链滴

利用程序算法分析北京新闻广播里都卖了啥好吃的

作者：MaidongAndYida

原文链接：https://ld246.com/article/1583318602129

来源网站：链滴

序

3 月 1 号听北京新闻广播的北广生活时间节目。两位男主持人在售卖一款深海无刺鳕鱼，并且本来六百的鳕鱼今天只卖 399。而且购买后是发放券，即随时可以兑换。就跟网购一样。被其中关键词所吸。由于我们不久后要去吃海鲜自助所以我想多搜集一些关于广播电台的推荐的食物以备将来所需。所在着手开发一款脚本用来获取相应的食材销售内容。

逻辑整理

- [北广生活时间节目的源文件，官网有 mp4 的音频文件， 下载列表]
- [首先将 mp4 文件的后缀名改为 mp3，然后就使用科大讯飞的语音转写功能，将语音文件转化为字]
- [利用大数据工具 IKAnalyzer 进行分词总结，找出此次录音文件中出现次数颇多的数个汉语词语]

科大讯飞语音转写

新用户在实名认证后，可以领取 5 小时的语音转写功能实际使用时间，足以支撑写完一个小 demo，个长达 1 小时的 MP3 文件，语音转写一次，需要大概十多分钟。

下载官方的 java-sdk 之后，项目可以直接运行，但是要修改一下配置文件：

```
# APP ID
app_id=your app id
# secret key
secret_key=your secret key
# we support both http and https prototype
lfasr_host=http://raasr.xfyun.cn/api
# file piece size
file_piece_size=10485760
```

```
# store path: 语音识别文件存放的根目录
store_path=D:\\
```

IKAnalyzer 分词

GitHub 上的 IKAnalyzer 给出建议的是在本地启动一个 IKAnalyzer 分词服务，然后调用链接传参数我不是很喜欢，还不如在 pom 里增加几个 maven 的 jar 包：

```
<!-- ikanalyzer 中文分词器  -->
<dependency>
   <groupId>com.janeluo</groupId>
   <artifactId>ikanalyzer</artifactId>
   <version>2012_u6</version>
   <exclusions>
      <exclusion>
         <groupId>org.apache.lucene</groupId>
         <artifactId>lucene-core</artifactId>
            </exclusion>
            <exclusion>
         <groupId>org.apache.lucene</groupId>
         <artifactId>lucene-queryparser</artifactId>
            </exclusion>
            <exclusion>
         <groupId>org.apache.lucene</groupId>
         <artifactId>lucene-analyzers-common</artifactId>
            </exclusion>
   </exclusions>
      </dependency>

<!--  lucene-queryparser 查询分析器模块 -->
<dependency>
   <groupId>org.apache.lucene</groupId>
   <artifactId>lucene-queryparser</artifactId>
   <version>7.3.0</version>
</dependency>
```

IKAnalyzer 的使用

1. IKAnalyzer.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
   <comment>IK Analyzer 扩展配置</comment>
   <entry key="ext_dict">extend.dic</entry>
   <entry key="ext_stopwords">stopword.dic</entry>
</properties>
```

1. extend.dic 扩展词典

```
这是一个
巨大的墙
```

1. stopword.dic 扩展停止词典，即此词典的值，不会转成文字

一个
我们
咱们
我
你
们
就是
一
个
的
1
2
3
4
5
6
7
8
9
10
回家
您
点
背
这是
那个
这个
里
今天
去
吃
吧
呗
呢
所以
好了
音乐
都是
唉
啊
广播
是
的
有
在
大家
什么
大
小
好
要
种

原文链接：利用程序算法分析北京新闻广播里都卖了啥好吃的

说
就
了
他
给
东西
滴
滴滴
会
但是
而且
嗯
对
人
包
来
这
那
滴答
盒
也
的人
自己
放心
大盒
一包
北
不是
让
起
营养
爱
真的
吗
上
非常
时候
最

分词工具类：

```
package com.example.demo;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.util.StringUtils;
import org.wltea.analyzer.core.IKSegmenter;
import org.wltea.analyzer.core.Lexeme;

import java.io.StringReader;
import java.util.ArrayList;
import java.util.List;

/**
```

```
 * 分词工具类
 */
public class IKAnalyzerSupport {

    private static final Logger LOGGER = LoggerFactory.getLogger(IKAnalyzerSupport.class);

    /**
     * IK分词
     * @param target
     * @return
     */
    public static List<String> iKSegmenterToList(String target) throws Exception {
        if (StringUtils.isEmpty(target)){
            return new ArrayList<>();
        }
        List<String> result = new ArrayList<>();
        StringReader sr = new StringReader(target);
        // 关闭智能分词 (对分词的精度影响较大)
        IKSegmenter ik = new IKSegmenter(sr, false);
        Lexeme lex;
        while((lex=ik.next())!=null) {
            String lexemeText = lex.getLexemeText();
            result.add(lexemeText);
        }

