

20 | TTS与语音合成：让你的机器人拥有声音

2023-04-21 徐文浩 来自北京

《AI大模型之美》



你好，我是徐文浩。

上一讲里，我们通过 Whisper 模型，让 AI “听懂”了我们在说什么。我们可以利用这个能力，让 AI 替我们听播客、做小结。不过，这只是我们和 AI 的单向沟通。那我们能不能更进一步，让 AI 不仅能“听懂”我们说的话，通过 ChatGPT 去回答我们问的问题，最后还能让 AI 把这些内容合成为语音，“说”给我们听呢？

当然可以，这也是我们这一讲的主题，我会带你一起来让 AI 说话。和上一讲一样，我不仅会教你如何使用云端 API 来做语音合成（Text-To-Speech），也会教你使用开源模型，给你一个用本地 CPU 就能实现的解决方案。这样，你也就不用担心数据安全的问题了。

使用 Azure 云进行语音合成

语音合成其实已经是一个非常成熟的技术了，现在在很多短视频平台里，你听到的很多配音其实都是通过语音合成技术完成的。国内外的各大公司都有类似的云服务，比如 [科大讯飞](#)、[阿里云](#)、[百度](#)、[AWS Polly](#)、[Google Cloud](#) 等等。不过，今天我们先来体验一下微软 Azure 云的语音合成 API。选用 Azure，主要有两个原因。

1. 因为微软和 OpenAI 有合作，Azure 还提供了 OpenAI 相关模型的托管。这样，我们在实际的生产环境使用的时候，只需要和一个云打交道就好了。
2. 价格比较便宜，并且提供了免费的额度。如果你每个月的用量在 50 万个字符以内，那么就不用花钱。

在运行代码之前，你需要先去注册一个 Azure 云的账号，并且开通 [微软认知服务](#)，然后开启对应的认知服务资源，获得自己的 API Key。我在这里放了对应文档的 [链接](#)，你照着文档一步步操作，就能完成。我在下面也放上了关键步骤的截图，具体注册过程，我就不一一介绍了。

[Home](#) >

Create Cognitive Services ...

[Basics](#) [Network](#) [Identity](#) [Tags](#) [Review + create](#)

Get access to Vision, Language, Search, and Speech Cognitive Services with a single API key. Quickly connect services together to achieve more insights into your content and easily integrate with other services like Azure Search.

[Learn more](#)

Project Details

Subscription * ⓘ

OPENAI-UAT



Resource group * ⓘ

[Create new](#)

Instance Details

Region ⓘ

East US



Name * ⓘ



Location specifies the region only for included regional services. This does not specify a region for included non-regional services. [Click here for more details.](#)



Pricing tier * ⓘ

[View full pricing details](#)

By checking this box I acknowledge that I
have read and understood all the terms
below *

☐

点击创建认知服务的链接，在自己的 Azure 云账号下，创建一个对应的认知服务

注：我选择了 East US 区域，因为这个区域也可以部署 OpenAI 的 ChatGPT 服务。

Microsoft Azure Search resources, services, and docs (G+/)

Home > Microsoft.CognitiveServicesAllInOne-20230408112525 | Overview

Deployment

Search

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name: Microsoft.CognitiveServicesAllIn... Start time: 4/8/2023, 11:26:57 AM
Subscription: OPENAI-UAT Correlation ID: bacde739-dee3-48b9-a3f3-974
Resource group: GEEKTIME-COURSE

Deployment details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

在创建认知服务完成之后，在部署成功的地方，点击Go to resource进入下一个界面

Home > Microsoft.CognitiveServicesAllInOne-20230408113235 | Overview > geektime-ai-course

geektime-ai-course | Keys and Endpoint

Cognitive services multi-service account

Search

Regenerate Key1 Regenerate Key2

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource Management

Keys and Endpoint

Pricing tier

Networking

Identity

Cost analysis

Properties

Locks

These keys are used to access your Cognitive Service API. Do not share your keys. Store them securely— for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1

KEY 2

Location/Region


eastus

Endpoint

https://geektime-ai-course.cognitiveservices.azure.com/


点击左侧的 Keys and Endpoint，然后点击右边的「复制」图标能够拿到对应的 API KEY

在拿到 API Key 之后，我还是建议你把 API Key 设置到环境变量里面。避免你使用 Notebook 或者撰写代码的时候，不小心把自己的 Key 暴露出去，被别人免费使用。同样的，我们也在环境变量里设置一下我们使用的 Azure 服务的区域 eastus。

