

小度

1. 方案描述

应用百度云语音识别代替手动输入,通过图像识别,检测当用户发送图片时弹出个性化推荐。

此应用为 IM 场景,当前为单向聊天,所以用智能聊天代替,常规场景中,此处为好友聊天,可通过@系统,来主动接入智能聊天。

2. APP 基本介绍

本 APP 取自 FamilyApp 中的单一聊天场景,对当前主流的 IM 聊天场景进行补充,增加发送图片检测,并根据图片内容(接入图像识别),在聊天场景下进行推荐;另一方面则是引用语音识别,代替用户手写输入,再将文字发送出去,随着语音识别的进一步升级,准确度已达到日常可用。

3. FAMILYAPP

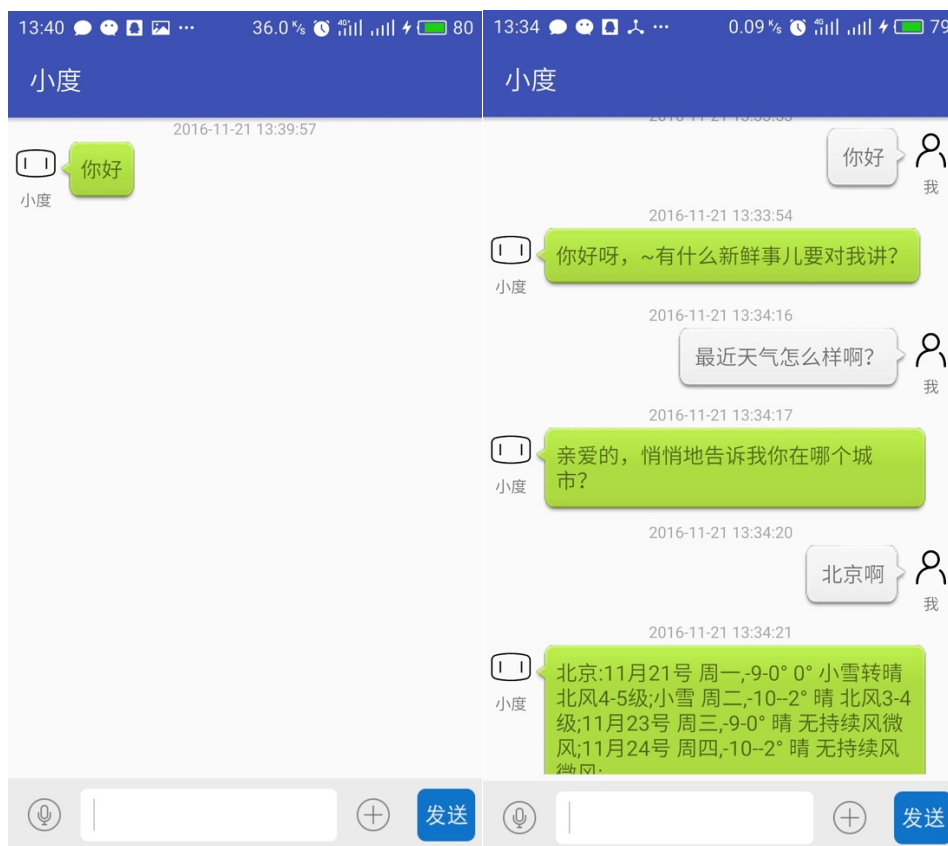
Family 取自家族一意,是一个小范围 IM 场景,适用于家族内部或核心交友群内部的一个 App,主打“简”和“便”。简是场景足够简单,当在家族时,场景为一“族”(类似常规中群),引入传统“辈分”概念,对“七大姑八大姨”进行标示;对家族重大事件进行通知,增加家族内部信息流通(解决:回老家过年时才知道,某一长辈于秋季已过世);对家族健康进行监控,推荐家族成员每年健康体检,共享家族内部健康状态,等场景。将负责场景进行拆分,使其足够简单。便,当这些功能,收集到足够信息时,就能对场景进行进一步优化,和智能化推荐,方便家族内各种场景。

