

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 课程论文

THESIS OF CURRICULUM



项目名称: “你并不孤单”

小组名称: “老实芭蕉组”

组 员: 马筱琛、李嘉翌、王菀婷

指导老师: 鲍杨

# 目录

1 背景 .....	3
1.1 选题原因 .....	3
1.2 项目介绍及思路 .....	3
1.3 类库介绍 .....	3-5
1.3.1 面部识别相关 .....	3-4
1.3.2 语音机器人相关 .....	4
1.3.3 其他准备工作 .....	5
2 代码解释 .....	6-14
2.1 面部表情识别 .....	6-9
2.2 语音自动对话机器人 .....	9-14
2.3 最终实现 .....	14
3 局限与改进 .....	15
3.1 局限性 .....	15
3.2 优化设想 .....	15
4 小结 .....	16

# 1 背景

## 1.1 选题原因（王菀婷）

疫情期间，大学生们在家以线上形式开展学习。但很多时候学生的负担因为网课的形式而加重不少，精神压力也更大了，我们注意到，同学们在朋友圈里表达消极情绪的频率更高了，很多相识的朋友们在聊天交流中也出现了否定自己、怀疑自己的情感倾向。

缓解心理问题的方法之一就是尽量与别人沟通交流，开放自己的内心，疏导情绪。但很多时候情绪过度低沉的人们缺乏交流欲，甚至不愿意开口与亲友交流，当亲友忙碌时就更不用说了。

因此我们想利用面部表情识别、语音识别和自动对话创建一个相对智能的语音交谈助手，识别到使用者带有消极情绪的表情后会自动开启语音对话，简单地帮助使用者排遣消极情绪，激发使用者的交流欲，在他们情绪过于低落时鼓励使用者寻求帮助。

在这个项目中我们将人脸识别和语音识别、自动回复三项系统改编组合、进行优化，尽管我们的技术十分简单，但我们希望能以这个项目传达出我们的关心，表达“你并不孤单”的思想，帮助大家纾解消极情绪。

## 1.2 项目介绍及思路（李嘉翌、马筱琛）

先调用摄像头，将摄像头捕捉的图像实时进行分析，得出面部区域，并保存为所选图片，通过已有模型对所选图片是否表现悲伤进行判断，若识别到悲伤表情，则开始进行语音交互对话。被识别对象说的话，先通过录音保存为语音文件，然后调用百度 API 实现语音转文本 STT，再然后调用图灵机器人 API 将文本输入得到图灵机器人的回复，最后将回复的文本转成语音输出 TTS，这样即可实现和机器人的语音对话了。

注：当人脸识别到伤心表情后，会自动结束面部识别的程序，并转入语音对话；当被用户说出“再见”时，语音交互程序会自动终止。

## 1.3 类库介绍及准备工作

### 1.3.1 面部识别（李嘉翌）

**opencv:** 人脸识别部分需要对图像进行识别，而 OpenCV 是一个开源的跨平台计算机视觉库，实现了图像处理和计算机视觉方面的很多通用算法。我们使用 opencv 作为帮助我们对人脸进行可视化的工具。

**imutils:** 在人脸情绪识别的过程中，我们需要对所获取的图像进行处理（改变大小等），这个类库在 opencv 基础上对一些方法进行了再次加工，所以可以更加便捷的对图像数据进行处理。

**numpy:** 在代码实现过程中, 我们需要运用数组的计算等, 而 numpy 这种工具可用来存储和处理大型矩阵, 比 Python 自身的嵌套列表结构要高效的多, 因此我们使用了 numpy。

**keras:** keras 是一个由 Python 编写的开源人工神经网络库, 可以进行深度学习模型的设计、调试、评估、应用和可视化。在人脸情绪识别方面可以利用模型对数据进行分析从而判断表情。

**time:** 这是 python 自带的一个关于时间的类库, 因为当检测出 sad 的情绪进入下一个语音识别的阶段, 而检测速度过快容易导致表情识别出现误差, 因而我们利用 time 库计算 sad 表情的保持时间, 从而减小误差。

### 1.3.2 语音自动聊天机器人 (马筱琛、王菀婷)

**speech\_recognition:** 在语音识别过程中, 由于电脑无法直接分析用户的语音, 故需要通过 speech\_recognition 包将用户的语音记录下来, 并保存到音频文件 (.wav 格式) 中, 以便下一步操作。该包在识别用户的语音时, 无需构建访问麦克风和从头开始处理音频文件的脚本, 只需几分钟即可自动完成音频输入、检索并运行, 同时代码较为简单, 因此易用性很高。