        //LOGGER.info("company:{}, iKSegmenterToList:{}", target, JSON.toJSON(result));
        return result;
    }
}
```

科大讯飞 +IKAnalyzer 使用代码：

```
package com.example.demo;

import java.util.*;

import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.iflytek.msp.cpdb.lfasr.client.LfasrClientImp;
import com.iflytek.msp.cpdb.lfasr.exception.LfasrException;
import com.iflytek.msp.cpdb.lfasr.model.LfasrType;
import com.iflytek.msp.cpdb.lfasr.model.Message;
import com.iflytek.msp.cpdb.lfasr.model.ProgressStatus;

/**
 * 非实时转写SDK调用demo
 * 此demo只是一个简单的调用示例, 不适合用到实际生产环境中
 *
 * @author white
 *
 */
public class LfasrSDKDemo {
```

```
    // 原始音频存放地址
    private static final String local_file = "D:/分词/demo/src/main/resources/audio/lfasr.wav";

    /*
     * 转写类型选择：标准版和电话版(旧版本, 不建议使用)分别为：
     * LfasrType.LFASR_STANDARD_RECORDED_AUDIO 和 LfasrType.LFASR_TELEPHONY_RECO
DED_AUDIO
     * */
    private static final LfasrType type = LfasrType.LFASR_STANDARD_RECORDED_AUDIO;

    // 等待时长（秒）
    private static int sleepSecond = 20;

    public static void main(String[] args) {
        // 初始化LFASRClient实例
        LfasrClientImp lc = null;
        try {
            lc = LfasrClientImp.initLfasrClient();
        } catch (LfasrException e) {
            // 初始化异常，解析异常描述信息
            Message initMsg = JSON.parseObject(e.getMessage(), Message.class);
            System.out.println("ecode=" + initMsg.getErr_no());
            System.out.println("failed=" + initMsg.getFailed());
        }

        // 获取上传任务ID
        String task_id = "";
        HashMap<String, String> params = new HashMap<String, String>();
        params.put("has_participle", "true");
        //合并后标准版开启电话版功能
        //params.put("has_seperate", "true");
        try {
            // 上传音频文件
            Message uploadMsg = lc.lfasrUpload(local_file, type, params);

            // 判断返回值
            int ok = uploadMsg.getOk();
            if (ok == 0) {
                // 创建任务成功
                task_id = uploadMsg.getData();
                System.out.println("task_id=" + task_id);
            } else {
                // 创建任务失败-服务端异常
                System.out.println("ecode=" + uploadMsg.getErr_no());
                System.out.println("failed=" + uploadMsg.getFailed());
            }
        } catch (LfasrException e) {
            // 上传异常，解析异常描述信息
            Message uploadMsg = JSON.parseObject(e.getMessage(), Message.class);
            System.out.println("ecode=" + uploadMsg.getErr_no());
            System.out.println("failed=" + uploadMsg.getFailed());
        }

        // 循环等待音频处理结果
```

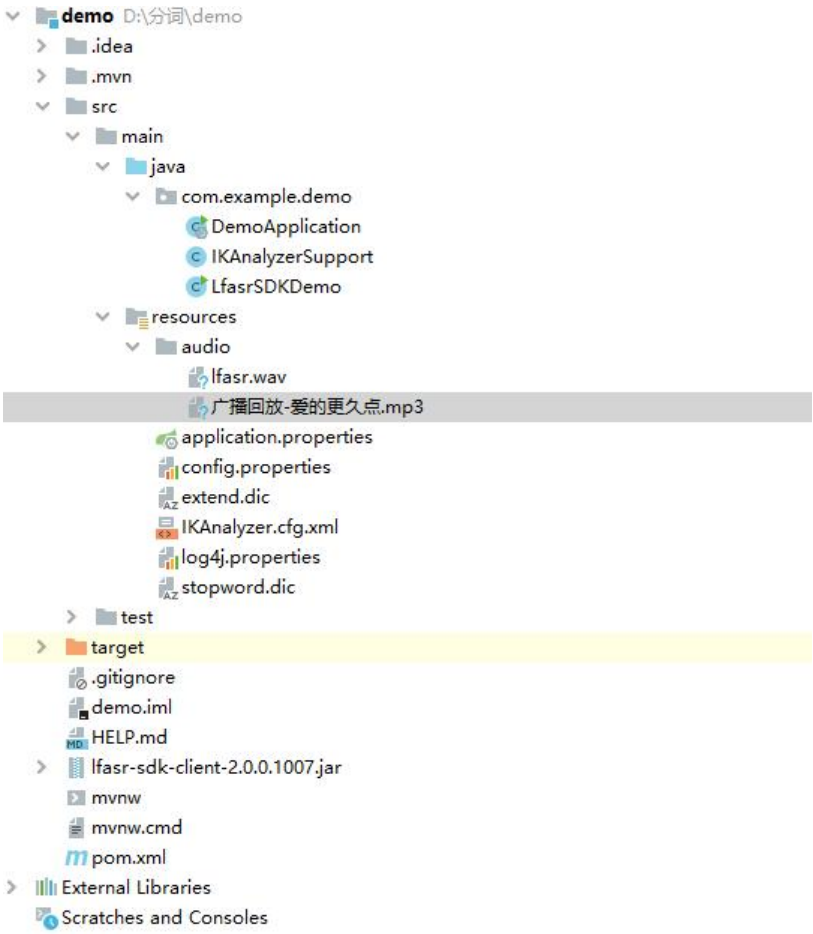
```
        while (true) {
            try {
                // 等待20s在获取任务进度
                Thread.sleep(sleepSecond * 1000);
                System.out.println("waiting ...");
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            try {
                // 获取处理进度
                Message progressMsg = lc.lfasrGetProgress(task_id);

