Software Engineering and Applications 软件工程与应用, 2020, 9(6), 475-486
Published Online December 2020 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2020.96055

Hans汉斯

基于语音识别和GPS定位的智能垃圾分类APP

黄　珂，杨文杰，刘　佳*

武汉商学院，信息工程学院，湖北　武汉
Email: 1402671892@qq.com, 1131168003@qq.com, *mikeliujia@wbu.edu.cn

收稿日期：2020年11月16日；录用日期：2020年12月17日；发布日期：2020年12月24日

摘　要

在垃圾分类逐渐立法、执法、司法的今天，如何引导民众正确地进行垃圾分类已成为社会要点问题之一。针对这一问题本文提出了一款智能垃圾分类APP的设计，采用经典三层结构设计，使用物联网技术及百度定位服务使得移动端与垃圾站点互联，实现站点信息的“可视化”同时支持路线导航功能；综合科大讯飞语音，可实现在线语音识别并分类。操作简单方便，实用性强，准确性高，适用范围广，符合国家创新、协调、绿色、开放、共享的发展理念。

关键词

物联网，语音识别，GPS定位，垃圾分类

Intelligent Garbage Classification APP Based on Speech Recognition and GPS Positioning

Ke Huang, Wenjie Yang, Jia Liu*

School of Information Engineering, Wuhan Business University, WBU, Wuhan Hubei
Email: 1402671892@qq.com, 1131168003@qq.com, *mikeliujia@wbu.edu.cn

Received: Nov. 16th, 2020; accepted: Dec. 17th, 2020; published: Dec. 24th, 2020

Abstract

With the gradual legislation, law enforcement and judicature of garbage classification, how to

*通讯作者。

文章引用: 黄珂, 杨文杰, 刘佳. 基于语音识别和 GPS 定位的智能垃圾分类 APP [J]. 软件工程与应用, 2020, 9(6): 475-486. DOI: 10.12677/sea.2020.96055

guide the public to conduct garbage classification correctly has become one of the key issues in the society. To solve this problem, this paper proposes the design of an intelligent garbage classification APP. It adopts the classic three-layer structure design and uses the Internet of Things technology and Baidu positioning service to connect the mobile terminal with garbage sites, so as to realize the "visualization" of site information and support the route navigation function. Integrated IFLYtek voice can realize online speech recognition and classification. This APP is simple and convenient operation, strong practicality, high accuracy, wide range of application, in line with the national innovation, coordination, green, open, sharing development concept.

Keywords

Internet of Things, Voice Recognition, GPS Positioning, Garbage Classification

1. 研究背景及意义

物联网浪潮的冲击，使得万物互联成为了可能，衍生出了不计其数的智能产品。在 2020 年的今天，“垃圾分类”已经成为了一个国民耳熟能详的一个词语。尽管如此，“垃圾”这个词汇仍然是压在我国头上的一块巨石。每年由于垃圾处理不当所引发的惨剧比比皆是。在这样的环境背景下，“互联网+垃圾分类”逐渐切入垃圾分类领域，很多市场参与者都在思考、尝试将互联网技术与垃圾分类有效结合[1]。尤其在上海市十五届人大二次会议表决通过《上海市生活垃圾管理条例》将“垃圾分类”纳入法治框架对违规者进行罚款处罚之后，“垃圾分类”再也不是一句挂在嘴边的空话，而是实实在在的行动。但由于大多数人尤其是老年人对“垃圾分类”知识所知甚少，如何正确地处理垃圾也是一个头疼的问题。因此，借助互联网技术、大数据技术、物联网技术等智能化手段进行城市垃圾分类[2]，开发一款适合所有年龄段，具有便携性且操作简单的智能垃圾分类系统恰合时宜。

本系统的主要功能在于，当用户需要扔垃圾时，只需要打开本 APP 的语音识别功能，对着手机说出手中垃圾的名称即可完成分类，操作简单方便，适合所有年龄段尤其是中老年人群。同时，本产品使用物联网技术，与硬件互联，用户可在移动端查看附近垃圾桶的位置及垃圾桶当前时刻信息。例如：人们在户外需要丢垃圾时，本系统可以帮助用户迅速定位到附近的垃圾桶，查看垃圾桶剩余容量、是否处于维修等信息且支持路线导航功能。