**pyaudio:** 录音接口。使用这个库可以进行录音, 播放, 生成 wav 文件等等。PyAudio 提供了 PortAudio 的 Python 语言版本, 这是一个跨平台的音频 I/O 库, 使用 PyAudio 可以在 Python 程序中及各种平台上播放和录制。

**baidu-aip:** 在将音频转文字的过程中, 我们需要调用百度语音识别的 API 接口, 因此会用到 baidu-aip 这个第三方类库, 它可以实现音频文件的读取、识别, 并将其转换为文字。

**requests:** requests 库是 python 中非常优秀的第三方库, 在调用图灵机器人, 使其为用户的话做出自动回复时

**json:** 这是 python 自带的一种轻量级的数据交换格式, 采用完全独立于编程语言的文本格式来存储和表示数据, 不仅容易阅读和编写, 而且也容易解析和生成, 有效的提升了网络传输效率, 在与后端的数据交互中有较为广泛的应用。

**pyttsx3:** 一个非常简单的文本到语音的转换库, 可以脱机工作, 支持多种 TTS 引擎 (sapi5、nsss、espeak), 通过这个库可以非常方便的将文字转换成语音。









**win32com.client:** 与 pyttsx3 一样, 也能实现文本到语音的转换, 而且代码更加简洁, 但是只有 Windows 系统能用。

### 1.3.3 其他准备工作（马筱琛）

**准备百度 API:** 在语音转文字的过程中,我们会用到百度语音识别 API 接口,因此我们需要首先登录百度 AI 开放平台 <https://ai.baidu.com/tech/speech/asr>,注册账号并创建应用,记住相应的 App ID、API Key、Secret Key,以便后续调用。并通过阅读平台上语音识别的相关技术文档,了解如何在 python 中调用 API 接口。

应用列表					
<a href="#">+ 创建应用</a>					
应用名称	AppID	API Key	Secret Key	创建时间	操作
1 group	20028567	G61DNXaoRBdexGzKn1Enc6bu	***** <a href="#">显示</a>	2020-05-23 18:47:26	<a href="#">报表</a> <a href="#">管理</a> <a href="#">删除</a>

**准备图灵机器人:** 图灵机器人大脑具备强大的中文语义分析能力,可准确理解中文含义并作出回应,是最擅长聊中文的机器人大脑。因此,在搭建一个新的自动回复机器人较为困难的情况下,我们选择运用已有的图灵机器人,并通过调用相关接口,帮我们实现自动回复功能。登录图灵机器人官网 <http://www.turingapi.com/>,注册账号并创建一个图灵机器人,并记住机器人的 apikey。注意保持密钥开关关闭,否则调用时会报错。此外,由于本项目的目的是安抚心情失落的人,让他们不再感到自己是孤立无援的,而默认的自动回复系统满足不了这个要求,所以我们还需要在平台上自己创建私有语料库,使图灵机器人回复的内容更加贴合我们的要求。下面是一些系统自带功能。

 笑话大全 <input checked="" type="checkbox"/>	 故事大全 <input checked="" type="checkbox"/>	 成语接龙 <input checked="" type="checkbox"/>	 新闻资讯 <input checked="" type="checkbox"/>	 星座运势 <input checked="" type="checkbox"/>	 脑筋急转弯 <input checked="" type="checkbox"/>
 歇后语 <input checked="" type="checkbox"/>	 绕口令 <input checked="" type="checkbox"/>	 顺口溜 <input checked="" type="checkbox"/>			

## 2 代码解释

### 2.1 面部表情识别（李嘉翌）

#### （1）步骤 1：导入所需的类库

```
import cv2 as cv
import imutils
import numpy as np
import keras
import time
```

我们使用 imutils，它是一个图像处理的工具集合，利用 numpy 库对数组进行计算，从而方便代入，keras 是一个已有的人工神经网络数据库，帮助我们对人脸进行识别（开源），而 opencv 则是一个计算机视觉库，作为帮助我们对人脸进行可视化的工具。（经过测试，有时脸歪无法识别）

