

باسمه تعالی

ابتدا کتابخانه های مورد نیاز را import می کنیم

```
# requirement packages
!pip install git+https://github.com/huggingface/datasets.git
!pip install git+https://github.com/huggingface/transformers.git
!pip install torchaudio
!pip install librosa
!pip install jiwer
!pip install hazm
import librosa
import torch
import torchaudio
import soundfile as sf
from transformers import Wav2Vec2ForCTC, Wav2Vec2Processor
from datasets import load_dataset

from IPython.display import Audio
from scipy.io import wavfile
import numpy as np

import IPython.display as ipd
```

صوت موردنظر را می خوانیم

```
file_name = 'untitled7.wav'

Audio(file_name)

data = wavfile.read(file_name)

print('Sample rate:',data[0],'Hz')
print('Total time:',len(data[1])/data[0],'s')

input_audio, _ = librosa.load(file_name, sr=16000)
input_audio.dtype
```

ماژول اول – بازشناسی گفتار:

صوت خوانده شده را به طور کامل به مدل از پیش آموزش داده شده فارسی wav2vec 2.0 می دهیم و متن متناظر با آن را دریافت می کنیم

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
processor = Wav2Vec2Processor.from_pretrained("m3hrdadfi/wav2vec2-large-
xlsr-persian")
```

```
model = Wav2Vec2ForCTC.from_pretrained("m3hrdadfi/wav2vec2-large-xlsr-persian").to(device)
```

```
input_values = processor(input_audio, return_tensors="pt").input_values
logits = model(input_values).logits
predicted_ids = torch.argmax(logits, dim=-1)
transcription = processor.batch_decode(predicted_ids)[0]
print(transcription)
```

مشکلات متن تولید شده را با استفاده از کدهای زیر برطرف می‌کنیم

```
import sys, os, difflib, argparse
from datetime import datetime, timezone
import keyword
#difflib.get_close_matches('appel', ['ape', 'apple', 'peach', 'puppy'])
#get_close_matches('wheel', keyword.kwlist)
sentence1 = difflib.get_close_matches('سپر', ['صفر', 'یک', 'دو', 'سه', 'چه', 'ار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=4)
sentence2 = difflib.get_close_matches('دعلاوه', ['صفر', 'یک', 'دو', 'سه', 'چهار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=1)
sentence3 = difflib.get_close_matches('س', ['صفر', 'یک', 'دو', 'سه', 'چهار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=1)
sentence4 = difflib.get_close_matches('بهعلاوه', ['صفر', 'یک', 'دو', 'سه', 'چهار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=1)
sentence5 = difflib.get_close_matches('من ها', ['صفر', 'یک', 'دو', 'سه', 'چهار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=1)
sentence6 = difflib.get_close_matches('بهعلاوه', ['صفر', 'یک', 'دو', 'سه', 'چهار', 'پنج', 'شش', 'هفت', 'هشت', 'نه', 'بعلاوه', 'منها', 'ضرب', 'تقسیم'], cutoff=0.3, n=1)
#print(sentence1)
a=str(sentence1[3])
b=str(sentence2[0])
c=str(sentence3[0])
d=str(sentence4[0])
e=str(sentence5[0])
f=str(sentence6[0])
print(c)
if 'سپر' in transcription:
    transcription = transcription.replace('سپر', a, 4)
if 'دعلاوه' in transcription:
```

```

transcription = transcription.replace('دعلاوه',b,2)
if 'س' in transcription and 'سه' not in transcription:
    transcription = transcription.replace('س',c,2)
if 'بهعلايه' in transcription:
    transcription = transcription.replace('بهعلايه',d,2)
if 'من ها' in transcription:
    transcription = transcription.replace('من ها',e,2)
if 'بهعلاوه' in transcription:
    transcription = transcription.replace('بهعلاوه',f,2)

print(transcription)

```

متن تولید شده را به سه بخش تبدیل می کنیم. بخش اول برابر با کلمات مربوط به عدد اول، بخش دوم مربوط به عملگر و بخش سوم مربوط به کلمات عدد دوم است و کلمات مربوط به عدد را با استفاده از دیکشنری به رقم تبدیل می کنیم و ارقام تولید شده را کنار هم قرار می دهیم و یک عدد n رقمی تولید می کنیم.

