

爱酱：一个基于多线程的聊天机器人

曹胜操

叶天

孙之清

北京大学信息科学技术学院

1500012783@pku.edu.cn

github.com/Edward-Sun/Robot

bilibili.com/video/av18617481

摘要

爱酱是一个基于多线程的即时语音聊天机器人。我们在设计中采用了独特的多线程通讯技术，使得我们可以做到语言打断与图形交互。此外，由于该机器人的核心代码仅为 200 多行，因此对爱酱的深度定制也十分简单，我们希望我们的语言机器人能起到作为其他组的即时语音聊天机器人的一个很好的 baseline 的作用，为进一步的作业效果评估起到便利。

1 功能列表

该聊天包含了如下五种主要功能：

- 基本的聊天功能，根据用户所说的话，作出相应的语音答复
- 机器人可以在自己说话时根据用户的语音主动打断语音答复
- 机器人可以将自己的语音答复和用户的语音区分开来
- 机器人可以自助判断用户开始说话和结束说话的时间并给出回应
- 实现GUI，在屏幕上显示一个萌妹的形象。当机器人在听用户说话时，爱酱会眨眼睛；当机器人在作语音答复时，爱酱会有随机的口型变化；当用户将要关闭程序时，爱酱会做出挥手再见的动作。

2 系统设计

我们将在这个部分对爱酱的系统设计进行详细的解释。

我们选择使用不同的进程来实现机器人的各种功能。进程的创建、调度和管理可以通过python里的API有效地实现。尽管在原始的设计中，我们计划使用线程而不是进程，因为线程间的切换、通信效率比进程高。但是考虑到整个程序的时间瓶颈其实在于网络通信，使用线程或进程对最终的效果影响很小。

2.1 进程设计

接下来简要介绍我们所使用的进程的种类以及它们执行的操作。我们实现的聊天机器人主要有以下6种进程：

main进程 程序的主进程，负责程序的初始化操作，起到主要的控制作用。具体包括创建其他各类进程的队列，创建初始的GUI进程、listen进程、manage进程，发现输入音频中的用户声音并创建相应的analyze进程等

manager进程 负责控制speak进程，实现音频的播放和打断。

listen进程 负责监听输入的声音。调用pyaudio库，不断监听麦克风输入的声音，并将监听到的声音数据保存在voice队列里，提供给main进程使用。

analyze进程 负责对用户音频的分析。将main进程提取到的用户的有效音频数据写入question.wav文件。调用外部程

序question.exe，该程序将question.wav文件里的音频转化为文字并保存在question.txt中。读取question.txt，将文本内容输出在命令行里，同时放入speak队列中，提供给speak进程使用。

speak进程 负责获取机器人的回答以及机器人的语音播放。将用户语音的文本发送到图灵机器人的服务器端，获得机器人回答的文本，将文本写入answer.txt文件。调用外部程序answer.exe，该程序将answer.txt文件里的文本转化为音频并保存在answer.wav中。读取answer.wav，将音频内容播放出来。

GUI进程 负责控制屏幕上的爱酱的动作和表情。GUI进程始终活跃，根据程序运行的状态，作出眨眼、动嘴、招手等动作。

2.2 进程间通信

在整个程序中，进程间的通信是必要的。原因主要有以下两点：

- 诸如用户语音的文本、机器人的回答的文本等数据，是需要多种进程共享的。
- 各个进程之间需要进行同步和协调，防止出现例如两个speak进程同时试图做出回答的情况。

为了满足上述需求，我们对应使用了以下三种进程间通信的方式：

- 共享文件。我们以wav和txt文件的形式，保存用户的语音及其对应的文本、机器人的回答及其对应的文本，从而使不同进程都能共享地访问这些数据。同时，文件也是利用外部程序对数据作进一步处理的有效的方式。
- 队列。我们为用户语音、GUI的动作等信息分别创建了队列，使不同的进程能访问这些队列，从而实现这些数据的方便快捷的共享。

- 互斥锁。我们在程序中使用PV，用来实现各个speak进程之间的相互协调，防止多个speak进程同时试图说话的情况。

3 使用的API和库

3.1 科大讯飞

讯飞开放平台是一个以语音交互为核心的人工智能开放平台，它提供了优秀的语音听写、语音合成技术，可以为我们所用。讯飞开放平台提供了一套完善的SDK，利用其中的接口我们就可以编写自己的程序，实现语音和文本之间的转换。在聊天机器人运行的过程中，完成了对音频的基本处理之后，我们就可以把音频发送给平台，从平台接收语音听写得到的文本。经过其它的处理之后得到应答的文本，我们再使用另一套类似的方式将文本发送给平台，得到语音合成的音频。