#### （2）步骤 2：定义变量和数据的获取

```
p=0 #定义一个判断是否开始计时的一个变量，保证开始只能进入starttime的运算
lasttime=0 #定义持续时间，并提供初始值，防止变量出错
model_path = "./model/CNN-Emotion-Model"
#引用已有的表情模型，借助他对所获得的图片信息进行表情的分析
face_path = cv.harcascades + 'haarcascade_frontalface_alt.xml'

emotion_label = ['Anger', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
#定义心情的标签
```

haarcascade\_frontalface\_alt2.xml 是 opencv 官方训练好的一个人脸识别器，属于 opencv 自带，该文件中会描述人脸的 Haar 特征值。Haar（全面 Haar-like features，是用于物体识别的一种数字图像特征。）特征值的用途可用来描述人脸、眼睛、嘴唇或是其它物体。我们导入他方便对于面部进行分析。

```
cap = cv.VideoCapture(0)
# 调用摄像头

face_model = cv.CascadeClassifier()
#创建一个新的CascadeClassifier文件（级联分类器）
```

级联分类器类的检测框架简而言之就是一个多尺度缩放+滑动窗口遍历搜索的框架。具体应用在后边的实例中会有所展示。

```
if not face_model.load(face_path):
    print('load face model failed')
    exit(0)
#从文件路径（即前面导入的自带的分类器的路径中）载入分类器文件，
# 如果没有成功载入则显示失败后退出

emotion_model = keras.models.load_model(model_path)
# 调用训练过的模型
```

(3) 步骤 3: 定义函数, 并为第二个函数的定义做准备 (第二个里面使用第一个进行框选和显示)

```
def drawFace(frame, frame_grey):
    global p #定义全局变量p

    faces = face_model.detectMultiScale(frame, minSize=(50,50))
    #此处检测出图片中所有的人脸, 并用向量保存各个人脸的坐标、大小 (用矩形表示)

    emotion = np.array([[0,0,0,0,0,0,0]])
    #创建数组, 将输入的数组转化为矩阵格式, 用来对应表情的各项参数

    for a,b,c,d in faces:
        #a, b对应识别出的人脸的左下角坐标的横纵坐标,
        # c, d表示宽和长 (不是长和宽) 是前面detectMultiScale保存的数据
        face = frame_grey[b:b+d,a:a+c]
        #定义人脸区域, frame_grey参数会在下一个函数中获得值(灰度图像),
        # (上文已经说过这个函数是为下一个函数做基础) 划分两个区域即长从b到b+d, 宽从a到a+c
```

定义全局变量 p, 是防止出现 UnboundLocalError: local variable 'p' referenced before assignment, 即无法确定是全局变量还是局部变量的问题。

detectMultiScale 这个函数被称作多尺度检测, minsize 保证人脸最小大小, 小于该值的对象会被忽略, frame 是第一个变量, 对应参数 image, 此处检测出图片中所有的人脸, 并用向量保存各个人脸的坐标、大小 (用矩形表示)

```
face = cv.resize(face, (48, 48))
#使函数确定到一个固定的大小 (模型默认数据)
#第一个参数是原图像, 第二个参数为dsize: 输出图像的大小。
#如果这个参数不为0, 那么就代表将原图像缩放到这个size(width, height)指定的大小
face = np.array(face)
face = face.reshape(1, 48, 48, 1)
#改变前面确定的数组形状

emotion = emotion_model.predict(face/255)
#运用已有的模型中的预测函数进行预测, 并保存相关数据
text = emotion_label[np.argmax(emotion)]
#按照对应的预测得到的表情所对应的数组显示表情的标签
```

按照对应的预测得到的表情所对应的数组显示表情的标签: emotion 得到的数据为一个 7 个元素对应的向量 (对应 7 个情绪的数据), 通过 np.argmax 方法比较出各个这个向量中的 7 个数字哪个最大, 判断出以第几个作为当前情绪的预测, 然后选择里 emotion\_label 中对应位置的情绪。

```
if text != 'Sad' and p == 1:
    endtime = time.time()
    lasttime = endtime-starttime
if text == 'Sad' and p == 0:
    starttime = time.time()
    p=1
```