在推进城市生活垃圾分类处理工作中，呼吁广大市民对垃圾分类管理工作的认知，提高市民良好的生活垃圾分类习惯尤为重要。通过使用本系统，不仅能提高年轻人垃圾分类的速度、培养小朋友的环保意识、养成垃圾分类习惯，同时还能减轻中老年人因不会垃圾分类而产生厌烦情绪，非常符合当前生态文明建设。

2. 系统研究分析

智能垃圾分类技术的研究整体呈增长趋势，这说明智能垃圾分类技术早已受到关注，随着技术手段、方法及理念的更新，对该领域的探究热度不断上升[3]。从目前国内外的研究成果来看，智能垃圾分类系统主要分为以下两种：

2.1. 纯硬件的智能垃圾桶

这种智能分类垃圾桶大多能实现桶盖自动开合、桶满报警、智能语音分类等功能，但因语音识别感知距离的局限性，使得其应用场景非常有限，很难在人多嘈杂的商场及街道拥有较好的表现。即使在安静的室内环境，让人靠近垃圾桶讲话是令人难以接受的。

2.2. 纯软件的传统智能垃圾分类 APP

这类 APP 大都把全部精力放在如何提升分类的准确度，毫无疑问，作为一款垃圾分类 APP，它很优秀。但相比于本文所介绍的智能分类系统，功能偏少，能为用户带来的便捷较少。

2.3. 本系统优势

本系统在 Android 环境下集成了语音识别技术和 GPS 定位技术，不仅实现了传统垃圾分类 APP 的智能分类识别功能，同时将语音识别从硬件层转移到了软件层，巧妙避开了语音识别对距离的局限性，能使系统能适应更多不同环境，为用户提供更好的体验。更重要的是，本系统使用物联网技术，使得系统与硬件端互联，可实现垃圾站点的信息可视化，用户可直接在手机查看垃圾站点当前时刻信息，而无须像纯硬件系统一样必须走进垃圾桶才能查看。同时，本系统使用了 GPS 技术，用户可直接在 APP 端定位到附近垃圾站点位置并导航，防止在繁华的城市中找不到垃圾桶的尴尬情况发生。

3. 系统设计

3.1. 总体架构

基于语音识别和 GPS 定位的智能垃圾分类 APP 采用了经典的三层结构设计，分为用户界面层、业务逻辑层和数据访问层。用户界面层的主要功能是展示和接收用户的操作数据，例如将地图及垃圾桶位置展示在手机端上；业务逻辑层主要负责对用户的操作逻辑进行计算，例如搜索用户需要得知的垃圾种类；数据访问层主要是负责数据的读取和传递，例如将保存在 MySQL 的数据读取出来。整个结构由业务逻辑层作为中间者，将数据访问层读取的数据进行逻辑计算并传送到用户界面层展示。

3.2. 设计环境

开发软件：Android Studio 3.0.1 + Navicat 10.0.11.0
开发语言：Java + XML + SQL
运行环境：Android 6.0 及以上

3.3. 系统功能设计

系统功能主要分为四个模块：定位功能模块、路线规划功能模块、垃圾桶信息查询功能模块和语音识别功能模块。这四个功能模块被嵌入在三个 APP 界面中，主界面、导航界面及语音识别界面。

系统主界面如图 1 所示，为了使得软件看起来更加精简美观，操作更加简单方便，本软件的主界面直接以地图的形式显示。在主界面中，存在着许多垃圾桶站点图标，用户可以非常直观的看到自己以及周围垃圾桶站点的位置。同时，这些站点在地图上都是可以点击的，当垃圾桶站点被用户选中后，会弹出当前垃圾桶站点的对应属性框，属性框中包含了多条垃圾桶站点的相关状态，例如：垃圾桶当前时刻所剩容量、是否正处于维修状态等信息，如图 2 所示。当用户在查看完垃圾桶信息后决定前往该站点时，可以点击属性框中的“到这去”按钮，此时系统会自动为用户导航路线，如图 3 所示。

