenv import dotenv_values

#link to IBM STT documentation:
#https://cloud.ibm.com/apidocs/speech-to-text?code=python#createlanguagemodel
```

0.2 Setup STT Service

Will be using the python-dotenv library to store API keys so that they are not publicly available on GitHub. I have confiured a .env file as described on the following link: https://github.com/theskumar/python-dotenv

[ ]:
```python
config = dotenv_values(".env")

apikey = config["stt_apikey"]
url = config["stt_url"]
```

[ ]:
```python
# Setup Service
authenticator = IAMAuthenticator(apikey)
```

```
stt = SpeechToTextV1(authenticator=authenticator)
stt.set_service_url(url)
```

0.3 Open Audio Source and Convert

[ ]:

```
# Perform conversion

#Audio file taken from following link:
#https://www.youtube.com/watch?v=pOZkbANHlTI

model_name = 'en-US_NarrowbandModel'

audio_file_path = "/Users/nikhilgopal/IBM Trainings/Watson STT Training/
 ↪news_file.mp3"

with open(audio_file_path, 'rb') as f:
    res = stt.recognize(audio=f, content_type='audio/mp3', model=model_name).
 ↪get_result()
```

[ ]:

```
#res
```

[ ]:

```
text = res['results'][0]['alternatives'][0]['transcript']
text
```

[ ]:

```
"the world come doing a record number of viewers and try and for Argentina's Lin
L. Massey here's a look by the numbers fifteen point four million U. S.
households watched yesterday's World Cup final and all time U. S. audience
record and again certainly delivered on the drama for just the third time in
history the World Cup title was decided by a penalty shoot out thirty six years
old that's a long Argentina had waited for a World Cup win thirty five year old
Lino Massey had not even been born yet the last time is country once he scored
two goals and the first of five should all penalties to win his team along
company championship meeting crowns on a legendary career with seven hundred
ninety three eagles forty two club an international titles and a record seven
alone is doing %HESITATION mole ready to his name the World Cup win makes messy
the most decorated soccer player in history along with Brazilian player Danielle
she's behind only Cristiano Ronaldo in all time goals but he is three hundred
fifty assist to his name that's one hundred sixteen more than one although and
that's despite playing one hundred forty three fewer games "
```

[ ]:

```
confidence_levels = {}
confidence = res['results'][0]['alternatives'][0]['confidence']
confidence_levels[model_name] = confidence
```

0.4 Save Output as Text File

[ ]:

```
with open('output.txt', 'w') as out:
    out.writelines(text)
f.close()
```

0.5 Change Language Models

Link to model catalog:

https://cloud.ibm.com/apidocs/speech-to-text?code=python#listmodels https://ibmdocs-test.mybluemix.net/docs/en/watson-libraries?topic=home-models-catalog

[ ]:

```
# Perform conversion
model_name2 = 'en-US_Multimedia'

with open(audio_file_path, 'rb') as f:
    res2 = stt.recognize(audio=f, content_type='audio/mp3', model=model_name2).
  ↪get_result()
```

[ ]:

```
#res
text2 = res2['results'][0]['alternatives'][0]['transcript']
text2
```

[ ]:

```
"the world come during a record number of viewers and triumph for argentina's
leonel messy here's a look by the numbers fifteen point four million u s
households watch yesterday's world cup final and all time u s audience record
and the game certainly delivered on the drama for just the third time in history
the world cup title was decided by a penalty shoot out thirty six years folks
that's how long argentina had waited for a world cup when thirty five year old
leonel messy had not even been born yet the last time his country won he scored
two goals and the first of five shoot out penalties to win his team the long
covered championship a fitting crown on a legendary career with seven hundred
ninety three goals forty two club and international titles and a record seven
balance day auto already to his name the world cup win makes messy the most
decorated soccer player in history along with brazilian player danny alves he's
behind only christian renaulto in all time goals but he has three hundred fifty
assists to his name that's one hundred sixteen more than renault and that's
despite playing one hundred forty three fewer games seventy one goals in a
single season make messy a world record holder that includes thirteen in world
cup tournaments he's the only player in history to score in a world cup in his
teens as twenties and also is thirties and now the countdown begins to twenty
twenty six the united states mexico in canada will host the next world cup it
will be the first time the event will be held across three countries well g fans
robert roberts here thanks for checking out our youtube channel lots of great
stuff here so go on click the subscribe button right over right over here to get
more of awesome videos and contents from gma every day anytime we thank you for
watching and we'll see you in the morning on g m a "
```

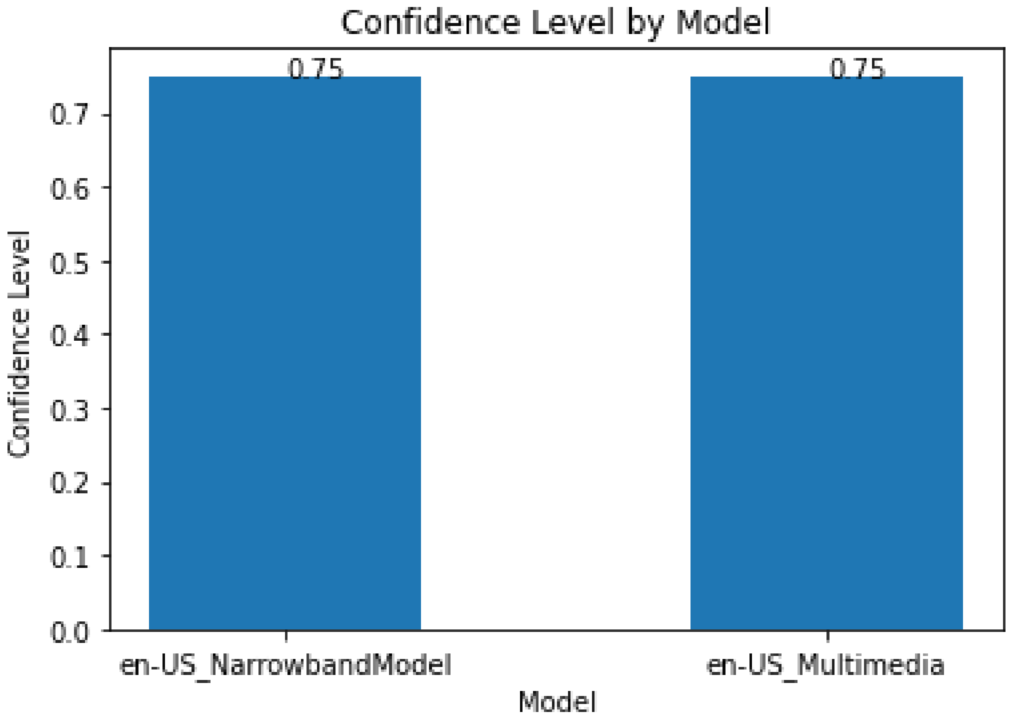
[ ]:

```
confidence2 = res['results'][0]['alternatives'][0]['confidence']
confidence2
confidence_levels[model_name2] = confidence2
```

It appears that the multimedia model has a higher confidence than the standard narrowband model:

[ ]:

```
# function to add value labels
def addlabels(x,y):
    for i in range(len(x)):
        plt.text(i,y[i],y[i])

if __name__ == '__main__':
    # creating data on which bar chart will be plot
    x = list(confidence_levels.keys())
    y = list(confidence_levels.values())

    # making the bar chart on the data
    plt.bar(x, y, width= 0.5, align="center")

    # calling the function to add value labels
    addlabels(x, y)

    # giving title to the plot
    plt.title("Confidence Level by Model")

    # giving X and Y labels
    plt.xlabel("Model")
    plt.ylabel("Confidence Level")

    # visualizing the plot
    plt.show()
```



0.6 Create A Custom Language Model

Import libraries:

[ ]:

```
authenticator = IAMAuthenticator(apikey)
speech_to_text = SpeechToTextV1(
    authenticator=authenticator
)

speech_to_text.set_service_url(url)
```

Create Model:

[ ]:

```
language_model = speech_to_text.create_language_model(
    'add_words',
    'en-US_BroadbandModel',
    description='Trying out adding words'
).get_result()

print(json.dumps(language_model, indent=2))
```

```
{
  "customization_id": "2b2fe6ab-b9c3-4c78-b41f-11b7e104ce1c"
}
```

Display all custom language models:

[ ]:
```
%%capture

#using capture to prevent display of output as list is too long, will paste the first two in markdown box below for demonstration purposes

language_models = speech_to_text.list_language_models().get_result()
print(json.dumps(language_models, indent=2))
```

{ “customizations”: [{ “owner”: “bfa0c14c-97ab-489d-93f6-4d45b9396076”, “base_model_name”: “en-US_BroadbandModel”, “customization_id”: “edf3b09e-3ccf-45ba-8459-f9d3654b1ffe”, “dialect”: “en-US”, “versions”: [“en-US_BroadbandModel.v2020-01-16”], “created”: “2023-01-10T02:49:20.563Z”, “name”: “add_words”, “description”: “Trying out adding words”, “progress”: 0, “language”: “en-US”, “updated”: “2023-01-10T02:49:20.563Z”, “status”: “pending” },

Get attributes of a specific model by custom ID:

[ ]:
```
model_id = "c72e276f-9262-4362-855a-89b8f21cda47"

language_model = speech_to_text.get_language_model(model_id).get_result()
print(json.dumps(language_model, indent=2))
```

```
{
  "owner": "bfa0c14c-97ab-489d-93f6-4d45b9396076",
  "base_model_name": "en-US_BroadbandModel",
  "customization_id": "c72e276f-9262-4362-855a-89b8f21cda47",
  "dialect": "en-US",
  "versions": [
    "en-US_BroadbandModel.v2020-01-16"
  ],
  "created": "2022-12-28T16:39:53.952Z",
  "name": "add_words",
  "description": "Trying out addinIntellijIdeaRulezzz ",
  "progress": 0,
  "language": "en-US",
  "updated": "2023-01-10T03:04:34.285Z",
  "status": "ready"
}
```

0.7 Train Language Model:

Add Custom Words:

Start with checking which custom words are already in the model, then add some more:

[ ]:
```
words = speech_to_text.list_words(model_id).get_result()
print(json.dumps(words, indent=2))
```

```python
authenticator = IAMAuthenticator(apikey)
speech_to_text = SpeechToTextV1(
    authenticator=authenticator
)

speech_to_text.set_service_url(url)

messi_word_obj = CustomWord(
    word='Messi',
    sounds_like=['messy', 'Messy', 'massey'],
    display_as='Messi'
)

lionel_word_obj = CustomWord(
    word='Lionel',
    sounds_like=['leonel']
)

trying_word_obj = CustomWord(
    word='trying',
    sounds_like=['tryin']
)

speech_to_text.add_words(
    model_id,
    [messi_word_obj, lionel_word_obj, trying_word_obj]
)
# Poll for language model status.
words = speech_to_text.list_words(model_id).get_result()
print(json.dumps(words, indent=2))
```