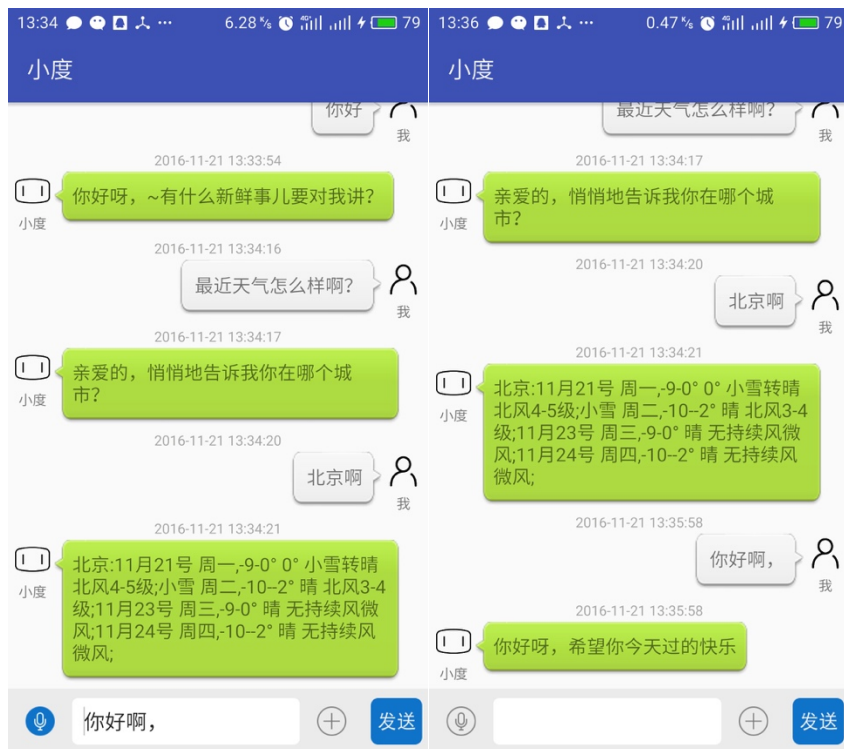
4. 总体功能架构

总结构分为客户端和服务端

- 1、语音识别则是由百度语音识别 SDK 在客户端实现，调用 SDK 实现语音识别。
- 2、图像识别，则是同 http 接口请求服务器，并提及图像数据，由服务端转发给百度 OCR 接口，实现图像识别。

5. 主界面





6. 引入百度云业务

- 1、语音识别 - 语音识别代替手动输入
- 2、图像识别 - 百度 OCR 图像识别

(代码截图)

1、语音识别回调

```
private VoiceRecognitionClient.VoiceClientStatusChangeListener mListener=new VoiceRecognitionClient.VoiceClientStatusChangeListener() {
    public void onClientStatusChange(int status, Object result) {
        Log.i(TAG, "onClientStatusChange");
        switch (status) {
            // 语音识别实际开始。这是真正开始识别的时间点。需在界面提示用户说话。
            case VoiceRecognitionClient.CLIENT_STATUS_START_RECORDING:
                IsRecognition = true;
                break;
            // 检测到语音起点
            case VoiceRecognitionClient.CLIENT_STATUS_SPEECH_START:
                break;
            //这里可以什么都不用作，简单地传入的数据做下记录
            case VoiceRecognitionClient.CLIENT_STATUS_AUDIO_DATA:
                break;
            // 已经检测到语音终点，等待网络返回
            case VoiceRecognitionClient.CLIENT_STATUS_SPEECH_END:
                break;
            // 语音识别完成，显示obj中的结果
            case VoiceRecognitionClient.CLIENT_STATUS_FINISH:
                UpdateRecognitionResult(result);
                IsRecognition = false;
                break;
            // 处理连续上屏
            case VoiceRecognitionClient.CLIENT_STATUS_UPDATE_RESULTS:
                UpdateRecognitionResult(result);
                break;
            // 用户取消
            case VoiceRecognitionClient.CLIENT_STATUS_USER_CANCELED:
                IsRecognition = false;
                break;
            default:
                break;
        }
    }
}
```