 复制代码

```
1 export AZURE_SPEECH_KEY=YOUR_API_KEY
2 export AZURE_SPEECH_REGION=eastus
```


当然，也不要忘了安装对应的 Python 包。

 复制代码

```
1 pip install azure-cognitiveservices-speech
```

基本的语音合成

账号和环境都设置好了之后，我们就可以动手来试试 Azure 语音合成的效果了。

 复制代码

```
1 import os
2 import azure.cognitiveservices.speech as speechsdk
3
4 # This example requires environment variables named "SPEECH_KEY" and "SPEECH_REGION"
5 speech_config = speechsdk.SpeechConfig(subscription=os.environ.get('AZURE_SPEECH_KEY'), region=os.environ.get('AZURE_SPEECH_REGION'))
6 audio_config = speechsdk.audio.AudioOutputConfig(use_default_speaker=True)
7
8 # The language of the voice that speaks.
9 speech_config.speech_synthesis_voice_name='zh-CN-XiaohanNeural1'
10
11 speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config, audio_config=audio_config)
12
13 text = "今天天气真不错，ChatGPT真好用。"
14
15 speech_synthesizer.speak_text_async(text)
```

运行上面这个代码，你就会听到一个女声说：“今天天气真不错，ChatGPT 真好用。”

这几行代码非常简单。

我们先通过配置读取了 API Key 和 Region。

然后通过 `speech_synthesis_voice_name` 这个配置参数指定了我们合成语音所使用的声音。

通过 `speak_text_async` 这个函数，就能异步调用 API 服务，直接把合成的声音播放出来了。


通过 `speech_synthesis_voice_name` 这个参数，我们还可以选用很多别的声音，包括不同语言和不同的人。对应的列表可以在 Azure 的 [🔗 Language and voice support](#) 文档里面找到。我们换一个其他的 `voice_name`，就可以把对应的语音换成男声。

 复制代码

```
1 speech_config.speech_synthesis_voice_name='zh-CN-YunfengNeural'  
2 speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config, aud  
3 speech_synthesizer.speak_text_async(text)
```

指定语音的风格与角色

如果你仔细看了 [🔗 Language and voice support](#) 的文档，你会发现它有很多很多 `voice_name`。而且很多 `voice_name` 里，我们还有额外的两个参数可以选择，那就是 `Styles` 和 `Roles`，它们分别代表了合成语音的语气和对应的角色。通过这两个参数，我们可以让 AI 把很多场景“演出来”。比如，下面的示例代码就演绎了一段母子之间关于买玩具的一段对话，你可以运行一下看看效果。

 复制代码


```
1 ssm1 = """<speech version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"  
2     xmlns:mstts="https://www.w3.org/2001/mstts" xml:lang="zh-CN">  
3     <voice name="zh-CN-YunyeNeural">  
4         儿子看见母亲走了过来，说到：  
5         <mstts:express-as role="Boy" style="cheerful">  
6             “妈妈，我想要买个新玩具”  
7         </mstts:express-as>  
8     </voice>  
9     <voice name="zh-CN-XiaomoNeural">  
10        母亲放下包，说：  
11        <mstts:express-as role="SeniorFemale" style="angry">  
12            “我看你长得像个玩具。”  
13        </mstts:express-as>  
14    </voice>
```



```
15 </speak>""
16
17 speech_synthesis_result = speech_synthesizer.speak_ssml_async(ssml).get()
```

Azure 并不是通过让你在 API 里面配置一些参数来指定一段文本的角色和语气，而是通过一个叫做 SSML 格式的 XML 文件做到这一点的。这个 SSML 是 Speech Synthesis Markup Language 的首字母缩写，翻译过来就是**语音合成标记语言**。它不是一个 Azure 云专属的格式，而是一个 W3C 的标准，所以同样的 XML 不仅可以用在 Azure 云里，也一样可以用在 Google Cloud 里。

通过 SSML 里面元素的属性配置，我们可以指定不同文本段的 voice_name、role 和 style。比如，在上面的这个例子里面，我们就用两个 voice 元素，表示了两个不同的人的声音。voice 元素里面的 name 属性，指定了这段声音的 voice_name。而在 voice 元素内部，你还可以内嵌 mstts:express-as 元素，在这个元素里我们可以指定 role 和 style。这样一来，我们就可以让一个 voice_name 在不同的场景片段下，用不同的语气和角色来说话。