```

split4= transcription.split(' ')
print(split4)
mydict = {'صفر':0,'یک':1,'دو':2,'سه':3,'چهار':4,'پنج':5,'شش':6,'هفت':7,'هش':8,'نه':9,'بعلاوه':+',','منها':'-','ضرب':'*','تقسیم': '/'}
print(mydict['تقسیم'])
amalgar = ['بعلاوه','منها','ضرب','تقسیم']
for char in amalgar:
    if char in split4:
        x = split4.index(char)
        print(x)
adadaval=split4[0:x]
print(adadaval)
amalgar=mydict[split4[x]]
print(amalgar)
adaddovom=split4[x+1:len(split4)]
print(adaddovom)

```

```

str1=''
i = range(len(adadaval))
for char in i:
    a = mydict[adadaval[char]]
    str1 = str1 + str(a)
str1 = int(str1)
print(str1)

```

```

str2=''
i = range(len(adaddovom))
for char in i:

```

```

b = mydict[adaddovom[char]]
str2 = str2 + str(b)
str2 = int(str2)
print(str2)

```

ارقام تولیدی و عملگر به دست آمده را به عنوان خروجی برمی گردانیم.

```

print(str1)
print(amalgar)
print(str2)

```

ماژول دوم – ماشین حساب:

این ماژول خروجی ماژول اول را دریافت کرده، عملیات ریاضی مورد نظر را انجام می دهد و نتیجه را به عنوان خروجی بر می گرداند.

```

num1 = int(str1)
num2 = int(str2)
if amalgar== '+':
    natije = num1+num2
elif amalgar== '-':
    natije = num1-num2
elif amalgar== '*':
    natije = num1*num2
elif amalgar== '/':
    natije = num1/num2
else:
    print("error")
print(natije)

```

ماژول سوم – سنتز گفتار:

ابتدا اعداد اول و دوم را از حالت رقمی به حالت نوشتاری تبدیل می کنیم و عملگر را هم به متن تبدیل کنیم.

```

mydictamalgar = {'+': 'به علاوه', '-': 'منها', '*': 'ضرب', '/': 'تقسیم'}

```

```

!pip install 'setuptools>=36.2.1'
!pip install num2fawords
from num2fawords import words, ordinal_words
num1 = words(str1)
num2 = words(str2)
natije = words(natije)
print(num1)
print(num2)
print(natije)
mydictamalgar[amalgar]
print(type(num1))

```

```

print(type(mydictamalgar[amalgar]))
matn = num1 + ' ' + mydictamalgar[amalgar] + ' ' + num2 + ' ' + 'برابر است ب'
      |' + ' ' + natije
print(matn)

```

فونم ها را با استفاده از کد زیر می خوانیم:

```

phone_dir1 = './phonems'
phone_dir2 = './phonems2'

import numpy as np
import os
osjoin = os.path.join
phones = {}

for ph in os.listdir(phone_dir1):
    if not os.path.isdir(ph):
        print(ph)
        y, sr = librosa.load(osjoin(phone_dir1, ph), sr=16000, mono=True)
        phones[os.path.basename(ph)[-4]] = y

for ph in os.listdir(phone_dir2):
    if not os.path.isdir(ph):
        print(ph)
        y, sr = librosa.load(osjoin(phone_dir2, ph), sr=16000, mono=True)
        phones[os.path.basename(ph)[-4]] = y

phones[' ' ] = np.zeros(2000)

```

حالا می خواهیم متن موردنظر را نرمالایز کنیم البته یک اروری که این جا دریافت کردم و برطرف کردم را لازم می دانم بنویسم
 شاید روزی به کار آمد ابتدا که متن را می خواندم ارور زیر را می داد که می گفت کاراکتر \u در متن هست