这个部分我们对于表情显示为 Sad 的时间进行检测, 防止偶然表情的显示导致的判断错误, 通过对开始和结束两个时间节点的差获得处于 Sad 情绪的时间, 减小误差。

```

cv.rectangle(frame, (a,b), (a+c, b+d), (0, 0, 255), 1)
# 框选对应区域
# 第一个参数为img, 即图像,
# 第二个参数为矩形一个顶点的坐标,
# 第三个为所取的第一个顶点的对角线上的另一个坐标,
# 第四个为线条颜色,
# 第五个为框线的粗细
cv.putText(frame, text, (a,b-10),cv.FONT_HERSHEY_TRIPLEX, 0.8, (0,0,255))
# 第一个参数为图像(即插入哪个图片),
# 第二个为文本, 此处直接用上面text中获得的表情标签的文本,
# 第三个是文本框的左下角的坐标,
# 第四个为字体, 查阅后发现8种opencv支持的字体, 我们随意选择了其中之一,
# 后面几个变量依次为字体大小, 颜色, 此处不做过多赘述

return emotion
#返回表情的数据

```

(4) 步骤 4: 定义运行的主体函数, 一方面调用图像并框选, 另一方面保证表情数据的可视化

```

def start():
    # 确认已调用实时的摄像头的数据

    while True:
        if not cap.isOpened():
            print('fail to open camera')
            break

```

防止特殊情况 cv.VideoCapture(0), 没有打开摄像头, 若调用摄像头失败则退出。

```

frame = cap.read()[1]
# 读取的是所捕捉到的摄像头中的视频的每一帧图像
frame = imutils.resize(frame, width=300)
# 读取当前的图像数据, 同时修改图像的大小
# 一般改变图像大小, 但保持原来图像的长宽比不变, 此处只对宽度进行修改

```

```

prob = np.ones((300, 500, 3), dtype='uint8')

```

ones() 返回一个全 1 的 n 维数组, 同样也有三个参数, 这里用了其中两个: shape (用来指定返回数组的大小, 此处返回一个, 其中第一个参数的 (x, y, z) 分别对应: x 对应这个数组由 x 个数组纵向排列构成, y 和 z 为这 x 个数组的长和宽)、dtype (数组元素的类型) (uint8 是 8 位无符号整型)

```

frame_grey = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)

```

为了能够便于处理图像, 将 frame 图像转化为灰度图, 参数为 CV\_RGB2GRAY 与 CV\_BGR2GRAY, 任选其中一个。



```
emotion = drawFace(frame, frame_grey)
```

调用准备的第一个 drawFace 函数，emotion 的值为 return 获得的 emotion 数据（7 个数据的数组）

```
sum = emotion.sum()
#对emotion数据求和，同时防止出现运行错误，即没有识别到人脸时各项数据全部为0
if not sum == 0:
    emotion = emotion / sum
for i in range(0, 7):
    cv.putText(prob, emotion_label[i], (10, (i + 1) * 35 + 10), cv.FONT_HERSHEY_TRIPLEX, 0.5, (255, 255, 255))
    t = 90 + int(emotion[0][i] * 300)
    cv.rectangle(prob, (90, i * 35 + 25), (90 + int(emotion[0][i] * 300), (i + 1) * 35 + 30), (0, 255, 0), -1)
    #此处大部分和上文相同，特别注意，线型为负值，所以会填满整个矩形，所以看到的为一整块矩形
    cv.putText(prob, str(int(emotion[0][i] * 100)) + '%', (t, (i + 1) * 35 + 10), cv.FONT_HERSHEY_TRIPLEX, 0.5, (255, 255, 255), 1)
    #cv.putText的参数与上文相同，此处文字为白色
    #数组array的格式: array[0][i]:意思为，提取这个数组中第一个数组的第i+1个元素
    #此处分别显示标签名，用矩形图像表示的数据，和概率百分比
```

```
cv.imshow('probability', prob)
cv.imshow('video', frame)
#imshow函数，把图像显示在窗口中，此处分别显示各种表情（情绪）的概率和已框选的人脸的图像

if cv.waitKey(1) == ord('q'):
    break
if lasttime >= 5:
    #确认保持难过的时间是否达到一定数值，确保表情检测准确
    break
if lasttime < 5:
    #如果没有达到一定数值就结束，则重新进入循环并把p变量恢复，从而使得先从starttime开始计时
    p=0
cap.release()
#如果在程序中按q则退出，（退出中文输入法的情况下）
```

