我的TensorFlow模型

我的TensorFlow模型选择的是智能回复

智能回复是一种与上下文相关的、一键式的响应，帮助用户高效、轻松地回复收到的文本消息(或电子邮件)。包括Gmail、Inbox和Allo在内的多个谷歌产品的智能回复都非常成功。

设备上的智能回复模型是针对文本聊天用例的。它的架构与基于云的同类产品完全不同，是专门为手机和手表等内存受限设备而设计的。它已经成功地用于在Android Wear上为所有第一和第三方应用程序提供智能回复。

但是，备上的模型的触发率比云上的模型要低(触发率是模型对传入消息建议响应的时间百分比)。

运作方式

设备上的智能回复演示应用程序的工作方式如下:

Android应用程序通过一个预测器库链接到JNI二进制文件。

在预测器库中，使用输入字符串列表调用getsegmentprediction。

2.1输入字符串可以是1-3条最近的对话消息，以字符串向量的形式输入。该模型将在这些输入语句上运行，并提供相应的智能回复。

2.2函数对输入数据进行预处理，包括:

分句:如果消息有多个句子，则输入消息将被分成多个句子。例如“how are you?”what is your name?会被分成两个不同的句子。

归一化:将个别句子归一化，将其转换成小写，去掉不必要的标点等。“how are you？？？？？”就会变成“how are you?”。

输入字符串内容将被转换为张量。

2.3函数对输入张量运行预测模型。

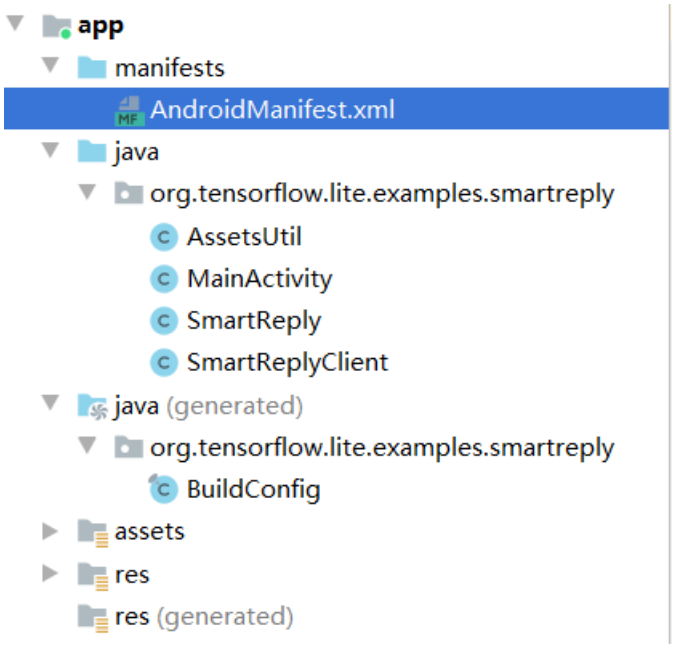
2.4该函数还进行了一些后处理，包括对2.2中输入的句子进行模型预测并返回相应的响应。

最后，从std::vector获取响应(s)，返回Android app。响应按照置信度降序排序。

如果模型没有触发任何响应，系统就会退回到建议来自一个固定的回退集合的响应，这个回退集合是根据聊天对话中观察到的流行响应意图编译而成的。

文件目录结构分析

这个模型的APP文件夹是这样的：



可以看到主要的JAVA文件有四个。

这四个文件的具体用处分析得知是：

AsstesUtil	读取资产文件
MainActivity	主界面组件渲染
SmartReply	输入字符串
SmartReplyClient	调用模型并输出

###

部分代码分析

加载TensorFlow模型

```
public synchronized void loadModel() {
  if (!isLibraryLoaded) {
    System.loadLibrary(JNI_LIB);
    isLibraryLoaded = true;
  }

  try {
    model = loadModelFile();
    String[] backoff = loadBackoffList();
    storage = loadJNI(model, backoff);
  } catch (IOException e) {
    Log.e(TAG, msg: "Fail to load model", e);
    return;
  }
}
```