 复制代码


```
1 ssml = <code>"""<code><speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
2     xmlns:mstts="https://www.w3.org/2001/mstts" xml:lang="en-US">
3     <voice name="en-US-JennyNeural">
4         <mstts:express-as style="excited">
5             That'd be just amazing!
6         </mstts:express-as>
7         <mstts:express-as style="friendly">
8             What's next?
9         </mstts:express-as>
10    </voice>
11 </speak>""</code>
12
13 speech_synthesis_result = speech_synthesizer.speak_ssml_async(ssml).get()
```

在我自己实际使用的体验里面，中文的语气和角色效果不算明显。但是英文的效果还是很明显的，你可以根据文档用不同的参数尝试一下。

SSML 这个格式，不只支持 style 和 role，还有更多丰富的参数可以配置，你可以去看看 Azure [文档的协议标准](#)。

指定语音的输出方式

到目前为止，我们都是使用异步调用的方式，直接把语音播放出来了。但很多时候，我们可能需要把对应的语音存储下来。那下面的代码就可以做到这一点。

 复制代码


```
1 speech_config.speech_synthesis_language='zh-CN'
2 speech_config.speech_synthesis_voice_name='zh-CN-XiaohanNeural'
3
4 audio_config = speechsdk.audio.AudioOutputConfig(filename="./data/tts.wav")
5
6 speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config, aud
7
8 text = "今天天气真不错，ChatGPT真好用"
9 speech_synthesizer.speak_text_async(text)
```

我们只需要把原先设置成 `use_default_speaker=True` 的 `AudioOutputConfig`，改为设置成一个 `.wav` 的输出文件就好了。我们之后调用 `speak_text_async` 的函数，就会把语音输出到相应的 `.wav` 文件里。

 复制代码

```
1 audio_config = speechsdk.audio.AudioOutputConfig(use_default_speaker=True)
```

当然，你可以把对应的语音，暂时放在内存里面，而不是存储到文件系统中，也可以把输出的内容通过我们习惯的 MP3 格式存储下来。

 复制代码

```
1 speech_config.set_speech_synthesis_output_format(speechsdk.SpeechSynthesisOutputF
2
3 speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config, aud
4 result = speech_synthesizer.speak_text_async(text).get()
5 stream = speechsdk.AudioDataStream(result)
6
7 stream.save_to_wav_file("./data/tts.mp3")
```



我们只需要给 `speech_config` 这个参数设定一个 `synthesis_output_format` 就好了。我们上面就是把输出格式设置成了一个 48kHz 采样、192K 码率的 MP3 格式。然后，这一次我们把 `AudioConfig` 设置成了 `None`。在 `speak_text_async` 函数被调用之后，我们又调用了一下 `get` 函数，拿到对应的 `SpeechSynthesisResult` 对象。然后把这个对象放到 `AudioDataStream` 里，之后我们就可以把这个 `AudioDataStream` 按照我们的需要进行处理了。这里，我们是直接把它存储成了一个 MP3 文件。

使用开源模型进行语音合成

虽然通过 Azure 云的 API，我们可以很容易地进行语音合成，速度也很快。但很多时候因为数据安全的问题，我们还是希望能够直接在我们自己的服务器上进行语音合成。当然，这也是能够办到的，有很多开源项目都支持语音合成。


我们在这里，就不妨试一下百度开源的 PaddleSpeech 的语音合成功能，看看效果怎么样。

我们还是要先安装 PaddleSpeech 相关的 Python 包。

 复制代码

```
1 %pip install paddlepaddle
2 %pip install paddlespeech
```

然后通过 PaddleSpeech 自带的 `TTSExecutor`，可以将对应的文本内容转换成 WAV 文件。需要注意，这个过程中，PaddleSpeech 需要下载对应的模型，所以第一次运行的时候也要花费一定的时间。


 复制代码

```
1 from paddlespeech.cli.tts.infer import TTSExecutor
2
3 tts_executor = TTSExecutor()
4
5 text = "今天天气十分不错，百度也能做语音合成。"
6 output_file = "./data/paddlespeech.wav"
7 tts_executor(text=text, output=output_file)
```

PaddleSpeech 的 TTSExecutor，只是把你的文本输入转化成了一个 WAV 文件。要在 Python 里面播放对应的声音，我们还要借助于 PyAudio 这个包。对应的，我们要先安装 PyAudio 依赖的 portaudio 库，然后再安装 PyAudio 包。


Mac 下可以通过 homebrew 来安装 portaudio。

```
1 brew install portaudio
```

 复制代码


如果在 Ubuntu 或者 Debian 下，你可以通过 apt-get 来安装 portaudio。

```
1 sudo apt-get install portaudio19-dev
```