2、监听语音识别按钮

```

private View.OnClickListener voiceSwitchKeyboard = (v) -> {
    int id = v.getId();
    if(id == R.id.btn_voice) {
        Log.i(TAG, "voice: " + IsRecognition.toString());

        Log.i(TAG, "voiceSwitchKeyboard:" + IsRecognition.toString());
        if(IsRecognition) {
            // 停止响应
            mASREngine.stopVoiceRecognition();
            btn_voice.setBackgroundResource(R.drawable.ic_voice);
            IsRecognition = false;
        } else {
            // 需要开始新识别, 首先设置参数
            VoiceRecognitionConfig config = new VoiceRecognitionConfig();

            // 开启语义解析
            config.enableNLU();
            // 开启音量反馈
            // config.enableVoicePower(true);
            config.setSampleRate(VoiceRecognitionConfig.SAMPLE_RATE_8K); // 设置采样率
            // 使用默认的麦克风作为音频来源
            config.setUseDefaultAudioSource(true);
            // 下面发起识别
            int code = mASREngine.startVoiceRecognition(
                mListener, config);
            if (code == VoiceRecognitionClient.START_WORK_RESULT_WORKING)
            { // 能够开始识别, 改变界面
                Log.i(TAG, "能够开始识别" + code);
                btn_voice.setBackgroundResource(R.drawable.ic_voice_hover);
                IsRecognition = true;
            } else {
                Log.i(TAG, "启动失败" + code);
            }
        }
    }
}

```

3、实例化识别器对象

```

initData(),
initListener();

// 友盟APP统计
MobclickAgent.setScenarioType(this, MobclickAgent.EScenarioType.E_UM_NORMAL);
MobclickAgent.setDebugMode( true );

// 获取识别器对象
mASREngine = VoiceRecognitionClient.getInstance(this);
// 获取授权
mASREngine.setTokenApis(Constant.EXTRA_KEY, Constant.EXTRA_SECRET);
}

```

4、图像识别则由后端完成

```

public static ChatMessage ocrChat(String str)
{
    OkHttpClientManager http = OkHttpClientManager.getInstance();
    ChatMessage chatMessage = new ChatMessage();
    String ret = "图片识别失败";

    String json = creatParam(str);

    Log.e(TAG, "ocrChat:json:" + json);
    try {
        ret = http.post(Transfer + "?type=chat", json);
        chatMessage.setMsg(ret);
    } catch (IOException e) {
        e.printStackTrace();
    }
    Log.e(TAG, "ocrChat:" + ret);
    chatMessage.setDate(new Date());
    chatMessage.setType(ChatMessage.Type.INCOMING);
    chatMessage.setMsg(ret);
    return chatMessage;
}

```

7. 市场前景

小范围，垂直化场景依然具有成长空间；在中国由于历史文化的祭奠，家族观念根深蒂固，近些年西方文化的冲突，导致家族观念的落寞，但是每年都会有大批华人回过寻根，足见家族的重要性。其实，在西方文化中，家族的观念尤甚之。快节奏的生活，导致当代人沉迷于“快餐文化”一切讲究效率，而我则认为，沉淀的重要性远比发展更重要。所以这就是这个 APP 的出发点 - 沉淀。

那么推广前景可观的话，盈利空间则更甚之，大家都知道“微商”的疯狂和其利润，那么平台完全可以在成功推广后，严格把控的接入符合具体人群的产品，可接入第三方商城等，重要的时，可以根据用户的聊天习惯，通过大数据分析的用户的购物/使用习惯，通过离散用户的角度进行产品推广。

当然，聊天的场景大有可做，看当前微信就能看出 IM 场景的空间，最重要的是平台化。

而因，现有大量成熟 IM 产品，所以可以从垂直化的角度进行切入，以小范围的用户场景进入市场，慢慢运作，占有用户量。