如果没有加载模型，就加载JNI的模型，然后检测有没有加载成功，如果成功就调用读取字符串函数和回复函数

```
private MappedByteBuffer loadModelFile() throws IOException {
  try (AssetFileDescriptor fileDescriptor =
          AssetsUtil.getAssetFileDescriptorOrCached(context, MODEL_PATH);
      FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor())) {
    FileChannel fileChannel = inputStream.getChannel();
    long startOffset = fileDescriptor.getStartOffset();
    long declaredLength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
  }
}
```

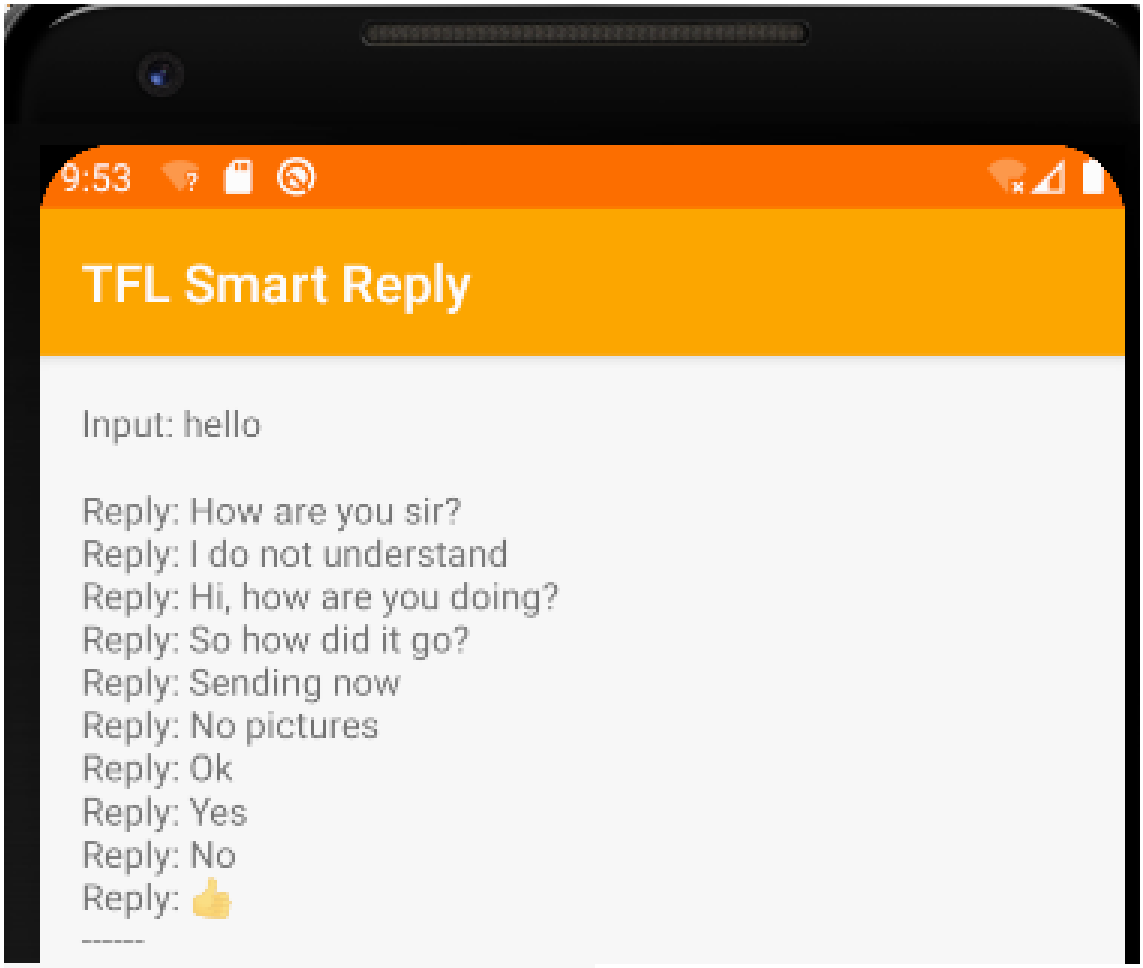
调用模型时所使用的函数，返回的后面两个参数应该是起始语句和字符串长度

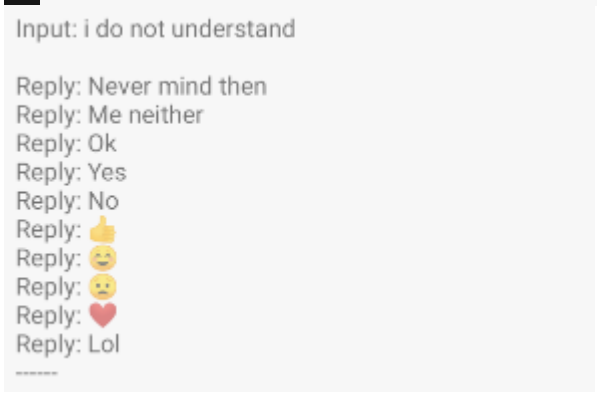
输入字符串并返回回复

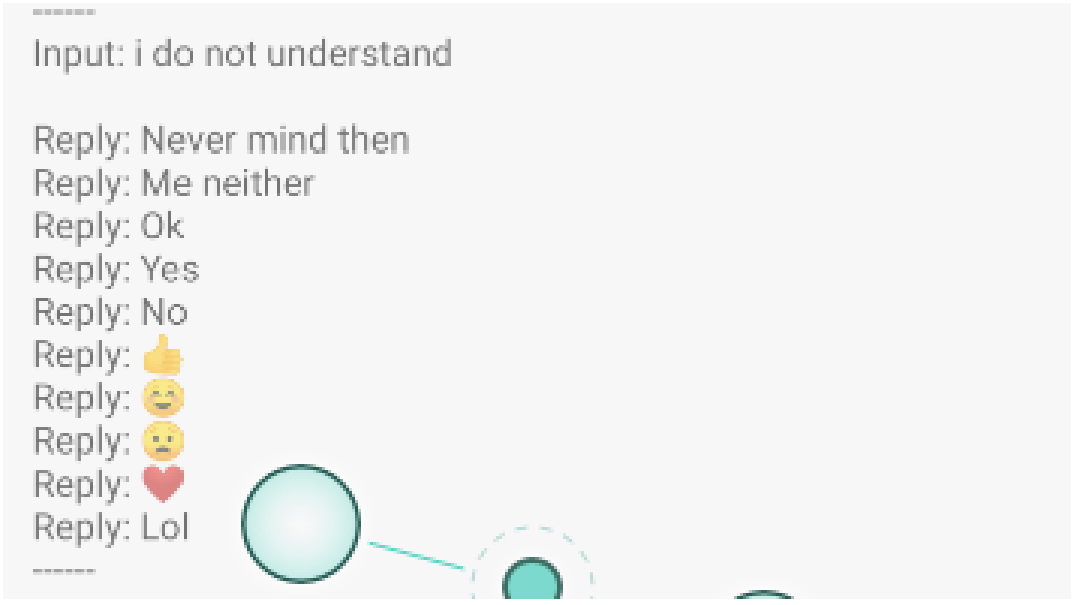
```
private String[] loadBackoffList() throws IOException {
  List<String> labelList = new ArrayList<~>();
  try (BufferedReader reader =
      new BufferedReader(new InputStreamReader(context.getAssets().open(BACKOFF_PATH)))) {
    String line;
    while ((line = reader.readLine()) != null) {
      if (!line.isEmpty()) {
        labelList.add(line);
      }
    }
  }
  String[] ans = new String[labelList.size()];
  labelList.toArray(ans);
  return ans;
}
```

labelist是输入的字符串， ans是回复

使用界面截图


9:53
TFL Smart Reply
Input: hello
Reply: How are you sir?
Reply: I do not understand
Reply: Hi, how are you doing?
Reply: So how did it go?
Reply: Sending now
Reply: No pictures
Reply: Ok
Reply: Yes
Reply: No
Reply: 👍


Input: i do not understand
Reply: Never mind then
Reply: Me neither
Reply: Ok
Reply: Yes
Reply: No
Reply: 👍
Reply: 😊
Reply: 😕
Reply: ❤️
Reply: Lol



Input: i do not understand
Reply: Never mind then
Reply: Me neither
Reply: Ok
Reply: Yes
Reply: No
Reply: 👍
Reply: 😊
Reply: 😕
Reply: ❤️
Reply: Lol

Input: lol

Reply: LOL haha
Reply: I knew that
Reply: That will be fun
Reply: You do that
Reply: Yeah great
Reply: Indeed it does
Reply: Ok
Reply: Yes
Reply: No
Reply: 👍
