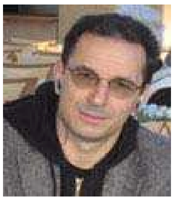
e-Conference & Webinar



Paolo Ferroni

DSP Application Day

2010

December 13, 2010

Android-based Automatic Speech Recognition front-end to access Web APIs

Mario Malcangi*, Paolo Ferroni**, Manuele Ettore Suzzani***

** DICO - Università degli Studi di Milano*
*** Eventi Telematici s.r.l. (www.evtel.com)*
**** Università degli studi di Milano Bicocca*

malcangi@dico.unimi.it
paolo.ferroni@eventitelematici.com
manuele.suzzani@n-3.it

Android-based Automatic Speech Recognition Front End to Access Web APIs

Mario Malcangi*, Paolo Ferroni, Manuele Ettore Suzzani*****

** DICO - Dipartimento di Informatica e Comunicazione Università degli Studi di Milano malcangi@dico.unimi.it*
*** Eventi Telematici s.r.l., paolo.ferroni@eventitelematici.com*
**** Università degli Studi di Milano Bicocca, manuele.suzzani@n-3.it*

Abstract: An automatic speech-recognition (ASR) front-end was implemented on Android platform to access RESTful API. The purpose was retrieving information from the Internet using speech as primary input. The Android application uses an ASR front-end to recognize uttered words and transform them into text patterns. Then it exploits on board GPS data and RESTful Twitter API to find twits containing the recognized text and coming from users located in a certain area around the device. Implementations of the same algorithm in C and in Java were compared, and performance on the Android platform was evaluated.

Key words: automatic speech recognition, Android, mobile development.

1. Introduction

Android is an open source operating system designed for mobile devices. It comes with a Java SDK and a valuable set of libraries to control on-board sensors and access web services.

In this work we want to investigate the following issues.

1. The complexity for implementing ASR algorithms on the Android platform and related performance.
2. The capability and handiness of programmatically using on-board sensors, like GPS, accelerometers, magnetometers, cameras, and so on.
3. The integration level with web services, in particular RESTful-based services and Google Maps.

1.1 Porting of the ASR algorithms

We started with an already implemented and well known prototype: a Matlab model developed at DICo [1], [2], [5], [6]. It recognizes a word in a set of isolated words by comparing the input voice with the information in a database words. A distance is calculated between each database element and the input pattern, the recognized word has the minimum distance from the input [8], [9], [11].

In order to minimize troubles due to numerical precision, we made an intermediate porting to ANSI-C [4], getting, as a side effect, a high performance and platform independent porting of the Matlab model. Actually, this represented the most of the work, mainly because of the various differences between Matlab and C in data and memory management and the lack of high-level constructs in C. For example, simple vector operations in Matlab become nested loops in C, dynamic range and matrix resize must be translated into complex memory-allocation procedures. Finally, the C version allowed us to assess the real performance of the speech-recognition model.

1.2 Porting to Java and Android

Once we obtained a stable C code, porting it to Java was very straightforward, and the better memory management was appreciated. Then, in a couple of days, we were up and running with the ASR engine on an Android smartphone [7], [10].

2. Interacting with web services

At this point we moved on to the web integration. Basically, the application we built is illustrated in the figure below.