```
{
  "words": [
    {
      "display_as": "Lionel",
      "sounds_like": [
        "leonel"
      ],
      "count": 1,
      "source": [
        "user"
      ],
      "word": "Lionel"
    },
    {
      "display_as": "Messi",
      "sounds_like": [
        "Messy",
        "massey",
        "messy"
      ],
      "count": 1,
      "source": [
        "user"
      ],
      "word": "Messi"
    },
    {
      "display_as": "Montiel's",
      "sounds_like": [
        "Montiel's"
      ],
      "count": 1,
      "source": [
        "corpus1"
      ],
      "word": "Montiel's"
    },
    {
      "display_as": "handywoman",
      "sounds_like": [
        "handywoman"
      ],
      "count": 1,
      "source": [
        "corpus1"
      ],
      "word": "handywoman"
    },
    {
      "display_as": "trying",
      "sounds_like": [
        "tryin"
      ],
      "count": 1,
      "source": [
        "user"
      ],
      "word": "trying"
    }
  ]
}
{
  "words": [
    {
      "sounds_like": [
        "leonel"
      ],
      "count": 1,
      "source": [
        "user"
      ],
      "word": "Lionel"
    },
    {
      "display_as": "Messi",
      "sounds_like": [
        "Messy",
        "massey",
        "messy"
      ],
      "count": 1,
      "source": [
        "user"
      ],
      "word": "Messi"
    },
    {
      "display_as": "Montiel's",
      "sounds_like": [
        "Montiel's"
      ],
      "count": 1,
      "source": [
        "corpus1"
      ],
      "word": "Montiel's"
    },
    {
      "display_as": "handywoman",
      "sounds_like": [
        "handywoman"
      ],
      "count": 1,
      "source": [
        "corpus1"
      ],
      "word": "handywoman"
    },
    {
      "sounds_like": [
        "tryin"
      ],
        "count": 1,
        "source": [
          "user"
        ],
        "word": "trying"
      }
    ]
  }
```

Run training:

[ ]:

```
speech_to_text.train_language_model(model_id, word_type_to_add="all",␣
 ↪strict=False)
```

0.8 Run and Test Trained Model

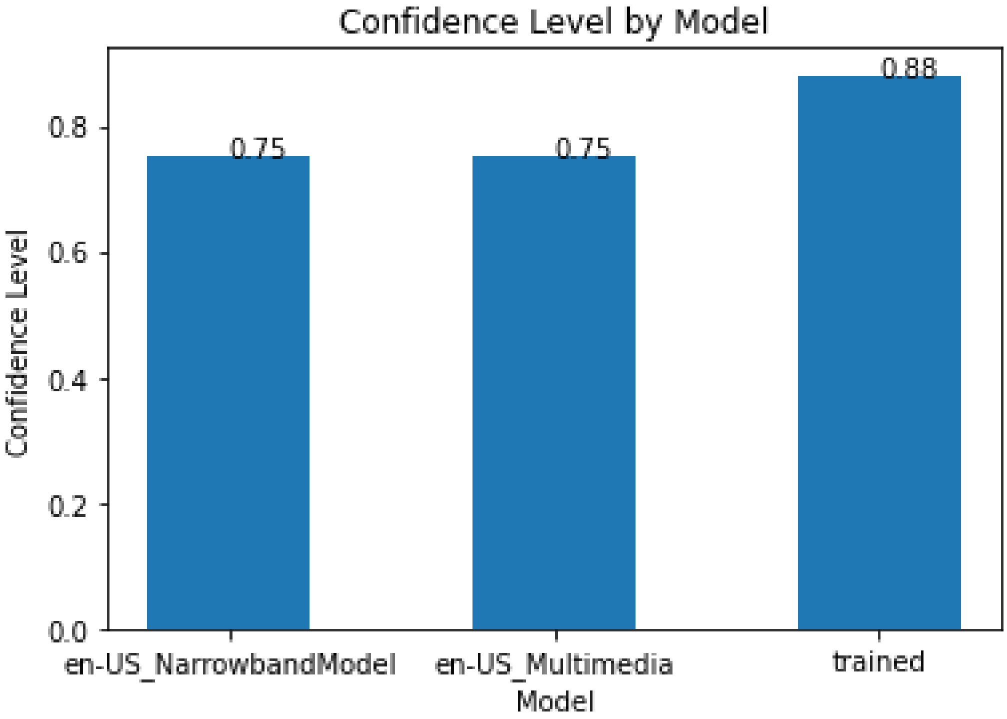
[ ]:

```
with open(audio_file_path, 'rb') as f:
    res_word_added = stt.recognize(audio=f, content_type='audio/mp3',␣
 ↪language_customization_id=model_id).get_result()
```

Save output of new trained model and compare confidence levels with old models:

[ ]:

```
text_trained = res_word_added['results'][0]['alternatives'][0]['transcript']

with open('output2.txt', 'w') as out:
    out.writelines(text_trained)
f.close()
out.close()

confidence_trained =␣
 ↪res_word_added['results'][0]['alternatives'][0]['confidence']

confidence_levels["trained"] = confidence_trained

if __name__ == '__main__':
    # creating data on which bar chart will be plot
    x = list(confidence_levels.keys())
    y = list(confidence_levels.values())

    # making the bar chart on the data
    plt.bar(x, y, width= 0.5, align="center")

    # calling the function to add value labels
    addlabels(x, y)

    # giving title to the plot
    plt.title("Confidence Level by Model")
    # giving X and Y labels
    plt.xlabel("Model")
    plt.ylabel("Confidence Level")

    # visualizing the plot
    plt.show()
```



Test effects of Custom Words

```
[ ]: new_file_path = "/Users/nikhilgopal/IBM Trainings/Watson STT Training/
      ↪messy-messi.mp3"

     with open(new_file_path, 'rb') as f:
         res_word_added = stt.recognize(audio=f, content_type='audio/mp3',␣
      ↪language_customization_id=model_id).get_result()
```

```
[ ]: text_test = res_word_added['results'][0]['alternatives'][0]['transcript']

     print(text_test)
```

```
the room is very Messi the room is extremely Messi Lionel Messi is the best
football player in the world
```

The model transcribes every word as Messi. Lets see if adding a Corpus can help the model distinguish between sports related and non-sports related and non sports related situations.

0.9 Add Corpus

Check if there are any corpora already:

```
[ ]: corpora = speech_to_text.list_corpora(model_id).get_result()
     print(json.dumps(corpora, indent=2))
```

```
{
  "corpora": [
    {
      "out_of_vocabulary_words": 2,
      "total_words": 528,
      "name": "corpus1",
      "status": "analyzed"
    }
  ]
}
```

```
[ ]: #pass path to corpus file

     with open("/Users/nikhilgopal/IBM Trainings/Watson STT Training/samplecorpus.
      ↪txt",
               'rb') as corpus_file:
         speech_to_text.add_corpus(
             model_id,
             'corpus1',
             corpus_file
         )

     # Poll for corpus status.
```

```
[ ]: #check if corpus added successfully

     corpora = speech_to_text.list_corpora(model_id).get_result()
     print(json.dumps(corpora, indent=2))
```

```
{
  "corpora": [
    {
      "out_of_vocabulary_words": 2,
      "total_words": 528,
      "name": "corpus1",
      "status": "analyzed"
    }
  ]
}
```

Check performance of model after adding Corpus:

[ ]:

```
with open(new_file_path, 'rb') as f:
    res_word_added = stt.recognize(audio=f, content_type='audio/mp3',␣
 ↪language_customization_id=model_id).get_result()

text_test2 = res_word_added['results'][0]['alternatives'][0]['transcript']

print(text_test2)
```

```
the room is very Messi the room is extremely Messi Lionel Messi is the best
football player in the world
```

0.10 Watson Translation

List supported languages

[ ]:

```
translate_api_key = config["translate_api_key"]
translate_url = config["translate_url"]

authenticator = IAMAuthenticator(translate_api_key)
language_translator = LanguageTranslatorV3(
    version='2018-05-01',
    authenticator=authenticator
)

language_translator.set_service_url(translate_url)

languages = language_translator.list_languages().get_result()
#print(json.dumps(languages, indent=2))
```

0.11 Identify Text Language

[ ]:

```
language = language_translator.identify(
    text_trained).get_result()
print(json.dumps(language, indent=2))
```