 复制代码


只有在 portaudio 安装成功之后，我们才能安装 PyAudio 包，不然会报缺少依赖的错误。

```
1 pip install pyaudio
```

 复制代码

通过 PyAudio，我们可以直接播放 WAV 文件的内容了。对应的代码我放在下面了，其实我不太熟悉 PyAudio 库，但是这样简单的代码直接让 ChatGPT 帮我写，一次就能运行成功。如果你仔细读一下这段代码，也不难理解它的含义。实际就是打开了一个 PyAudio 的 Stream，然后不断从我们的 WAV 文件里面读入数据，然后写入这个 Stream，写入之后声音就播放出来了。如果你把 stream.write(data) 那一行去掉，那么你就会发现整个程序运行的过程里，是没有声音的。

```
1 import wave
2 import pyaudio
3
4 def play_wav_audio(wav_file):
5     # open the wave file
```

 复制代码

```

6     wf = wave.open(wav_file, 'rb')
7
8     # instantiate PyAudio
9     p = pyaudio.PyAudio()
10
11    # open a stream
12    stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
13                    channels=wf.getnchannels(),
14                    rate=wf.getframerate(),
15                    output=True)
16
17    # read data from the wave file and play it
18    data = wf.readframes(1024)
19    while data:
20        stream.write(data)
21        data = wf.readframes(1024)
22
23    # close the stream and terminate PyAudio
24    stream.stop_stream()
25    stream.close()
26    p.terminate()
27
28    play_wav_audio(output_file)

```

不过，我们调用的 PaddleSpeech 代码里的默认参数有一个小问题，就是它只支持中文的语音合成。如果你的文本带上英文运行一下，你会发现合成的语音里面只有中文，没有英文。

 复制代码

```


1 tts_executor = TTSExecutor()
2
3 text = "今天天气十分不错, Paddle Speech也能做语音合成。"
4 output_file = "./data/paddlespeech_missing.wav"
5 tts_executor(text=text, output=output_file)
6
7 play_wav_audio(output_file)

```

运行上面的代码，你会发现，PaddleSpeech 在合成的语音里面丢失了。

这是因为，PaddleSpeech 默认情况下使用的是一个只支持中文的模型。我们可以通过一些参数来指定使用的模型，一样能够做中英文混合的语音合成。

对应的代码：

 复制代码

```
1 tts_executor = TTSExecutor()  
2  
3 text = "早上好, how are you? 百度Paddle Speech一样能做中英文混合的语音合成。"  
4 output_file = "./data/paddlespeech_mix.wav"  
5 tts_executor(text=text, output=output_file,  
6             am="fastspeech2_mix", voc="hifigan_csmsc",  
7             lang="mix", spk_id=174)  
8  
9 play_wav_audio(output_file)
```

可以看到，和上面的代码相比，我们增加了 4 个参数。

am，是 acoustic model 的缩写，也就是我们使用的声学模型。我们这里选用的是 fastspeech2_mix。fastspeech2 也是一个基于 Transformer 的语音合成模型，速度快、质量高。这里带了一个 mix，代表这个模型是支持中英文混合生成的。

voc，是 vocoder 的缩写，叫做音码器。声学模型只是把我们的文本变成了一个声音波形的信号。我们还需要通过音码器，把声学模型给出的波形变成可以播放的音频。我们这里选择的 HiFiGAN_csMSC，是一个高保真（HiFi）、基于对抗生成网络（GAN）技术的模型，它的训练数据用到了 HiFiSinger 和 csMSC，而模型的名字就来自这些关键词的组合。

lang，代表我们模型支持的语言，这里我们自然应该选 mix。

spk_id，类似于我们之前在 Azure 里看到的 voice_name，不同的 spk_id 听起来就是不同的人说的话。

运行这个代码，一样能够正常地生成中英文混合的语音内容。如果你想要了解 PaddleSpeech 的语音合成功能，还有它所支持的各种模型和各种应用场景，可以参看 GitHub 上的 [🔗 Demo 文档](#)。

小结

好了，这一讲到这里也就结束了。

这一讲我们学会了两种语音合成的方式。一种是使用 Azure 云提供的 API，另一种则是使用百度开源的 PaddleSpeech。Azure 云的语音合成，不仅仅是能简单地把文本变成人声，还能通过 SSML 这个 W3C 标准下的 XML 标记语言，指定不同的人声 (voice_name)、语气 (style) 还有角色 (role)。这些功能都是非常有实用价值的，能够帮助我们处理各种场景下的语音合成需求。