                // 如果返回状态不等于0，则任务失败
                if (progressMsg.getOk() != 0) {
                    System.out.println("task was fail. task_id:" + task_id);
                    System.out.println("ecode=" + progressMsg.getErr_no());
                    System.out.println("failed=" + progressMsg.getFailed());

                    return;
                } else {
                    ProgressStatus progressStatus = JSON.parseObject(progressMsg.getData(), Prog
essStatus.class);
                    if (progressStatus.getStatus() == 9) {
                        // 处理完成
                        System.out.println("task was completed. task_id:" + task_id);
                        break;
                    } else {
                        // 未处理完成
                        System.out.println("task is incomplete. task_id:" + task_id + ", status:" + progre
sStatus.getDesc());
                        continue;
                    }
                }
            } catch (LfasrException e) {
                // 获取进度异常处理，根据返回信息排查问题后，再次进行获取
                Message progressMsg = JSON.parseObject(e.getMessage(), Message.class);
                System.out.println("ecode=" + progressMsg.getErr_no());
                System.out.println("failed=" + progressMsg.getFailed());
            }
        }

        // 获取任务结果
        try {
            Message resultMsg = lc.lfasrGetResult(task_id);
            // 如果返回状态等于0，则获取任务结果成功
            if (resultMsg.getOk() == 0) {
                // 打印转写结果

                String data = resultMsg.getData();
                JSONArray jsonObject1 = JSONArray.parseArray(data);

                StringBuffer sb = new StringBuffer();

                for (int i = 0; i < jsonObject1.size(); i++) {
```

```
                String o = jsonObject1.get(i).toString();
                JSONObject jsonObject = JSONObject.parseObject(o);
                String onebest = jsonObject.get("onebest").toString();
                sb.append(onebest);
            }

            try {
                List<String> strings = IKAnalyzerSupport.iKSegmenterToList(sb.toString());
                Map<String,Integer> countMap = new HashMap();

                strings.forEach(str ->{
                    if(countMap.containsKey(str)){
                        int o = countMap.get(str);
                        countMap.put(str,o+1);
                    }else{
                        countMap.put(str,1);
                    }
                });

                //排序
                List<Map.Entry<String, Integer>> list = new ArrayList<>(countMap.entrySet());
                Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
                    @Override
                    public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer>
o2)
                    {
                        //按照value值，重小到大排序
                        //return o1.getValue() - o2.getValue();

                        //按照value值，从大到小排序
                        return o2.getValue() - o1.getValue();
                    }
                });
                System.out.println(JSONObject.toJSONString(list));
            } catch (Exception e) {
                e.printStackTrace();
            }

        } else {
            // 获取任务结果失败
            System.out.println("ecode=" + resultMsg.getErr_no());
            System.out.println("failed=" + resultMsg.getFailed());
        }
    } catch (LfasrException e) {
        // 获取结果异常处理，解析异常描述信息
        Message resultMsg = JSON.parseObject(e.getMessage(), Message.class);
        System.out.println("ecode=" + resultMsg.getErr_no());
        System.out.println("failed=" + resultMsg.getFailed());
    }
  }
}
```

上面的原始音频是官方 sdk 自带音频，其运行结果如下：

```
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager
 Connection [id: 4][route: {}->http://raasr.xfyun.cn:80] can be kept alive indefinitely
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager
 Connection released: [id: 4][route: {}->http://raasr.xfyun.cn:80][total kept alive: 1; route alloca
ed: 1 of 2; total allocated: 1 of 20]
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager
 Connection manager is shutting down
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.DefaultManagedHttpClientConnection
 http-outgoing-4: Close connection
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.DefaultManagedHttpClientConnection
 http-outgoing-4: Close connection
18:40:48.973 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager
 Connection manager shut down
加载扩展词典：extend.dic
加载扩展停止词典：stopword.dic
[{"技术":1},{"商":1},{"语音":1},{"提供":1},{"大智":1},{"科大":1},{"中国":1},{"提供商":1},{"飞":1},{"讯":1},{
智能":1}]

Process finished with exit code 0
```

整体项目结构：



原文链接：利用程序算法分析北京新闻广播里都卖了啥好吃的