我们通过调用question.exe和answer.exe程序，来完成讯飞平台的使用，实现将用户的音频转化为文本，以及将机器人回答的文本转化为音频的功能。

3.2 图灵机器人

图灵机器人是一个基于自然语言处理等技术的平台，可以提供具有流畅自然的中文对话回答功能的机器人，有着准确分析语义、学习上下文场景的能力。图灵机器人提供了一套简洁易用的网络API，通过发送json请求就可以得到机器人的回复。在聊天机器人运行的过程中，得到用户所说的文本之后，我们就可以通过网络请求利用该平台得到一个期望的回答文本，然后再进行后续的处理即可。该平台提供的机器人可以进行有上下文环境的聊天，这样就可以根据用户之前所说的历史信息来进行更准确的应答。此外，机器人还可以具有其它一些扩展功能，如数字计算、天气查询等等，给用户提供一些更丰富的体验。

3.3 爱酱爱酱

MikuMikuDance是一款轻量级的3D动画制作软件，在导入已经制作好的3D人物模型之后，就可以利用MMD制作出简单的3D动画。我们可以预先渲染好一些虚拟人物形象的基本动作形态，在聊天机器人运行的过程中，根据实时计算得到的文本和语音来动态展现虚拟人物形象，呈现出更加真实有趣的聊天效果。

4 使用方法

接下来，我们阐述从Github上 clone 下我们的项目后，如何让它跑起来。

4.1 运行环境

我们的程序需要在Windows系统下运行。由于我们将要进行语音形式的聊天，因此需要有麦克风和扬声器来分别输入和输出语音。又由于对语音和文字信息的处理要用到在线API，因此需要保证运行时有良好的网络环境。

我们所写的代码基于python3.5或更高的版本。现在假设已经安装好了python3.5，并且确保可以在Windows命令行下运行python程序。

4.2 相关python库的安装

在运行程序之前，需要安装python代码中使用到的库。安装方法如下：

打开Windows的命令行，

*输入命令 `pip install pyaudio`，安装 `pyaudio` 库

*输入命令 `pip install wave`，安装 `wave` 库

*输入命令 `pip install json`，安装 `json` 库

*输入命令 `pip install requests`，安装 `requests` 库

*输入命令 `pip install collections`，安装 `collections` 库

*输入命令 `pip install webtrcvad`，安装 `webtrcvad` 库

*输入命令 `pip install numpy`，安装 `numpy` 库

*输入命令 `pip install opencv-python`，安装 `cv2` 库

此时，准备工作已经完成。

4.3 开始运行

根据我们的经验，如果在python的IDE中运行此程序，会出现无法显示中间的输出结果以及响应时间较长等问题。因此，我们选择在Windows命令行下运行程序。

打开Windows的命令行，在main.py的相同目录下，输入命令`python main.py`，程序即可开始运行。

在程序运行的整个过程中，每当一个新的进程被创建，都会在命令行中输出一条信息，以便观察程序的运行情况。

程序刚开始运行时，可以看见程序创建了main进程、GUI进程和listen进程，此时机器人还未启动。用户需要对麦克风说出“开机”两个字后，机器人才会正式启动。

用户发出“开机”命令后，屏幕上会出现爱酱形象的动画人物。机器人首先会向用户打招呼，然后等待用户说话。

4.4 对话过程

用户对麦克风说话后，经过一小段时间的响应，命令行中会输出机器人识别的用户语音所转化成的文字。再经过一小段时间的响应，命令行中会输出机器人所要作的回答所对应的文字，然后机器人会开始说话。在机器人说话且尚未说完时，如果用户继续说话，机器人会停止说话，重新对用户说的话进行分析和回复。当然，这期间也会有一定的响应时间；以上所有响应时间，与当前计算机所在的网络环境关系很大。因此，若要获得更好的使用体验，需要尽量让网络环境保持良好。

4.5 关闭机器人

用户对麦克风说出“关机”两个字后，机器人会挥手说“再见”，然后爱酱消失。此时，

程序并没有停止，如果再次发出“开机”指令，机器人就会再次启动。若要结束程序，需要手动杀死程序。

5 工作量说明

我们使用python语言编写代码，自写的代码在main.py中，行数约为250行。除了代码之外，我们的工作量还包括机器人爱酱的声音音色的优化和调整，以及爱酱的动作和表情的制作等。

分工情况如下

- 曹胜操： answer.exe 与 question.exe的编写、 gui设计与实现
- 叶天： 代码测试；编写文档
- 孙之清： 多进程的设计与实现