语音识别界面如图 4 所示，当用户需要对垃圾进行分类时，点击“说话”按钮弹出倾听对话框，说出手头垃圾的名字，系统会在数据库中自动匹配该垃圾相应的类别并将其显示。

Figure 1. System main interface
图 1. 系统主界面

Figure 2. Attribute of station
图 2. 站点属性框

Figure 3. Route navigation interface
图 3. 路线导航界面

Figure 4. Speech recognition interface
图 4. 语音识别界面

系统流程图如图 5 所示。

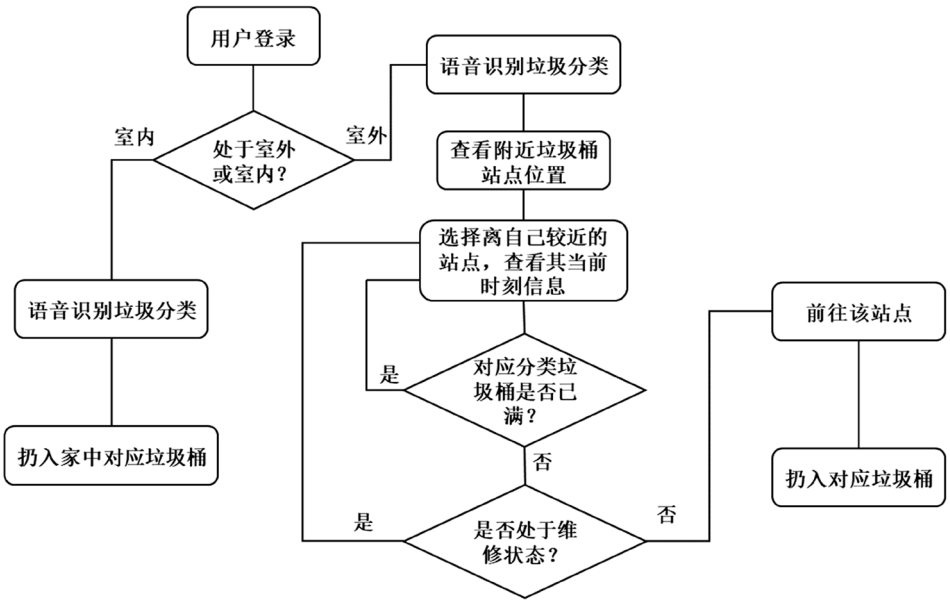

Figure 5. System flowchart
图 5. 系统流程图

4. 系统实现

4.1. 系统模块分析

4.1.1. 定位及导航模块分析

随着智能手机的普及，如今大多数手机都内置了 GPS 及方向传感器，衍生出了众多的定位服务。百度地图就是其中一个，其具有高成功率、低功耗、响应快的特性。测试结果显示，在平均事务响应时间方面，百度地图 API 极具优势，其平均事务响应时间不超过 20 ms [4]，这种高响应的定位服务正是本系统所需要的。同时它还支持智能硬件定位，可通过蓝牙、WiFi、基站、GPS 实现精准定位，能满足绝大

多数智能硬件的定位需求，非常契合本系统。因此，本系统通过百度地图开放平台提供的服务接口和云服务器，实现定位及导航功能[5]。

4.1.2. 语音识别模块分析

考虑到语音的多样性和识别语音的效率，在本系统中采用了基于科大讯飞提供的 SDK，利用其 SDK 的高识别准确性和个性化的语音识别设计出了本系统的语音识别模块。中国的智能语音产业的发展的一个亮点，就是在自然语音的处理上有着很高的技术水准，如科大讯飞其对于语音的识别的准确率可以达到 96%以上[6]。在本系统中，将语音识别模块与 MySQL 数据库结合使用，能够在线检索垃圾所属的分类。当用户说出新的垃圾名词时，本系统会优先匹配与该名词相接近的名词以供用户选择，同时将该陌生名词上传到云端数据库，由开发者根据垃圾分类相关法律条规手动将其添加到数据库中并更新。

4.1.3. 数据库模块分析

物联网正在慢慢改变我们与周围世界互动和感知的方式，随着越来越多的智能设备开始生成数据，我们需要利用数据库系统来解决收集、存储和理解数据的问题。作为关系型数据库，MySQL 具有体积小、速度快、成本低、服务稳定且支持多种操作系统和开发语言等特点，能满足本系统的所有需求。

