【café速递】张帆：语音助手中的自然语言理解技术

核心提示：自然语言理解技术在人工智能领域有些广泛的应用，目前较为流行的应用是语言识别方向，小爱同学、Siri 和天猫精灵等常见的语音识别工具是基于自然语言理解技术实现的。本报告基于语音识别的自然语言理解技术原理，以小爱同学的语音识别应用为例，加深对自然语言理解技术的算法解释。

人物名片：张帆，武汉大学测绘遥感信息工程国家重点实验室 17 届博士生，师从张良培教授，发表 SCI 论文 5 篇，Google Scholar 引用 1000+次。曾任职于阿里巴巴人工智能实验室-天猫精灵算法专家。目前任职于小米人工智能部-小爱同学，负责自然语言处理算法部分。

报告现场：2020 年 11 月 7 日下午 3 点，武汉大学测绘遥感信息工程国家重点实验室 17 届博士生张帆做客 GeoScience Café第 276 期讲座。张帆以自然语言理解技术的框架为主题，从自然语言理解技术的算法流程思路出发，详细介绍每个流程的具体含义。在该基础上，张帆以小爱同学的具体应用为实际例子，阐述了小爱同学语音识别背后的原理。

图 1 报告现场

1. 自然语言理解技术的算法基本流程

如图 2 所示，自然语言理解技术的整个系统主要由 ARS, NLU,

Ranking/Dialog Manager, TTS 组成。

其中，ARS (Automatic Speech Recognition) 的作用为将输入的语音转化为文本；NLU(Natural Language understanding)的作用为将 ARS 处理得到的文本进行关键词识别，挖掘语音表达意思的关键词；Ranking/Dialog Manager (DM)的作用是对 NLU 挖掘出的关键词信息结合语音转码之后的主要文本内容进行执行操作，实现用户需求；TTS (Text-to-Speech) 的作用则是将输入的文本信息综合转化为语音输出。

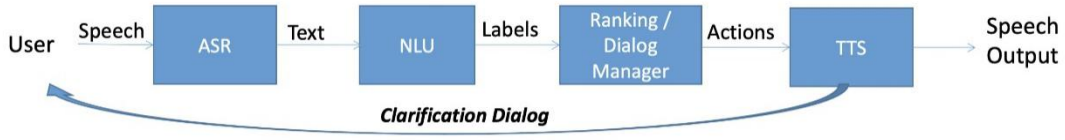

图 2 自然语言理解技术实现的流程图

表 1 自然语言理解技术对应流程的定义及实例

Component	Input	Output	Example
Automatic Speech Recognition (ASR)	Speech	Text (1-best or lattice of alternatives)	"Play Two Steps Behind by Def Leppard"
Natural Language Understanding (NLU)	Text	Slots and Intent Type	**Intent**: PlayMusicIntent **Slots**: Artist Name=Def Leppard Song Title: Two Steps Behind
Ranking / Dialog Manager (DM)	Labels & Context	Dialog Actions	Ask the application to play the song or clarify
Text-to-Speech (TTS)	Text	Speech	*"Which artist?"* or *"Playing Two Steps Behind by Def Leppard"*

在实际语音识别处理过程中，自然语言理解系统的工作流程如下：首先，ARS 将用户输入的语音转化成文本。其次，NLU 将转出的文本内容进行关键词识别，目的是理解文本的关键信息和主要内容。然后，DM 将 NLU 挖掘出的关键词信息及文本主要内容进行执行操作，满足用户的需求。最后，TTS 将文本信息根据语言习惯转化为个性化语音输出。

2. ASR 系统介绍

ASR 作为自然语言理解技术的重要组成部分，其主要功能是将输入的语音信息转化为高准确率的文本信息。

传统的 ASR 识别流程如图 3 所示，主要包括：(1) 首先，将输入的语音进行特征提取；(2) 其次，应用 DNN/RNN Acoustic Model 将提取出来的特征进行进

一步处理；(3)然后，将 DNN/RNN Acoustic Model 处理之后得到的结果进行解码；(4)最后，对初步解码的信息做进一步分析，进行二次解码，然后输出文本。