最后

```
start()
#使用这个函数即可
```

## 2.2 语音自动对话机器人

（0）步骤 0：导入所需要的类库(马筱琛)

```
from aip import AipSpeech
import requests
import json
import speech_recognition as sr
import pyttsx3 #import win32com.client
                #Windows用户还可以用这个类库，效果与pyttsx3相同
```

### (1) 步骤 1: 语音生成音频文件 (马筱琛)

```
# 1、语音生成音频文件, 录音并以当前时间戳保存到voices文件中
# Use SpeechRecognition to record 使用语音识别录制

def my_record(rate=16000): #录音参数必须满足 16k 采样率
    r = sr.Recognizer() #实例化一个识别器r
    with sr.Microphone(sample_rate=rate) as source: # 打开麦克风
        print("please say something")
        audio = r.listen(source) #通过麦克风进行录音
    with open("voices/myvoices.wav", "wb") as f: #设置文件名, 类型
        f.write(audio.get_wav_data()) #将录音数据转换成wav格式写入文件
```

在这里我们使用的包是 `speech_recognition`, 它可以将我们的话录制下来并保存为音频文件。需要注意, 由于音频文件是二进制格式, 所以在使用 `open()` 函数时, 要用 “wb” 模式。

### (2) 步骤 2: 音频文件转文字 (马筱琛)

```
# 2、音频文件转文字: 采用百度的语音识别python-SDK
# 导入我们需要的模块名, 然后将音频文件发送出去, 返回文字。
# 百度语音识别API配置参数
APP_ID = '20028567'
API_KEY = 'G61DNXaoRBdexGzKn1Enc6bu'
SECRET_KEY = 'd86Zd9ZCeGYH7YIDhRRRe3bAbIZxga1y'
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
path = 'voices/myvoices.wav'
```

这里用到的是我们从百度语音识别平台创建的接口, 我们通过配置具体参数来使用它, 以便完成语音转文字的工作。需要的类库为 `baidu-aip`

```
# 将语音转文本STT
def listen():
    # 读取录音文件
    with open(path, 'rb') as fp:
        voices = fp.read()
    try:
        result = client.asr(voices, 'wav', 16000, {'dev_pid': 1537, })
        result_text = result["result"][0]
        print("you said: " + result_text)
        return result_text
    except KeyError:
        print("KeyError")
        engine.say("我没有听清楚, 请再说一遍...")
        engine.runAndWait()
```

通过定义 `listen()` 函数, 我们实现了读取录音文件, 识别语音内容, 并转换为文字的过程, `asr()` 函数中的 `dev_pid` 参数代表识别何种语音, 如 1537 代表普通话识别、1737 代表英语识别、1637 代表粤语识别等。此外, 为了防止未能成功识别的情况发生, 我们使用 `try...except...` 来避免程序报错。

### (3) 步骤 3: 与图灵机器人对话 (王菀婷)

```
# 3、与机器人对话: 调用的是图灵机器人
# 图灵机器人的API_KEY、API_URL
turing_api_key = "your turing_api_key"
api_url = "http://openapi.tuling123.com/openapi/api/v2"
headers = {'Content-Type': 'application/json;charset=UTF-8'}
```

与调用百度 API 接口类似, 这里我们通过调用图灵机器人的 API 接口完成自动应答。

```
# 图灵机器人回复
def Turing(text_words=""):
    req = {
        "reqType": 0,          # 0表示输入类型为文本
        "perception": {
            "inputText": {
                "text": text_words # 输入文本信息
            },
            "selfInfo": {        # 客户端属性
                "location": {
                    "city": "上海",
                    "province": "上海",
                    "street": "闵行区"
                }
            }
        },
        "userInfo": {
            "apiKey": "274956079ca343ac91d77932c83d658b", #图灵机器人的API_KEY
            "userId": "MINO" #用户唯一标识, 可随意填写 (不能包含中文)
        }
    }

    #给json串赋值
    req["perception"]["inputText"]["text"] = text_words

    #向接口网站发送请求
    response = requests.request("post", api_url, json=req, headers=headers)
    response_dict = json.loads(response.text)

    #得到接口的回复进行解析
    result = response_dict["results"][0]["values"]["text"]
    print("Happy Robot said: " + result)
    return result #返回回复内容
```

通过定义 Turing(), 我们得到了图灵机器人自动回复的文本内容。为了使语音自动对话机器人能够更好地起到安慰人的作用, 我们还在图灵机器人官方网站上为我们创建的机器人添加了部分私有语料库, 以便有针对性地为用户提出建议, 具体内容如下:

Q1我觉得我哪里都不如别人。/为什么大佬这么强? /为什么他们这么厉害?