4.2. 定位及导航模块实现

4.2.1. 定位及导航模块实现流程图

定位模块流程图如图 6 所示，路线导航模块流程图如图 7 所示。

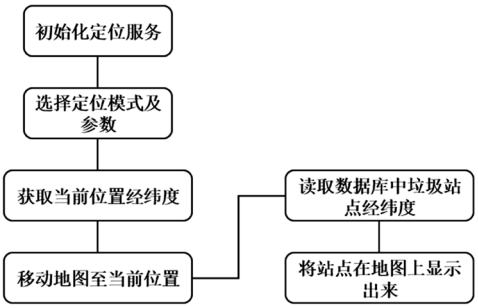

Figure 6. Location flowchart
图 6. 定位模块流程图

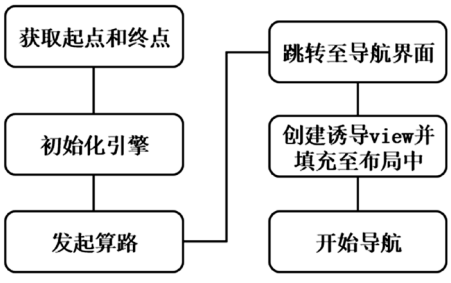

Figure 7. Route navigation flowchart
图 7. 路线导航模块流程图

4.2.2. 垃圾站点定位实现

1) 在实现定位功能之前，我们需要检查 APP 的权限，如果没有权限，向用户弹出是否允许该权限的对话框，若用户拒绝，则退出程序。权限相关的关键代码如下：

```
List<String> permissionList = new ArrayList<>();
if(ContextCompat.checkSelfPermission(MainActivity.this, Manifest.
        permission.ACCESS_FINE_LOCATION)!= PackageManager.PERMISSION_GRANTED){
    permissionList.add(Manifest.permission.ACCESS_FINE_LOCATION);
}
if(!permissionList.isEmpty()){
    String[] permissions = permissionList.toArray(new String[permissionList.size()]);
    ActivityCompat.requestPermissions(MainActivity.this, permissions,1);
}else{
    requestLocation();
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
  switch (requestCode){
    case 1:
      if(grantResults.length>0){
        for(int result : grantResults){
          if(result != PackageManager.PERMISSION_GRANTED){
            Toast.makeText(this, "必须同意所有权限才能使用本程序", Toast.LENGTH_SHORT).show();
            finish();
            return;
          }else{
            Toast.makeText(this, "发生未知错误",Toast.LENGTH_SHORT).show();
            finish();}
          break; }}
    default:}}
```

代码说明：

上图代码的第一段首先定义了一个 List 容器用来存放本 APP 所需要但未被授权的权限，checkSelfPermission() 是一个类方法，用于检查权限是否被授予，若权限已被授权则返回 PackageManager.PERMISSION_GRANTED。这里对本 APP 所需的所有权限进行检查(当然实际远不止一条，为了方便省略了很多)，将那些未被授予的加入 List 容器，然后调用 ActivityCompat.requestPermissions() 方法请求权限，请求 id 为 1，当权限请求结束后系统会回调 onRequestPermissionsResult()方法。在第二段代码中我们重写了 onRequestPermissionsResult()这个方法，并在里面加入逻辑，判断当请求 id 为 1 时，对收到的 List 权限容器遍历，检查每一个权限是否被授予，若用户没有将全部权限授权，提示用户必须全部授权并退出程序。

2) 垃圾站点在地图中的显示，代码如下：

```
//定义Maker坐标点
point = new LatLng(30.468412, 114.094199);
//构建Marker图标
BitmapDescriptor bitmap = BitmapDescriptorFactory
        .fromResource(R.drawable.bin32);
//构建MarkerOption，用于在地图上添加Marker
OverlayOptions option = new MarkerOptions()
        .position(point)
        .icon(bitmap);
```

代码说明：

LatLng 是一个地理坐标基本数据结构的类，其构造方法中的两个参数分别对应着纬度和经度；BitmapDescriptor 用于对坐标点设置图案，OverlayOptions 是地图覆盖物选型基类，用于创建地图覆盖物并选择已设置好的覆盖物的位置和图案。

4.2.3. 路线导航实现

路线导航分为两个步骤，初始化引擎和发起算路，其关键代码如下：

```
public void startCalcRoute(){
   //获取起点
   LatLng startPt = new LatLng(location.getLatitude(),location.getLongitude());
   //获取终点
   LatLng endPt = point;
   final WalkNaviLaunchParam param = new WalkNaviLaunchParam().stPt(startPt).endPt(endPt);
   //使用步行导航前，需要初始化引擎
   WalkNavigateHelper.getInstance().initNaviEngine(this, new IWEngineInitListener() {
      @Override
      public void engineInitSuccess() {
         Log.d("------","引擎初始化成功");
         routePlaneWithParam(param);
      }
      @Override
      public void engineInitFail() {
         Log.d("------","初始化引擎失败");}
   });
}
```