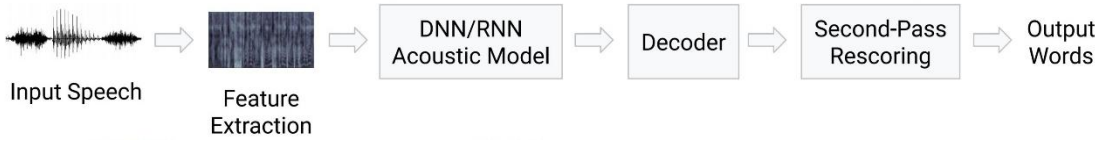

图 3 传统 ASR 模型的结构及处理流程

为了简化语音识别并转化为文本的过程，实现文本直接输出的功能，基于典型语音系统 Acoustic Model, Pronunciation Model, Verbalizer, Language Model, 2nd-Pass Rescoring 的原理，新型 ASR（主要是基于
End2End Trained Sequence-to-Sequence Recognizer 的原理）被开发出来。
End-to-End ASR 是目前被广泛应用的新型 ASR，该系统可将输入的声学特征序列直接映射一系列单词或字母。而且，在经过训练后，该系统可优化语音识别的评估指标，降低转化文本中单词的错误率。关于 End-to-End ASR，其核心算法主要有
CTC, Attention-based Encoder Decoder Models, Contextual LAS Models, Optimization improvements, Endpointer, End-of-query detector.

CTC (Connectionist Temporal Classification)的主要功能是对输入的语音序列逐一转化进行编码转化为相应的字母；Attention-based Encoder-Decoder Models 则将语音序列转化为更高级别的编码，再进一步构建模型进行解码；相较于 Attention-based Encoder-Decoder 而言，Contextual LAS Models 在编码过程中加入了误差因子以提高模型的准确性。

Optimization improvements 结合了编码及解码构建模型的过程，将语音输入解码结果与潜在用户信息进行融合，对文本转化单词错误率进行评估，挖掘出错误率最低的单词作为最佳匹配结果。

Endpointer 的功能主要是快速准确地确定用户何时完成发言，其具体工作流程如图 4 所示。

End-of-query detector 则是将语音识别输入过程按时间和信息综合分成四个阶段：(1) Speech，语音输入阶段，探测器能够探测到语音的时间段，用数字 0 表示；(2) intermediate non-speech，语音序列间隔空窗期，该阶段对应为相邻两个语音序列片段之间无语音输入的阶段，用数字 1 表示；(3) initial non-speech，语音片段输入之前的空窗期，即第一个语音片段被检测识别前的无声期，用数字 2 表示；(4) Final non-speech (end-of-query)，语音片段输入结束之后的空窗期，