```
[28] !pip install hazm
!pip install https://github.com/sobhe/hazm/archive/master.zip --upgrade
from __future__ import unicode_literals
from hazm import *
print(matn)
normalizer = Normalizer()
normall = normalizer.normalize(matn)
print(normall)

Requirement already satisfied: hazm in /usr/local/lib/python3.7/dist-packages (0.7.1)
Requirement already satisfied: libwapi==0.2.1 in /usr/local/lib/python3.7/dist-packages (from hazm) (0.2.1)
Requirement already satisfied: nltk==3.4 in /usr/local/lib/python3.7/dist-packages (from hazm) (3.4)
Requirement already satisfied: singledispatch in /usr/local/lib/python3.7/dist-packages (from nltk==3.4->hazm) (3.7.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk==3.4->hazm) (1.15.0)
Collecting https://github.com/sobhe/hazm/archive/master.zip
  Using cached https://github.com/sobhe/hazm/archive/master.zip
Requirement already satisfied: nltk==3.4 in /usr/local/lib/python3.7/dist-packages (from hazm==0.7.1) (3.4)
Requirement already satisfied: libwapi==0.2.1 in /usr/local/lib/python3.7/dist-packages (from hazm==0.7.1) (0.2.1)
Requirement already satisfied: singledispatch in /usr/local/lib/python3.7/dist-packages (from nltk==3.4->hazm==0.7.1) (3.7.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk==3.4->hazm==0.7.1) (1.15.0)
چهارصد و هشتاد و شش بعلاوه دویست و ده برابر است با شصت و هشتاد و شش
-----
KeyError                                Traceback (most recent call last)
/usr/lib/python3.7/sre_parse.py in parse_template(source, pattern)
    1014         try:
-> 1015             this = chr(ESCAPES[this][1])
    1016         except KeyError:

KeyError: '\\u'

During handling of the above exception, another exception occurred:

error                                Traceback (most recent call last)
5 frames
/usr/lib/python3.7/sre_parse.py in parse_template(source, pattern)
    1016         except KeyError:
    1017             if c in ASCII_LETTERS:
-> 1018                 raise s.error('bad escape %s' % this, len(this))
    1019             lappend(this)
    1020     else:

error: bad escape \u at position 0

SEARCH STACK OVERFLOW

!pip install PersianG2p
!pip install librosa
!pip install soundfile
import matplotlib.pyplot as plt
import numpy as np
import soundfile as sf
from PersianG2p import Persian_g2p_converter

PersianG2pConverter = Persian_g2p_converter(use_large = True)
input_text = PersianG2pConverter.transliterate(matn, tidy = False)
print(input_text)
input_text = str(input_text)
print(type(input_text))

Installing collected packages: PersianG2p
Successfully installed PersianG2p-0.3.0
Requirement already satisfied: librosa in /usr/local/lib/python3.7/dist-packages (0.8.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from librosa) (1.19.5)
7s  completed at 3:44 PM
```

حالا برای این که این مشکل را برطرف کنیم کد را به صورت زیر تغییر دادم و لازم بود کتابخانه regex را import کنیم چون در خود متن به صورت خودکار کاراکتر اضافه \u200c اضافه می شد این کاراکتر را با فاصله جایگزین کردم و کد به صورت زیر شد و درست کار کرد

```
import re
!pip install hazm
from __future__ import unicode_literals
from hazm import *
normalizer = Normalizer()
normall = normalizer.normalize(re.sub(r'\u200c', ' ', matn))
print(normall)
```

با استفاده از G2P، متن را به فونم های موردنظر تبدیل می کنیم و فونم ؟ را به _ تبدیل می کنیم.