A1每个人都有自己的闪光点,能和他们站在一起,你一定也有什么是别人比不上的。

Q2为什么高数怎么学都学不会? /关键词: 数学

A2即便不一定考最高分,每一点付出也会有一点回报,你一定会越来越好的。

Q3 TA不喜欢我/TA有了喜欢的人/表白失败了/我失恋了。

A3 很想给你一个拥抱,相信你会遇到可以真的给你一个拥抱的人。

Q4为什么付出很多精力,却没有回报? /为什么没有回报?

A4回报可能正在来的路上,只是迟到,不要心急。

Q5我缺乏自控力/无法自律/学不下去 学不进去。/我太拖延了。/我今天一事无成。

A5 那么就现在!放下手机,和我一起开始学习吧~什么时候开始都不晚。

Q6我和(朋友/舍友/同学)的关系很僵/我处理不好人际关系。

A6只要诚心待人,也会被人诚心对待。对于能否成为密友,还是洒脱点好。

Q7我的体重减不下去。

A7控制饮食,规律作息,保持最健康的体型就好,健康的就是最美的!

Q8我很想念家人。/我好想家。

A8家人们一定也很想你,互相思念的人再相见时一定会更加幸福。就在分别的时刻里更加努力地自己的工作吧~

Q9我处理不好和亲人的关系。

A9只要有亲情和爱,没有什么是不好起来的。希望你们能互相理解。

Q10我觉得自己长相一般。/我长得不漂亮。/这不是一个看脸的社会? /我长得是不是很难看?

A10美有许多种,不能断言谁是美丽的,谁是不美丽的。每个人都有自己美丽的一面,坦然接受自己,自信才会让人从内而外散发光彩。

Q11人生的意义是什么/人为什么要活着?

A11人生的意义就是去寻觅自己所真正热爱的,并为之燃烧自己的生命。你也会遇到自己真正热爱的事情,在那之前,耐心等待,不要放弃。

Q12梦想太遥远,无法实现。/现实和我想象中总是不一样。

A12奔向梦想的旅程其实就是梦想的意义本身。改变自己所能改变的,接受不能改变的,就算没有成功,说不定也能获得99.9%。

Q13要期末了,作业太多,压力太大。(期末)/不知道能不能活过期末,太累了。

A13做好时间规划,不逃避问题,适当放松心情,考完就可以好好休息啦。

Q14我最近总是失眠。/晚上睡不着觉。

A14睡前喝一杯热牛奶,或者听一听轻音乐。长期睡眠困难的话可以去看看医生,吃一些褪黑素。夜晚想太多,不如好好睡一觉~

Q15我不想活了。/人生真的好失败。/我对人生好绝望。

A15人生远比你想象的要精彩,还有太多的内容你没有体验。任何的坏事都会有转机。

Q16没有人在意我。/我对任何人来说都不重要/无关紧要)/没人在乎我的感受。

A16你是家人最宝贵的家人,你也是朋友最宝贵的朋友。

Q17这个城市/这所学校/所有人都好冷漠。

A17每个人都有着各自关心的事情,有时候可能就无暇停下来关心他人。互相理解,尽力去发现陌生人的温暖吧。

Q18我没有朋友。/我觉得好孤单。

A18一个人的时候就尽力提升自己,谁说没有朋友不是一件好事呢,也挺酷的嘛。

Q19我觉得自己的前途很渺茫。/我对未来很迷茫。/我在这个社会里找不到自己的位置。

A19前途、未来都是在不断的探索中才能逐渐明朗的。给自己多一些接触社会的机会,多了解、多发现,找到自己的兴趣所在,并向着这个方向去提升自己的能力。未来有一天会自己找到你。

Q20我会不会找不到工作/我会不会失业。

A20拥有专业能力,或者只是拥有一项基本的技能,都可以找到安身之处,关键是减少不必要的焦虑,努力提升自己。

Q21我好累/我好疲惫/撑不下去了。

A21完成那项最紧急的工作,抽一个下午或早晨给自己彻底放个假吧。

Q22我怕爱我的人失望。/我怕符合不了别人的期待。

A22值得你为之努力的人都是最希望你开心的人。在健康快乐的基础上付出自己所有的努力就可以问心无愧了。

Q23我的生活费不够了。/我为什么这么穷?