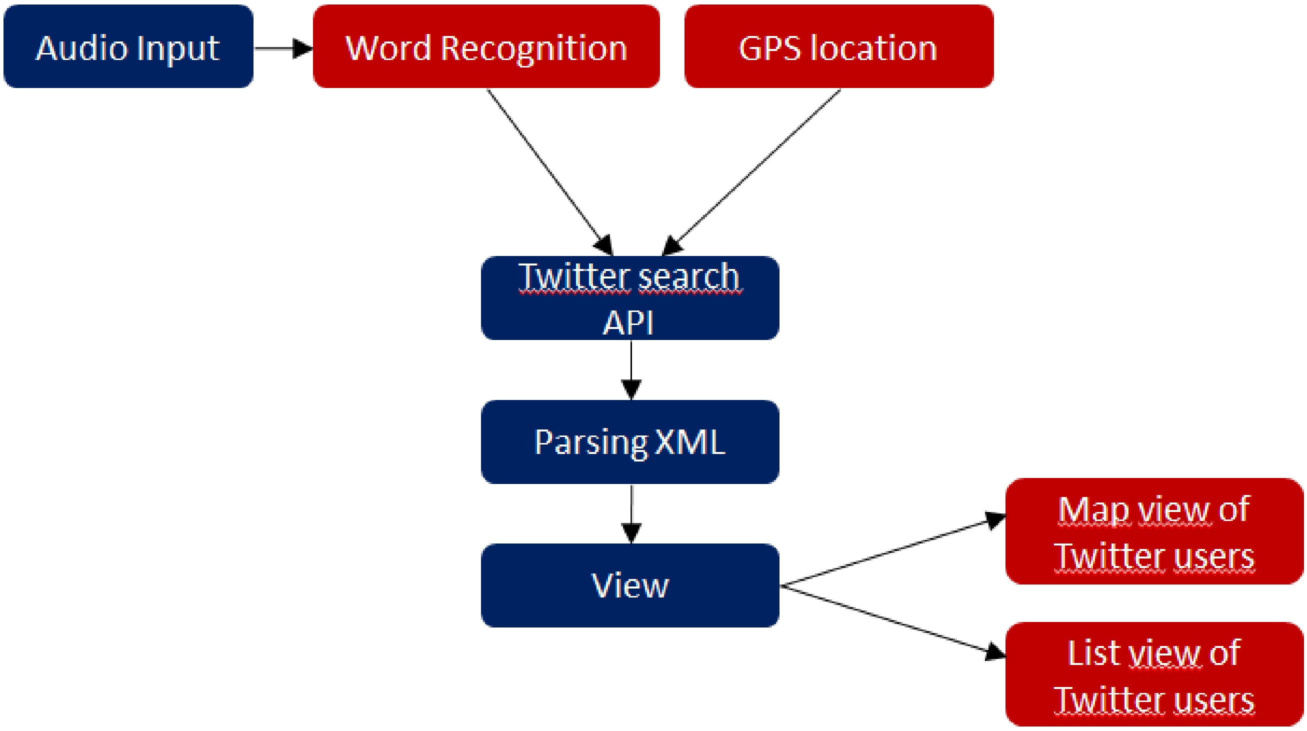


Figure 1. Simplified diagram of the application architecture.

The user pronounces a word, which is recognized by the system and put together with the GPS coordinates of the device. The information is sent to a Twitter API that returns an XML list of tweets posted by users located in a certain area around the device (for instance within a 10-km radius). This XML is parsed using the Android XML-manipulation classes, so that the application presents two views: the HTML list view and the geographic view, i.e. a map displaying the users' position.

Of course, this application is just a proof of concept, useful to address the above-mentioned points. These the main Android SDK packages utilized:

GPS	*android.location.Location;* *android.location.LocationListener;* *android.location.LocationManager;*
Google Maps	*com.google.android.maps.GeoPoint;* *com.google.android.maps.MapView;* *com.google.android.maps.Overlay;* *android.location.Address;* *android.location.Geocoder;*
WebView	*android.webkit.WebView ;*
Twitter Search	*org.apache.http.*;* *javax.xml.parsers.*;* *org.xml.sax.*;*

Figure 2. Screenshots showing the list and map views.

3. Performance comparison among the ASR implementations

The table below shows the average recognition time in three different environments.

LANGUAGE	SYSTEM	AVERAGE TIME (ms)
C	Windows XP Intel 2 Core Duo 2.00Ghz	36
JAVA	Windows XP Intel 2 Core Duo 2.00Ghz	65
JAVA	ANDROID 2.1 SnapDragon 1.00Ghz	3121

As expected, C version on PC is the fastest and JAVA on Android is the slowest. This is quite obvious because of language and platform differences. We were rather surprised by the closeness performance between C and JAVA on PC, and we consider the Android's figure acceptable. We reasonably think that three seconds may decrease further using some optimization technique and become even better using C code for computationally intensive tasks (it is possible to do so on Android). Anyway, no significant delay has been experienced using the application.

