12_Text_To_Speech

April 23, 2022

Converting text to speech is another useful NLP technique. The simplest way is to use the gTTs library.

[ ]:
```
#!pip install gTTS
```

[ ]:
```
from gtts import gTTS
```

Now after you run the following code snippet, whatever you input in the text parameter gets converted into audio.

[ ]:
```
#chooses the language, English('en')
convert = gTTS(text='I like this NLP. How about dude!', lang="en", slow=False)
# Saving the converted audio in a mp3 file named
convert.save("audio.mp3")
```

[ ]:
```
!pip3 install pyttsx3
```

[ ]:
```
#pip3 install pyttsx3
#apt-get install alsa-utils
import pyttsx3, time
engine = pyttsx3.init()
engine.say("Hi, I am text to speach")
engine.runAndWait()
```

[ ]:
```
import pyttsx3
engine = pyttsx3.init() # object creation

""" RATE"""
rate = engine.getProperty('rate')   # getting details of current speaking rate
print (rate)                        #printing current voice rate
engine.setProperty('rate', 125)     # setting up new voice rate


"""VOLUME"""
volume = engine.getProperty('volume')   #getting to know current volume level␣
↪(min=0 and max=1)
print (volume)                          #printing current volume level
engine.setProperty('volume',1.0)    # setting up volume level  between 0 and 1
```

```
"""VOICE"""
voices = engine.getProperty('voices')       #getting details of current voice
engine.setProperty('voice', voices[0].id)  #changing index, changes voices. o␣
 ↪for male
#engine.setProperty('voice', voices[1].id)   #changing index, changes voices. 1␣
 ↪for female

engine.say("Hello World!")
engine.say('My current speaking rate is ' + str(rate))
engine.runAndWait()
engine.stop()

"""Saving Voice to a file"""
# On linux make sure that 'espeak' and 'ffmpeg' are installed
engine.save_to_file('Hello World', 'test.mp3')
engine.runAndWait()
```

1 Translating Speech

Whenever you try to analyze data from blogs hosted across the globe, especially websites from countries like China, where the Chinese language is predominant, analyzing such data or performing NLP tasks on such data would be difficult. That's where language translation comes to the rescue. You want to translate one language to another.

The easiest way to is to use the goslate library.

[ ]:
```
!pip install goslate
```

[ ]:
```
import goslate
```

[ ]:
```
text = "Bonjour le monde"
```

[ ]:
```
gs = goslate.Goslate()
translatedText = gs.translate(text,'en')
print(translatedText)
```

You can also use the polyglot library. It has various multilingual applications and supports more than 100 languages in NLP tasks, such as language detection, tokenization, NER, POS tagging, and sentiment analysis

[ ]:
```
import goslate
gs = goslate.Goslate()
language_id = gs.detect('hallo welt')
language_id
gs.get_languages()[language_id]
```

[ ]:
```
gs = goslate.Goslate(service_urls=['http://translate.google.de'])
language_id = gs.detect('hallo welt')
language_id
gs.get_languages()[language_id]
```

[ ]:
```
!pip install translate
```

[11]:
```
from translate import Translator
translator= Translator(to_lang="zh")
translation = translator.translate("This is a pen.")
```

[12]:
```
translation
```

[12]:
```
'   '
```

[13]:
```
from translate import Translator
translator= Translator(to_lang="hi")
translation = translator.translate("hello world")
translation
```

[13]:
```
'  -       '
```

[ ]:
```

```