A23经济真的有问题的话可以申请助学金哦。个人的经济条件要靠自己的努力去创造,坦然面对经济差距,并为之努力吧。

Q24这场考试考砸了。/我考试考得好烂。

A24我相信你心里知道自己的问题在哪里，勇敢面对，并且努力去改善。考试只是一种检测，下次会更好。

Q25是不是死了就能解脱？/是不是死亡能解除痛苦？

A25一个人的死亡是以很多人的痛苦为代价的逃避。不要做懦夫，要做勇敢的人。

Q26为什么没有人理解我？/他们说我的痛苦都是小题大做。

A26确实只有切肤之痛才能体会得最深。可以试着向身边的人敞开心扉，或者和亲人、和老师聊一聊。  
总会有人理解你的不容易，在那之前一个人也要坚强。

Q27我会不会得了抑郁症？/我害怕得抑郁症。

A27讳疾忌医不可行哦。如果真的出现了疑似症状一定要告诉亲友，同时积极去医院就诊。

Q28人生太艰难了。/努力实在太累了。

A28先苦后甜还是先甜后苦？这应当是一个值得好好思索的问题。任何有回报的事情都不会轻轻松松简简单单，选择了努力，我们就要勇敢。

Q29我控制不住自己熬夜。/熬夜以后会后悔。

A29 到了该睡觉的时候就什么也别想，果断地放下手机吧。坚持几天后你会喜欢上早睡早起的感觉的。

Q30可是你也只是个机器人。

A30我也是另一个人设计了很久，专门来哄你开心的呀~

Q31今天运气好差。/今天好倒霉啊。/每件小事都不顺心。

A31今天的运气攒下来，会留给明天的，不顺心的小事和朋友说说笑笑，就过去啦~

Q32我的痛苦什么时候才能结束啊？/这样的日子什么时候才能到头啊？

A32终点不远，但我们要学会在一切日子里找到属于自己的快乐。

#### （4）步骤 4：文字转语音（马筱琛）

```
# 4、实现文字转语音
# 初始化语音
engine = pyttsx3.init()
# 设置语速
rate = engine.getProperty('rate')
engine.setProperty('rate', rate-50)

# 语音合成，输出机器人的回答
while True:
    my_record()
    request = listen()
    if request == "再见。":break
    #当用户说“再见”时，程序自动退出
    response = Turing(request)
    engine.say(response)
    engine.runAndWait()
```

我们通过类库 `pyttsx3` 实现了文字转语音工作，最后运用一个 `while` 循环将整个交互式语音识别的流程串起来，并为语音自动回复机器人的终止进行了设置：当用户说出“再见”（也可以设置为其他关键词）时，程序自动终止。

```

import win32com.client

# 初始化语音
speaker = win32com.client.Dispatch("SAPI.SpVoice")

# 语音合成，输出机器人的回答
while True:
    my_record()
    request = listen()
    if request == "再见。":break
    response = Turing(request)
    speaker.Speak(response)

```

我们还可以使用更为简便的 `win32com.client` 来完成文字转语音工作。由于这个类库为 Windows 特有的，所以从普遍适用性上考虑，我们仍然决定使用 `ppttsx3`。

## 2.3 最终实现（马筱琛）

```

def main():
    start()
    while True:
        my_record()
        request = listen()
        if request == "再见。":break
        response = Turing(request)
        engine.say(response)
        engine.runAndWait()

main()

```

最后的 `main()` 函数中，`start()` 用来调用面部识别相关内容，当识别到悲伤的表情后，会自动退出，开始进入接下来的语音交互过程，最后听到用户说出“再见”时，整个程序会终止。

