

NLU Framework for Voice Enabling Non-Native Applications on Smart Devices

Soujanya Lanka, Deepika Pathania, Pooja Kushalappa, Pradeep Varakantham

i.am+ LLC. 809 North Cahunga Blvd, Los Angeles - 90036
{soujanya, deepika.pathania, pooja, pradeep}@iamplus.com

Abstract

Voice is a critical user interface on smart devices (wearables, phones, speakers, televisions) to access applications (or services) available on them. Unfortunately, only a few native applications (provided by the OS developer) are typically voice enabled in devices of today. Since, the utility of a smart device is determined more by the strength of external applications developed for the device, voice enabling non-native applications in a scalable, seamless manner within the device is a critical use case and is the focus of our work.

We have developed a Natural Language Understanding (NLU) framework that uses templates supported by the application (as determined by the application developer). This framework can be employed in any mobile OS (Android, iOS, Tizen, Android wear) for a wide range of devices. To aid this demonstration, we have implemented the framework as a service in Android OS. When the user issues a voice command, the natural language query is obtained by this service (using one of local, cloud based or hybrid speech recognizers). The service then executes our NLU framework to identify the relevant application and particular action details. In this demonstration, we will showcase this NLU framework implemented as an Android service on a set of applications that will be installed on the fly. Specifically, we will show how the voice queries are understood and necessary services are launched on android smart wearables and phones.

Introduction

Smart devices, these days (wearables, speakers, televisions, phones) are characterized by good processors and memory, which enable multiple capabilities such as location awareness, activity awareness and multi-tasking (running multiple applications at once). Such powerful devices need an engaging and hands free user experience, interface for the user to perform various operations and to fully utilize the

capabilities of the device. Voice seems to be the best option to ensure a reliable user experience.

Voice assistants like Siri, Google Voice, Cortana are already present in many smart phones and smart watches. Unfortunately, these existing voice assistants limit the user commands to a few native applications like e-mail, music, messaging, calendar and clock/alarms and do not service non-native applications. By non-native applications, we refer to the ones that are developed by third party developers and are available on AppStores (like iTunes and Google Play) for the users to download and install.

For example, a private cab-booking application, developed by a cab-services company, cannot be associated with a voice command like 'get me a cab' as the application is not a native application. We believe that this inability to handle valid user commands related to third party installable applications is a major limitation in existing voice assistants.

To bridge this limitation with existing voice assistants, we have prototyped an existing voice assistant, AneedA (aneed A) with the additional capability of associating voice commands to non-native applications. In the following section we elaborate on the framework architecture and the workflow associated with this framework.

NLU Framework for Voice-Enabling Applications

Our underlying idea is to bundle non-native applications with a file that defines the grammar for the various natural language command templates that the application can service. Upon installing such a grammar bundled application from AppStore, the NLU framework checks presence of grammar files and prepares ground to map the voice commands to the applications.

Implementation details

We have developed the NLU framework as an Android service that runs on the device. This NLU service works in sync with i.am+ in-house developed voice assistant

AneedA. When a third party application is installed from the AppStore, the service checks for any bundled grammar file. If a grammar file is found, it registers the grammar file to the application's functionality and adds this grammar to its database.

The sequence of operations (shown in Figure 1) upon receiving a voice command is as follows:

- 1 When user issues a voice command, the natural language query is obtained by using a (local/cloud based/hybrid) speech recognizer.
- 2 If the voice assistant cannot service the natural language query with native applications, then the command is passed on to the local NLU service.
- 3 The command is checked against the list of registered grammars maintained by the local NLU service.
- 4 If it matches one particular application's grammar, it launches the associated action along with the particular action details that could be parsed.
- 5 If it matches grammars of multiple applications (ex: multiple cab-booking applications can have similar query templates), then the voice assistant will seek input from the user on the application of interest.
- 6 If the query does not match any, then it returns a FAILURE event back to the Voice Assistant, which then speaks out that the query could not be understood/serviced.

For example, with the help of this framework, a flights search application could be voice-enabled for voice commands like 'find flights from <origin> to <destination>' and 'search for flights from <origin> to <destination>'. The application developer creates a grammar file and launches Flights app functionality based on the specific user input. Please refer to Figure 1, screenshot (a) displays the user's speech recognition screen and in screenshot (b), the Flights app receives the data parameters and displays results as per the query.

Grammar-based NLU

We have based the NLU module on the grammar modules available in CMU SPHINX 4 (Ziad Al Bawab 2009). The

Android NLU service that is built expects a JSGF grammar (Java Speech Grammar) file. The grammar file should list the command templates along with the data tags that need to be sent to the application upon successful match. The grammar file is expected to be a set of simple, dis-ambiguous commands. In the near future, we expect to have a component that will check for any ambiguity and errors in grammar files.

As new applications are installed from the AppStore, the service checks for bundled grammar files to register and enlarge its database of grammar files; thus making it scalable as more apps are installed.

Demonstration

During the demo, we will showcase a wide variety of non-native applications that can be triggered through voice commands. Specifically, for multiple non-native applications of varying complexity, we will show creation of the grammar database and execution of voice queries. To demonstrate generality across smart devices, we will also show case execution on a smart watch and a smart phone.

Acknowledgments

We would like to thank Will.i.am (CEO, i.am+), Chandrasekar Rathakrishnan (CPO, i.am+), TVS Deepak, Vids Samanta, Yuan-Yi Fan and Ranu Upadhyay for their valuable feedback on the NLU framework and the support in developing a set of demonstrable voice-enabled android applications.

References

- Java Speech Grammar Format, <http://www.w3.org/TR/jsgf/>
- AneedA, <http://www.nuance.com/company/news-room/press-releases/iamAneedA.docx>
- Ziad Al Bawab, 2009. An Analysis-by-Synthesis Approach to Vocal Tract Modeling for Robust Speech Recognition, Ph.D. Thesis, ECE Department, CMU.

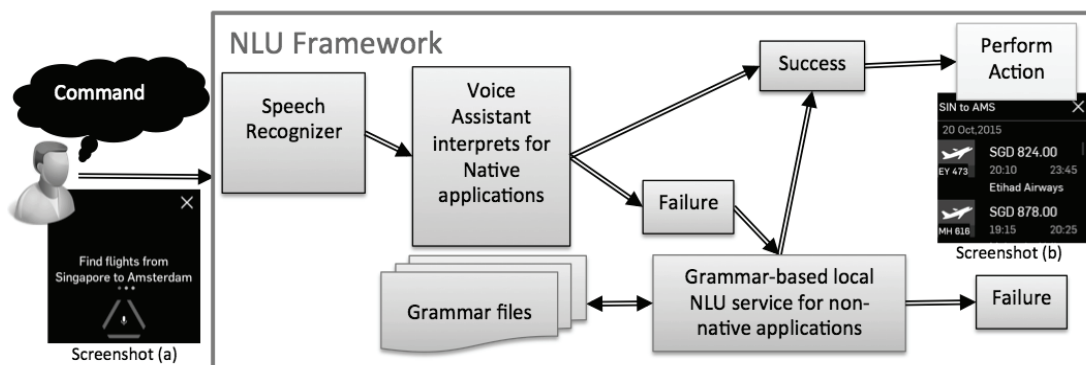


Figure 1. NLU Framework to Voice Enable Non-Native Applications