4. Conclusion

We try to answer our initial questions in the Implementing ASR algorithms on Android environment may be quite easy, provided that most of the work be already done with C. Needless to say, if you have a well-designed and well-coded application you surely reduce your effort to implement it on Android. Android has a good SDK, which allows developing programs that make (good?) use of the on-board sensors. They may be substantial in number and mixed in various ways to develop new types of applications. The integration level with the web is fine through REST APIs, but we really felt the lack of support for Javascript APIs, so widely used on the web.

References

[1] M. Malcangi, Soft-computing Approach to Fit a Speech Recognition System on a Single-chip, in *2002 International Workshop System-on-Chip for Real-Time Applications Proceedings*, Banff, Canada, July 6-7, 2002.

[2] M. Malcangi, Improving Speech Endpoint Detection Using Fuzzy Logic-based Methodologies, in: *Proceedings of the Thirteenth Turkish Symposium on Artificial Intelligence and Neural Networks,* Izmir, Turkey, June 10-11, 2004.

[3] Haseman Chris, *Android Essential*, Apress, 2008.

[4] Kumar Ram, Agrawal Rakesh, Programming in ANSI C, West Publishing Company, 1991.

[5] Pisoni Roberto, Sistema robusto di identificazione biometrica del parlatore tramite combinazione dei processi di riconoscimento dell'impronta vocale e di parole chiave, Tesi di laurea, Università degli Studi di Milano, 2005.

[6] Minardi Cecilia, Estrazione delle caratteristiche vocali e di riconoscimento di parole isolate finalizzato alla realizzazione di un sistema robusto di identifcazione biometrica, DICo-DSP&RTS Università degli Studi di Milano, 2004-2005, b.

[7] Naughton Patrick, Morrison Michael, The Java Handbook, Osborne/McGraw-Hill, 1996.

[8] O'Shaughnessy Douglas, *Speech Communication Human and Machine*, Addison Wesley ED, 1999.

[9] Parson W. Thomas, *VOICE AND SPEECH PROCESSING*, Mc Graw-Hill Book Company ED, 1986.

[10] Rogers Rick, Lombardo John, Mednieks Zigurd, Meike Blake, *Android Application Development*, O'Reilly Media, 2009.

[11] Smith W. Steven, *The scientist and Engineer's Guide to DIGITAL SIGNAL PROCESSING*, Analog Device ED,1997.

Web sites

Minimalist GNU for Windows, 06/01/2010,
http://www.mingw.org/

Code Blocks, The open source, cross platform, IDE, 06/01/2010,
http://www.codeblocks.org/

Java JDK SE 1.6 update 18, 02/04/2010,
http://java.sun.com/javase/

ECLIPSE GALILEO Release 3.5.2, 02/04/2010,
http://www.eclipse.org/galileo/

Java sound API documentation, 11/04/2010,
http://java.sun.com/j2se/1.5.0/docs/guide/sound/programmer_guide/contents.html

Android 2.1 SDK release 6, 21/04/2010,
http://developer.android.com/sdk/index.html

Android 2.1 SDK reference page, 23/04/2010,
http://developer.android.com/reference/packages.html

Twitter search API documentation, 10/05/2010,
http://apiwiki.twitter.com/Twitter-API-Documentation

About Twitter, 10/05/2010,
http://twitter.com/about
Introducing JSON, 11/05/2010,
http://www.json.org/
The Atom Syndication Format, 11/05/2010,
http://atompub.org/2005/07/11/draft-ietf-atompub-format-10.html
Using Google Maps on Android (tutorial), 15/05/2010,
http://mobiforge.com/developing/story/using-google-maps-android
Open Hanset Alliance, 10/06/2010,
http://openhandsetalliance.com/
About REST service, 10/05/2010,
http://it.wikipedia.org/wiki/Representational_State_Transfer