即最后一个语音片段被检测识别之后的无声期，用数字 3 表示表示。

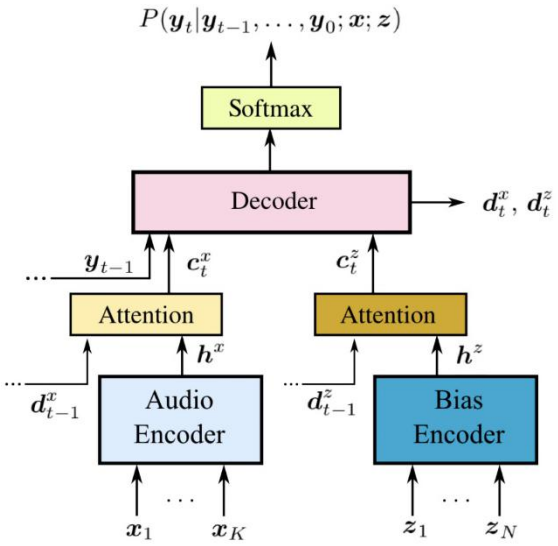

图 4 Optimization improvements 算法流程

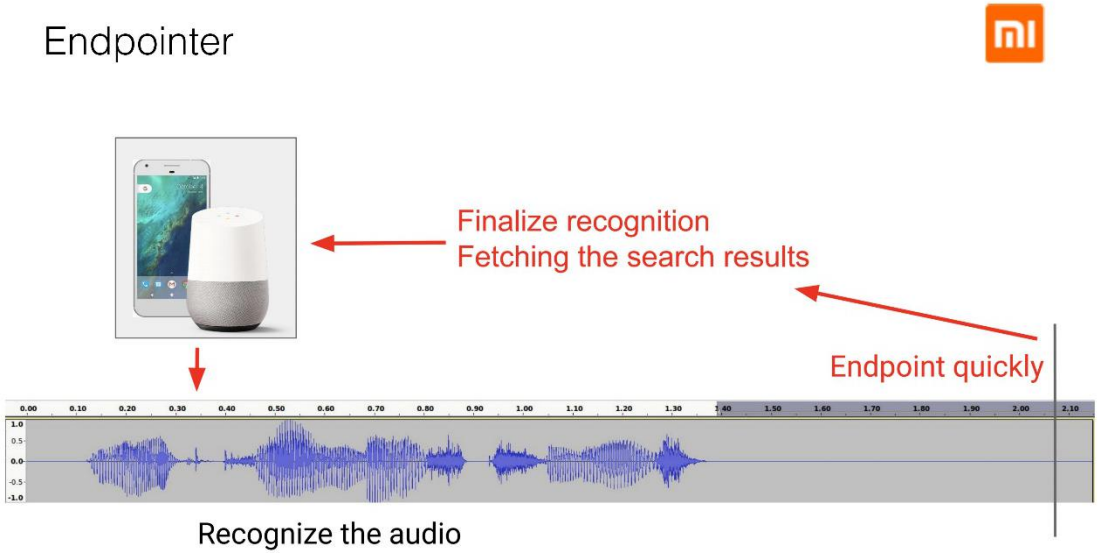

图 5 Endpointer 工作流程

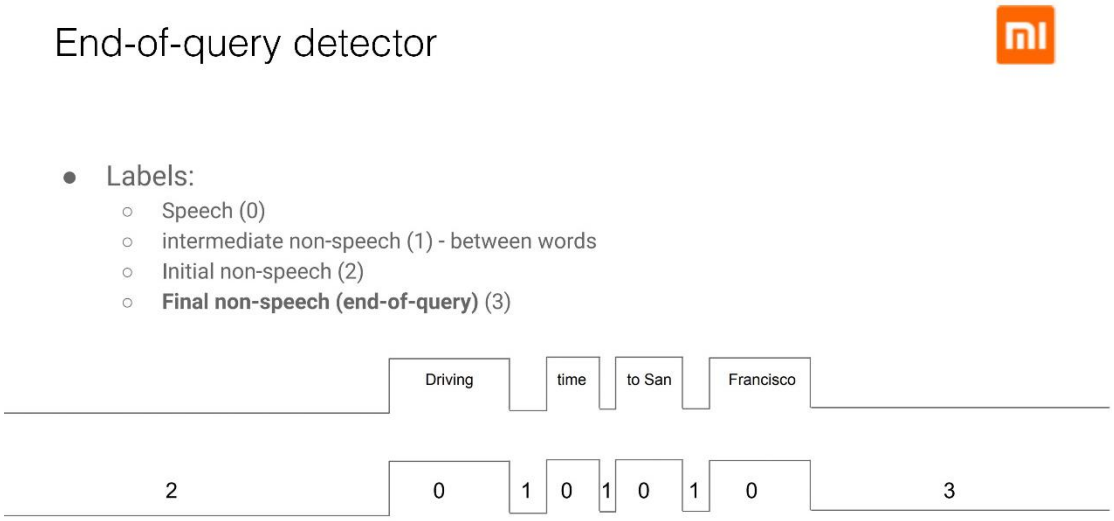

图 6 End-of-query 工作流程

3. 小爱同学应用实例

当语音输入“播放音乐”时，语音识别助手将通过自然语言理解系统进行语音识别，挖掘出语音的领域识别为“音乐”，识别意图为“播放（无实体）”，此时将随机播放一首歌曲；在改基础上补充输入“下一首”时，系统将在语音输入的第一个片段基础上进行领域继承，并进一步挖掘出公共意图识别为“intent.next”，系统输出另一首歌；若进一步加入语音片段“我要听芒中”，则系统会进行 Query 改写，将领域识别为“音乐”，识别意图为“播放（实体：Song）”，实际上，“芒中”原本表达的意思是“芒种”这首歌，基于“芒种”本身的流行性和知名度，系统会进行相应的改正，能将“芒中”识别成“芒种”，在原有对应的实体库中找到“芒种”这首歌并播放，最终满足用户需求。