代码说明：

初始化阶段需要获取“我”当前的位置经纬度 startPt 以及垃圾站点的位置经纬度 endPt，并将其封装为 WalkNaviLaunchParam 对象，WalkNaviLaunchParam 其实就是一个参数类，其存在的意义就是为了传给 routePlaneWithParam()方法。routePlaneWithParam()方法是百度对外提供的一个路线规划方法，将含有起点和终点的参数 param 传给它即可实现自动算路。

4.3. 语音识别模块实现

4.3.1. 语音识别模块流程

以下是语音识别模块实现流程：

1) 首先要开启 Android 手机的录音功能，否则无法进行语音识别。

2) 点击“我要分”按钮，开始调用科大讯飞的语音识别接口，初始化 RecognizeDialog，接着将语音识别引擎设置为 sms (文字)，地域设置为 zh_cn (中国)，语言设置为 mandarin (普通话)设置识别侦听，等待语音输入结束。

3) 语音识别结束后将识别到的 json 数据解析成文字并在垃圾分类库中搜索。

流程图如图 8 所示：

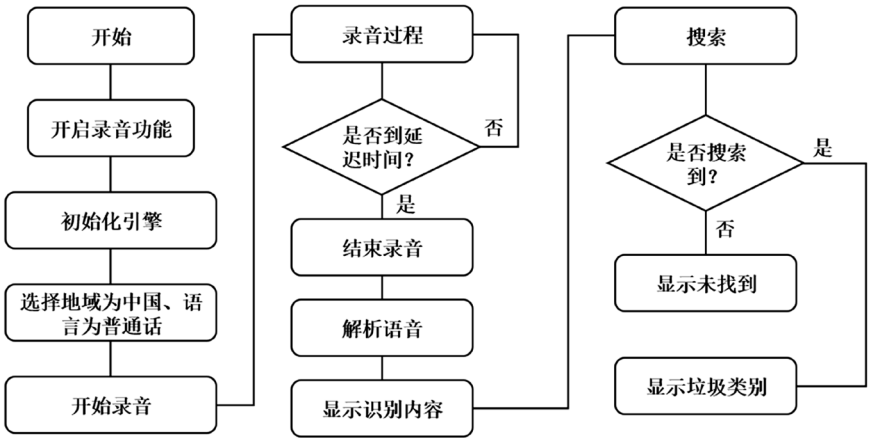

Figure 8. Speech recognition flowchart
图 8. 语音识别模块流程图

4.3.2. 语音识别实现

1) 在实现语音识别之前，首先要检测 APP 的权限授予情况，方法与 4.2.2 一致，这里不再赘述。

2) 对语音识别的初始化代码如下：

```
//1.创建RecognizerDialog对象
RecognizerDialog mDialog = new RecognizerDialog(context, null);
//2.设置accent、language等参数
mDialog.setParameter(SpeechConstant.LANGUAGE, "zh_cn");
mDialog.setParameter(SpeechConstant.ACCENT, "mandarin");
```

代码说明：

RecognizerDialog 是讯飞对外提供的一个语音识别控件，可以理解为语音的接收者，它是一个继承于 Dialog 类的子类，会以对话框的形式显示出来，这一部分为其设置了地域、语言参数。

3) 识别控件初始化好后，设置回调监听，开始录音处理语音数据，代码如下：

```
//3.设置回调接口
mDialog.setListener(new RecognizerDialogListener() {
    @Override
    public void onResult(RecognizerResult recognizerResult, boolean isLast) {
        if (!isLast) {
            //解析语音
            //返回的result为识别后的汉字,直接赋值到TextView上即可
            String result = parseVoice(recognizerResult.getResultString());
            resultText.setText(result);
            for(Rubbish r : rList){
                Log.d("sortActivity", r.getName());
                if(r.getName().equals(result)){
                    Log.d("equals", r.getVariety());
                    sortText.setText(r.getVariety());
                    break;}
                    sortText.setText("识别失败，请重试");}}}
    @Override
    public void onError(SpeechError speechError) {
    }
});
//4.显示dialog，接收语音输入
mDialog.show();
```

代码说明：

当用户点击开始录音时，RecognizerDialog 会设置回调监听，并调用了 show()方法将接收语音的对话框显示出来，回调接口中重写了 onResult()方法，该方法传入了两个参数，其中 RecognizerResult recognizerResult 代表的是语音识别的结果，但这个结果是 Json 类型的，所以我们需要将其解析为我们需要的汉字类型；boolean isLast 表示录音是否还在持续，当麦克风在一段时间内没有感知到声音时会返回 false。因此这段代码的逻辑是，当录音开始时，讯飞引擎会帮助我们将声音进行分帧、拼接等操作，待录音结束，调用 parseVoice()方法将传入的 json 类型结果解析为汉字类型以供使用。