```
{
  "languages": [
    {
      "language": "en",
      "confidence": 0.999999999983779
    },
    {
      "language": "nn",
      "confidence": 5.205084059226688e-12
    },
    {
      "language": "tr",
      "confidence": 2.0377123947884035e-12
```

```
},
{
  "language": "tl",
  "confidence": 1.995105145518952e-12
},
{
  "language": "nl",
  "confidence": 1.4529303286820957e-12
},
{
  "language": "vi",
  "confidence": 1.4519940530481691e-12
},
{
  "language": "de",
  "confidence": 1.2823257547489229e-12
},
{
  "language": "da",
  "confidence": 8.721800702126013e-13
},
{
  "language": "af",
  "confidence": 7.587439877135221e-13
},
{
  "language": "ku",
  "confidence": 2.2913998442867956e-13
},
{
  "language": "nb",
  "confidence": 2.0782208303034792e-13
},
{
  "language": "es",
  "confidence": 1.821021283721029e-13
},
{
  "language": "ms",
  "confidence": 1.3013896502491997e-13
},
{
  "language": "fr",
  "confidence": 1.0670265199566752e-13
},
{
  "language": "sv",
  "confidence": 7.16234490568824e-14
```

```
},
{
  "language": "ht",
  "confidence": 6.20015820314456e-14
},
{
  "language": "mt",
  "confidence": 5.992351213106732e-14
},
{
  "language": "cs",
  "confidence": 2.7243560180003025e-14
},
{
  "language": "it",
  "confidence": 2.3478306758178337e-14
},
{
  "language": "et",
  "confidence": 1.3718248828564024e-14
},
{
  "language": "sq",
  "confidence": 1.2402888800313151e-14
},
{
  "language": "ro",
  "confidence": 1.1261935736672415e-14
},
{
  "language": "hu",
  "confidence": 4.88116665454589e-15
},
{
  "language": "ca",
  "confidence": 4.494262228979742e-15
},
{
  "language": "pt",
  "confidence": 2.9822016905045175e-15
},
{
  "language": "sk",
  "confidence": 2.9797746516755215e-15
},
{
  "language": "fi",
  "confidence": 2.9057197915298757e-15
```

```
},
{
  "language": "so",
  "confidence": 2.0520670612217515e-15
},
{
  "language": "ja",
  "confidence": 8.970665982814521e-16
},
{
  "language": "az",
  "confidence": 8.966687143040534e-16
},
{
  "language": "hr",
  "confidence": 8.581262966557105e-16
},
{
  "language": "is",
  "confidence": 6.31924634060837e-16
},
{
  "language": "lt",
  "confidence": 6.198650395101557e-16
},
{
  "language": "lv",
  "confidence": 5.410543463883793e-16
},
{
  "language": "cy",
  "confidence": 5.248654983183351e-16
},
{
  "language": "ko",
  "confidence": 5.085963701117192e-16
},
{
  "language": "sl",
  "confidence": 4.1137574945464203e-16
},
{
  "language": "pl",
  "confidence": 4.050396038025295e-16
},
{
  "language": "eu",
  "confidence": 2.777259012399554e-16
```

```
    },
    {
      "language": "ga",
      "confidence": 2.1596756816470315e-16
    },
    {
      "language": "eo",
      "confidence": 2.0096580560045378e-16
    },
    {
      "language": "el",
      "confidence": 7.797080707008497e-17
    },
    {
      "language": "th",
      "confidence": 3.325022046295735e-17
    },
    {
      "language": "zh",
      "confidence": 3.306948477481636e-17
    },
    {
      "language": "zh-TW",
      "confidence": 3.115600910829709e-17
    },
    {
      "language": "hi",
      "confidence": 8.931800897980414e-18
    },
    {
      "language": "ar",
      "confidence": 1.7780690344255185e-18
    },
    {
      "language": "ru",
      "confidence": 1.676687061614231e-18
    },
    {
      "language": "mn",
      "confidence": 1.073830111661764e-18
    },
    {
      "language": "he",
      "confidence": 9.470203910211584e-19
    },
    {
      "language": "ur",
      "confidence": 6.095914211309237e-19
```