### 3 局限与改进（王菀婷）

#### 3.1 局限性

1. 在程序中我们通过图灵机器人实现自动对话，对话回复内容源自我们自己编写的语料库，可以回答心情沮丧时比较常见的问题。但是语料库内容有限，并且为了成功实现每次对话，问题匹配度设置在了 70%。这样的情况下造成答非所问的几率是比较大的，一些脱离语料库问题的回答很有可能不能让使用者感到满意。
2. 我们所做程序的设想是识别表情后自动开始语音对话，在语音对话的实现方式上我们采用了 `pyttsx3` 或 `win32com.client` 将文字转化为语音并读出。受到该方式的局限，语音回复的声音为人工合成音，听起来比较生硬，语音对话的代入感不强。
3. 全部的交流过程通过文字和语音来实现，内容和形式过于单一，带给使用者的体验感过于单薄。
4. 通过 `python` 第三方包和各种类库调用电脑摄像头、进行文字和语音转换等操作只能在配置了 `python` 环境并安装了各种配件的电脑上运行，下载操作比较繁琐。此外程序局限在电脑端，不能随身携带，时空上有很大局限性。

#### 3.2 优化可能性及展望

1. 针对答非所问的情况，可以再安装相应的信息采集和返回系统，在使用过程中收集各类问题，不断更新和扩充语料库，并提升匹配度要求，尽量做到通顺对话，语义明确。
2. 针对合成语音生硬的问题，我们设想能够寻找并调用优化过的人声系统（例如，某些听书软件中经过优化处理的合成音能够接近于真人发声效果），或者自己进行声音的采集和录用：考虑到建立整个人声系统比较艰难，我们可以一条一条录制回答，甚至也可以分出不同角色类型的声音供选择。
3. 在对话过程中，我们可以增设聊天界面，在语音回复之外恰当使用好玩的表情包或者放送一些令人心情愉悦的视频、推荐节奏舒缓轻松的歌曲等，丰富聊天内容，加强沟通效果，更好地改善使用者的精神状态，增强沟通欲。
4. 我们希望在一系列的优化和改进后，程序可以导入移动端，通过微信公众号或者小程序运行，或者添加在音乐 app 的附属功能中。程序能够更加便捷灵活，功能也更强大。

## 4 小结（马筱琛）

在程序设计方面，这次的课程作业对于我们小组的三个成员来说都是一次新的尝试。在总体设计程序、查找类库、debug 等过程中，我们都遇到了很多问题。于是，请教学长和老师、查找资料、不断尝试和突破……我们在一次次的失败与成功中反复横跳，强迫自己面对问题、解决问题，探索所有可能成功的方法，付出了不少的精力与时间，才攻克下我们眼中的“难题”。但是，也正是如此，我们才获得了运行成功后强烈的喜悦和满足，才进一步地想要探索程序设计的更深奥的世界。

与此同时，人文精神也流淌在看似冷冰冰的编程过程中。队友们在磨合中更加了解彼此的想法，在合作中不断优化和改进程序，正如优化我们的团队意识和合作精神。付出越多努力，我们反而越想实现自己创作项目的初衷：希望带给闷闷不乐的人以欢乐，激发他们的沟通欲和倾诉欲。这样看来，也许编程正是一件非常浪漫的事：看似没有生命的代码，却是出自满怀热情之人的双手，是为了人们的快乐和便捷才诞生在一个个终端。

希望我们在不久的将来能不断优化和改进我们的程序，让它变得更有应用性、更有意义；希望我们也能成为深谙编程趣味的人，能够用程序便利工作和学习，也为其他的人带来便捷和快乐。

具体分工如下：

李嘉翌同学负责面部识别代码的编写与注释工作，和相应的前期准备（具体见报告标注，包括但不限于所用类库的介绍），还有 demo 的拍摄

马筱琛同学负责语音交互机器人中语音转音频文件、音频文件转文字、文字转语音代码的编写，和相应的前期准备（具体见报告标注，包括但不限于所用类库的介绍）；还负责了报告中的“小结”和整个报告的图片、排版、整合工作。

王菀婷同学负责语音交互机器人中自动对话代码的编写，并设置了专有的语料库，还有相应的前期准备（具体见报告标注，包括但不限于所用类库的介绍）；同时还负责了报告中的“选题原因”“局限性与改进”，和 demo 的剪辑。