若语音片段输入“我要听青花瓷”，语音识别助手将通过自然语言理解系统进行语音识别，挖掘出语音的领域识别为“音乐”，识别意图为“播放（实体：Song）”，进一步补充语音片段“周杰伦的”，则系统进行多轮领域&实体继承，进一步识别出“实体：Artist(周杰伦)”，根据识别的结果，系统最终将播放周杰伦的《青花瓷》以满足用户需求。

当输入语音片段为“播放小猪佩奇”时，语音识别助手将通过自然语言理解系统进行语音识别，挖掘出语音的领域识别为“音乐-视频”，进行 Rank 决策，播放关于小猪佩奇的音频。

除了“音乐”领域识别外，当用户输入语音片段“我要签到”，则系统识别

出技能匹配为“签到技能”，并将意图识别为“签到（无实体）”；而当输入语音为“打开空调”，系统将会挖掘出语音的领域识别为“家居”，意图识别为“打开（实体：Device）”，系统将会相应打开空调，若在此基础上补充语音“现在温度”，则系统进行多轮领域继承，将进一步将意图识别为“属性查询”，语音输出对应空调设置的温度。

4. 提问环节

问题 1：张帆师兄，您好！我有两个问题想请教一下，第一个是你们的训练集是怎样获取的，然后训练集的体量大概有多大？然后第二个问题就是比如像 Siri 那样，刚开始用的时候 Siri 语言助手会叫我说两三句话，然后语音助手就能识别这个人是我，然后后续就能自动识别出我的声音，然后这个过程才能实现？

回答 1：目前来说，一种是外包去实现，比如现在要实现音乐的功能，我们会跟外包说一下，让一些他们来帮我们写一些模板。我们把这些模板作为一个外包的服务发布出去，让每个同学根据自己想要实现的功能，写成对应的话，然后把这些话返回过来，变成一个基本的数据，然后来进行训练。目前每个人以音乐为例，应该是有几十万条的数据。我们有几十万这种数据，同时它的领域意图和实体都是以标注好的信息来进行训练的。如果要配置业务的话，对说话人声音的识别，就是系统会把你的音频录下来，其实就类似于人脸匹配的方式一样，我会把你音频上的特征当作一个模板存在的手机上，然后当你在进行新的音频录入时，我们会进行声纹识别，来判断新进来语音的特征和你录在本地的特征是否是相似或者对应一致的。Apple 和天猫里面其实都有类似的功能，小爱同学后续应该也会有。

问题 2：张帆师兄，您好！因为我们自己家里有小爱同学这样的语音识别助手，然后其中有一个我一直很好奇的事情，就是你在家里或者在一个公开区域里和小爱同学说话时，会存在你跟小爱同学在说话的过程中受到很多背景音干扰的现象。如果这区域内有其他人在说话，那小爱同学是怎么识别出用户声音和其他的一些背景声音之间的差异？怎么去获取用户的输入的语音信息？比如我和小爱同学说话的同时，其他人也说了话，小爱同学在这种情况下是怎么避免干扰的？

回答 2：确实是存在这种情况的。其实这部分是在 ASR 那部分实现的。有音频进来的时候，我们会对你的音频进行一些背景的消除，噪声消除的实现 ASR 有相应的处理过程，还有另外一部分更先进的算法。当你唤醒音箱以后，我们会通过唤醒词来判断这次唤醒的声音是谁来唤醒的，然后能根据唤醒词的特征来增强跟唤醒声音相似的音频，同时压抑非相似的音频信号，这种处理方式的优势就是唤醒音箱的这个人，他所说的话肯定是最准确的，同时把别人的声音给消除，而且唤醒音箱的同时，我们也会跟踪这个音频信号是来自于哪个方向的，增强这个方向的音频信号，同时抑制掉其他地方的一些信号来对与唤醒音箱音频相似的信号进行增强，然后来确保 ASR 的识别效果。