```
},
{
  "language": "my",
  "confidence": 5.525344896600658e-19
},
{
  "language": "mr",
  "confidence": 3.048639350570273e-19
},
{
  "language": "uk",
  "confidence": 2.529210354455543e-19
},
{
  "language": "lo",
  "confidence": 2.459216617465464e-19
},
{
  "language": "ka",
  "confidence": 1.909009509685216e-19
},
{
  "language": "sr",
  "confidence": 1.8520649439821095e-19
},
{
  "language": "km",
  "confidence": 1.2896208402159602e-19
},
{
  "language": "ne",
  "confidence": 9.211279433790811e-20
},
{
  "language": "pa",
  "confidence": 8.448563775453104e-20
},
{
  "language": "bn",
  "confidence": 7.570302085992943e-20
},
{
  "language": "ky",
  "confidence": 6.868065732356594e-20
},
{
  "language": "be",
  "confidence": 6.793994937676317e-20
```

```
},
{
  "language": "bg",
  "confidence": 6.109030892669676e-20
},
{
  "language": "ta",
  "confidence": 6.064900593247758e-20
},
{
  "language": "kk",
  "confidence": 6.055944439737917e-20
},
{
  "language": "ml",
  "confidence": 5.887847920927901e-20
},
{
  "language": "ps",
  "confidence": 4.9559388935298433e-20
},
{
  "language": "te",
  "confidence": 3.909932665375015e-20
},
{
  "language": "hy",
  "confidence": 3.562987128731175e-20
},
{
  "language": "ba",
  "confidence": 2.941990123077224e-20
},
{
  "language": "fa",
  "confidence": 2.852190727456381e-20
},
{
  "language": "pa-PK",
  "confidence": 2.1707259394152715e-20
},
{
  "language": "cv",
  "confidence": 1.7933799913344533e-20
},
{
  "language": "gu",
  "confidence": 1.6225920100138852e-20
```

```
    },
    {
      "language": "si",
      "confidence": 1.533190439927532e-20
    }
  ]
}
```

0.12 Translate the text

[ ]:

```
translation = language_translator.translate(
    text=text_trained,
    model_id='en-es').get_result()

print(json.dumps(translation, indent=2, ensure_ascii=False))
```

```
{
  "translations": [
    {
      "translation": "el mundo vienen a unirse a un número récord de
espectadores y tratando de que el argentino Lionel Messi aquí es una mirada por
los números quince puntos cuatro millones de U. S. hogares ver la final de la
Copa del Mundo de ayer y todo el tiempo U. S. récord de audiencia y juego sin
duda entregado en el drama por sólo la tercera vez en la historia el título de
la Copa del Mundo fue decidido por un tiro penal de treinta y seis años amigos
que 'es cuánto tiempo Argentina había esperado a una Copa del Mundo ganar
treinta y cinco años Lionel Messi ni siquiera había nacido todavía la última vez
que su país marcó dos goles y el primero de cinco tiros fuera de penales para
ganar a su equipo el largo ansiado campeonato de tiro corona en una carrera
legendaria con setecientos noventa y tres goles cuarenta dos clubes un
internacional y un récord de siete Malone ya está ahí a su nombre la Copa del
Mundo gana hace a Messi el jugador de fútbol más condecorado en la historia
junto con El jugador brasileño Dani Alves sólo está detrás de Cristiano Ronaldo
en todos los goles de tiempo, pero él es trescientos cincuenta asistencias a su
nombre que es cien dieciséis más que Ronaldo y eso es a pesar de jugar cien
cuarenta y tres juegos menos "
    }
  ],
  "word_count": 212,
  "character_count": 1133
}
```

Save a text file of the translation:

[ ]:

```
text_translated = translation['translations'][0]['translation']

with open("output_translated.txt", 'w') as f:
    f.writelines(text_translated)
```

```
f.close()
```