而 PaddleSpeech 则带给了我们一个开源方案，并且它也支持中英文混合在一起的语音生成。它背后可供选择的模型里，我们使用的也是基于 Transformer 的 fastspeech2 模型。可以看到，目前 Transformer 类型的模型在各个领域都已经占据了主流。

学到这里，我们的 AI 就拥有了声音。而在下一讲里，我会拿我们已经学到的知识，搭建一个可以通过语音和你聊天的机器人。并且更进一步地，我们还会为它配上你的虚拟形象，希望你和我一样对下一讲充满期待！

思考题

最后，给你留一道思考题。PaddleSpeech 不仅能拿来做语音合成，也能用来做语音识别。你能试试看用它做语音识别的效果吗？和 OpenAI Whisper 比起来，你觉得它们两个哪个效果更好？欢迎你把你体验之后的感受分享出来，也欢迎你把这一讲分享给需要的朋友，我们下一讲再见！

推荐阅读

PaddleSpeech 的 [🔗 中文文档](#) 里面，包含了大量使用 PaddleSpeech 这个开源库的场景和方法，你有兴趣可以去看一看。百度的 PaddlePaddle 的社区里面，也有专门的 [🔗 语音相关的课程](#)，如果你想深入了解的话，也可以去看一看。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (5)



胡萝卜

2023-04-24 来自上海

这个英文转语音效果不错 <https://github.com/suno-ai/bark>

作者回复: 👍 bark是最近比较火的一个模型



👍 3



Toni

2023-04-23 来自瑞士

尝试用百度的 PaddlePaddle, 将语音文件(.wav) 转换成文字(语音识别)。

1. 语音中只含中文, 实现代码如下:

```
from paddlespeech.cli.asr.infer import ASRExecutor
asr = ASRExecutor()
audio_file="./data/BaiduTTS/zh.wav"
result = asr(audio_file=audio_file)
print(result)
```

输出结果:

我认为跑步最重要的就是给我带来了身体健康

语音原文:

我认为跑步最重要的就是给我带来了身体健康

2. 语音为中英文混合的文件 "./data/BaiduTTS/paddlespeech_mix_1.wav", 用上面的代码运行

输出结果:

早上好哈沃尔姨百度他都斯一样能做中英文混合的语音合成

语音原文:

早上好, how are you? 百度 Paddle Speech 一样能做中英文混合的语音合成

处理中英文混合的语音文件, 进行语音识别时, 需要给 ASRExecutor() 添加参数, 代码如下:

```
from paddlespeech.cli.asr import ASRExecutor
asr = ASRExecutor()
audio_file="./data/BaiduTTS/paddlespeech_mix_1.wav"
```

```
result = asr(model='conformer_talcs', lang='zh_en', codeswitch=True, sample_rate=16000,
audio_file=audio_file, config=None, ckpt_path=None, force_yes=False)
print(result)
```

输出结果:

早上好 how are you 百度它读 speech 一样能做中英文混合的语音合成

对照语音原文, ASRExecutor() 将语音 "百度 Paddle Speech" 转成了 "百度它读 speech", 并不完美。

期待更好的解决方案。

参考:

【PaddleSpeech】一键预测, 快速上手Speech开发任务

<https://aistudio.baidu.com/aistudio/projectdetail/4353348?sUId=2470186&shared=1&ts=1660878142250>

一文读懂 PaddleSpeech 中英混合语音识别技术

<https://xie.infoq.cn/article/c05479afe4291255d91ed950f>

Load specified model files for TTS cli #2225

<https://github.com/PaddlePaddle/PaddleSpeech/issues/2225>

PaddlePaddle/PaddleSpeech

https://github.com/PaddlePaddle/PaddleSpeech/blob/develop/demos/audio_tagging/README_cn.md

作者回复: 👍



👍 1



劉仲仲

2023-05-08 来自美国

老师, 为甚么我用Azure语音服务, 在jupyter notebook上已经跑通而且可以播放声音, 但是一部署到hugging face上面就发不出声音呢

作者回复: 看看是否浏览器权限设置不能播放声音?



Steven

2023-04-25 来自辽宁

补充 Windows 下安装 portaudio 库:

1, 下载安装 MSYS2:

<https://www.msys2.org/>

2, MSYS2 安装完成后在其命令行窗口中执行:

```
pacman -S mingw-w64-x86_64-portaudio
```

作者回复: 👍



zhihai.tu

2023-04-21 来自上海

openai本身没有tts的api吗? 期待下一讲。

作者回复: OpenAI没有, 语音是相对比较成熟的领域, 各种开源和商业的方案很多, 价格也比较便宜。

共 3 条评论 >