```
!pip install PersianG2p
!pip install librosa
!pip install soundfile
import matplotlib.pyplot as plt
import numpy as np
import soundfile as sf
from PersianG2p import Persian_g2p_converter

PersianG2Pconverter = Persian_g2p_converter(use_large = True)
input_text = PersianG2Pconverter.transliterate(normal1, tidy = False)
print(input_text)
input_text = str(input_text)
print(type(input_text))

input_text1 = input_text.replace("?", "_")
print(input_text1)
```

صوت موردنظر را به دست می آوریم و ذخیره می کنیم

```
audio = np.concatenate([phones[i] for i in input_text1])
# audio = librosa.effects.time_stretch(audio, 1.5)
sf.write('output.wav', audio, 16000)
```

```
Audio('output.wav')
```

خروجی ها

خروجی مربوط به صوت untitled7.wav که توسط خودم ضبط شده است

ماژول اول:

چهار هفت شش د علاوه دو یک سپر
اصلاح شده

چهار هفت شش بعلاوه دو یک صفر

```
['چهار', 'هفت', 'شش']
+
['دو', 'یک', 'صفر']
476
+
210
686
```

ماژول دوم:

ماژول سوم:

چهارصد و هفتاد و شش بعلاوه دویست و ده برابر است با ششصد و هشتاد و شش

```
CAhArsad v a haftAd v a S e S be?alA?u dovist v a d a h b a r A b
a r a s t b A SeSsad v a haStAd v a S e S
```

```
CAhArsad v a haftAd v a S e S be_alA_u dovist v a d a h b a r A b
a r a s t b A SeSsad v a haStAd v a S e S
```

خروجی untitled7output.wav به دست آمد.

خروجی مربوط به صوت test0.wav

ماژول اول:

یک دو چهار بهعلاوه دو سه پنج
اصلاح شده

یک دو چهار بعلاوه دو سه پنج

```
['یک', 'دو', 'چهار']
+
['دو', 'سه', 'پنج']
124
+
235
```

ماژول دوم:

359

ماژول سوم:

یکصد و بیست و چهار بعلاوه دویست و سی و پنج برابر است با سیصد و پنجاه و نه

```
yeksad v a b i s t v a C A h A r be?alA?u dovist v a s i v a p
a n j b a r A b a r a s t b A sisad v a panjAh v a n o h
```

```
yeksad v a b i s t v a C A h A r be_alA_u dovist v a s i v a p
a n j b a r A b a r a s t b A sisad v a panjAh v a n o h
```

خروجی test0output.wav به دست آمد.

خروجی مربوط به صوت test1.wav

ماژول اول:

دو صفر س منها یک نه دو
اصلاح شده

دو صفر سه منها یک نه دو

['دو', 'صفر', 'سه']

-

['یک', 'نه', 'دو']

203

-

192

ماژول دوم:

11

ماژول سوم:

دویست و سه منها یکصد و نود و دو برابر است با یازده

dovist v a s e menhA yeksad v a navad v a d o b a r A b a r a s
t b A yAzdah

خروجی test1output.wav به دست آمد.

خروجی مربوط به صوت test2.wav

ماژول اول:

چهار هفت شش بهعلاوه دو یک صفر
اصلاح شده

چهار هفت شش بعلاوه دو یک صفر

['چهار', 'هفت', 'شش']

+

['دو', 'یک', 'صفر']

476

+

210

ماژول دوم:

686

ماژول سوم:

چهارصد و هفتاد و شش بعلاوه دویست و ده برابر است با ششصد و هشتاد و شش

CAhArsad v a haftAd v a S e S be?alA?u dovist v a d a h b a r A b
a r a s t b A SeSsad v a haStAd v a S e S

CAhArsad v a haftAd v a S e S be_alA_u dovist v a d a h b a r A b
a r a s t b A SeSsad v a haStAd v a S e S

خروجی test2output.wav به دست آمد.

خروجی مربوط به صوت test3.wav

ماژول اول:

سه یک پنج من ها سه یک پنج
اصلاح شده

سه یک پنج منها سه یک پنج

['سه', 'یک', 'پنج']

-

['سه', 'یک', 'پنج']

315

-

315

ماژول دوم:

0

ماژول سوم:

سیصد و پانزده منها سیصد و پانزده برابر است با صفر

sisad v a pAnzdah menhA sisad v a pAnzdah b a r A b a r a s t b A
s e f r

خروجی test3output.wav به دست آمد.

کد روی cpu در گوگل کولب اجرا شده است.