4) 其中 parseVoice()方法需要开发者自己编写，解析 Json 的方法有很多，这里选用谷歌推出的 Gson 库实现 Json 对象的反序列化，部分代码如下：

```
public String parseVoice(String resultString) {
    Gson gson = new Gson();
    Voice voiceBean = gson.fromJson(resultString, Voice.class);
    StringBuffer sb = new StringBuffer();
    ArrayList<Voice.WSBean> ws = voiceBean.ws;
    for (Voice.WSBean wsBean : ws) {
        String word = wsBean.cw.get(0).w;
        sb.append(word);
    }
    return sb.toString();
}
```

4.3.3. 语音识别模块测试

我们在垃圾分类库中随机选取了十种垃圾进行语音测试，目的是为了测试语音识别的准确性，测试结果如表 1 所示，语音识别结果显示如图 9 和图 10 所示：

Table 1. Speech test results
表 1. 语音测试结果

理想值	饼干	菜叶	西瓜皮	卫生纸	筷子	电池	酒精	报纸	罐头	塑料瓶
实际值	饼干	彩页	西瓜皮	卫生纸	筷子	电池	酒精	豹子	罐头	塑料瓶
检测结果	湿垃圾	失败	湿垃圾	干垃圾	干垃圾	有害垃圾	有害垃圾	失败	可回收	可回收

Figure 9. Recognition successful
图 9. 识别成功

Figure 10. Recognition failed
图 10. 识别失败

由测试结果可见：在大多数情况下语音识别准确性较高，但有时会受到口音、同音字、平翘舌等因素的影响。

4.4. 后端数据库连接

本系统的 mysql 数据库是储存在阿里云服务器中，为数据库提供了 web 服务环境。采用的连接方式为 JDBC 连接，连接的关键代码如下：

```
private void connectMySQL() {
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                Log.d("------------------","正在连接.....");
                Class.forName(JDBC_DRIVER);
                conn = (Connection) DriverManager.getConnection(DB_URL,USER,PASS);
                statement = conn.createStatement();
                Log.d("------------------","连接成功");
            } catch (SQLException e) {
                e.printStackTrace();
                Log.d("------------------","连接失败");
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
                Log.d("------------------","驱动出错");
            }
    thread.start();
}
```

代码说明：

上述代码我们可以看到，这段连接 mysql 的代码是放在子线程 thread 中的，这是因为 Android 系统的保护机制不允许把耗时操作放入主线程执行，而数据库的连接属于耗时操作，所以必须创建一个子线程执行数据库的连接。

5. 结论

本系统初步实现了一个以 Android 开源环境搭建的智能垃圾分类系统，利用百度定位服务实现位置信息的获取，MySQL 数据库实现传感器的数据采集、存储及读取，讯飞语音转写功能将语言音频转换为文本输出。具有较好的简便性和实用性，在这些功能中，识别垃圾种类是难度最高、最待公关的一道难题。这也是本系统目前正遇到的问题：如何提高语音识别的准确性及如何对垃圾准确无误的分类。在后续工作中将致力于通过软件算法进一步提升语音识别的准确性，主要是减少同音词条对检索结果的影响。

基金项目

本篇论文获得项目编号为 202011654040 武汉商学院 2020 年度大学生创新创业训练项目《“慧”分魔桶》的研究资助。

参考文献

[1] 王鸿飞, 何静, 龙云凤. “互联网+垃圾分类”企业发展现状及对策建议[J]. 广东科技, 2020, 29(9): 51-56.

[2] 张兆莹. 基于循环经济推行城市生活垃圾智能化分类的探究[J]. 资源再生, 2020(8): 18-20.

[3] 贾宁. 中国智能垃圾分类技术专利申请现状[J]. 中国自动识别技术, 2020(4): 53-57.

[4] 向玉云, 高爽, 陈云红, 黄嘉成, 许新华. 百度、高德及 Google 地图 API 比较研究[J]. 软件导刊, 2017, 16(9): 19-21+25.

[5] 宋红凯, 杜洪波, 程宇航, 张先卓. 基于“百度地图”的掌上校车 APP 设计[J]. 软件, 2019, 40(5): 21-25.

[6] 任智. 浅析中国智能语音产业的发展潜力[J]. 广西质量监督导报, 2019(8): 93-94.