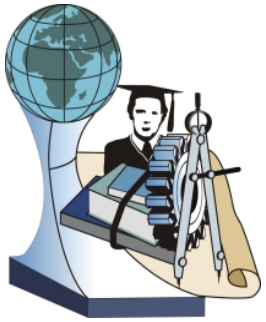

FIATAL MŰSZAKIAK
TUDOMÁNYOS ÜLÉSSZAKA XIX.

Kolozsvár, 2014. március 20–21.

LAPS CSR: FELHŐ ALAPÚ NYÍLT ELÉRHETŐSÉGŰ BESZÉDFELISMERŐ RENDSZER

LAPS CSR: A FREE DISTRIBUTED CLOUD SPEECH RECOGNITION SYSTEM

Cassio TRINDADE BATISTA[(1)], Thiago BARROS COELHO[(2)], Bruno Gomes HAICK[(3)], Nelson Cruz SAMPAIO NETO[(4)], Aldebaro Barreto da ROCHA KLAUTAU Jr[(5)].

(1) Óbuda University: 1034 Hungary, Budapest, Bécsi út, 96/b; Tel: +36-70-2061915; Email: cassio.batista@itec.ufpa.br

(2) Óbuda University: 1034 Hungary, Budapest, Bécsi út, 96/b; Tel: +36-30-9971096; Email: thiago.coelho@itec.ufpa.br

(3) Signal Processing Laboratory - UFPa: 66075110 Brazil, Belém, Rua Augusto Corrêa, 1; Tel: +55-91-32017674; Email: bghaick@ufpa.br

(4) Signal Processing Laboratory - UFPa: 66075110 Brazil, Belém, Rua Augusto Corrêa, 1; Tel: +55-91-32017674; Email: nelsonneto@ufpa.br

(5) Signal Processing Laboratory - UFPa: 66075110 Brazil, Belém, Rua Augusto Corrêa, 1; Tel: +55-91-32017674; Email: aldebaro@ufpa.br

Abstract

This paper describes a cloud speech recognition service based on Julius decoder running in server mode. The system was set up to recognize speech in Brazilian Portuguese. The support to the language was developed by the authors with FalaBrasil research group tools, which are free and available on the group's site. Julius uses the FalaBrasil cloud to provide online and distributed speech recognition via Internet. The client side was built on the Android 2.2 platform. The application can record and send audio, detect the end of the user speech and listen to the decoder result. To test the system efficiency, recognition time and accuracy rate were estimated by comparing it to *SpeechRecognizer* API provided by Google.

Keywords: Android, Brazilian Portuguese, Speech Recognition.

Összefoglaló

A cikk egy olyan felhő alapú beszédfelismerő szolgáltatást ír le, amely a szerver módban futó Julius dekóderen alapul. A rendszert a brazil portugál nyelv felismerésére állították be. A nyelvi támogatást a szerzők fejlesztették ki a FalaBrasil kutatócsoport eszközeit használva, amelyek ingyenesen elérhetők a csoport weboldalán. A FalaBrasil cloud-ot használva a Julius online és elosztott beszédfelismerést nyújt az interneten keresztül. A kliens Android 2.2 platformra készült. Az alkalmazás képes hang felvételére és küldésére, valamint detektálja a felhasználó beszédjének végét és figyeli a dekóder eredményét. Teszteléskénti a rendszert a Google *SpeechRecognizer* API-jával hasonlították össze hatékonyság, felismerési idő és pontosság alapján.

Kulcsszavak: android, brazílial portugál, beszédfelismerő.

1. Introduction

The interaction between man and computers has grown a lot. Nowadays, keyboards and mouse are the dominants controllers in the world of desktops. On the other hand, the touch screen is the most used technology in the smartphones. The Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) [1] technologies appear in this scenario as new forms of manipulating such equipment.

Many companies have invested in ASR. IBM created its own desktop dictation system, the IBM Via Voice [2]; Nuance developed its own as well, called Dragon Naturally Speaking [3] and a personal assistant called DragonGo!, similar to Siri developed by Apple. Google have the Voice Search, and provides the SpeechRecognizer API for Android developers [4].

Most applications is requires ASR nowadays have support for Brazilian Portuguese (BP). However, there is a lack of free resources for BP.

This work presents a distributed ASR system in which Julius [5] decoder operates in server mode and the client is an android application. This cloud ASR system is called **LaPS CSR**.

2. LaPS CSR

LaPS CSR is a client-server system in which the Julius decoder was set up on the FalaBrasil cloud to offer a distributed speech recognition via internet. To be able to recognize BP, Julius needs an acoustic model and a context-free grammar. These tools were developed using the FalaBrasil group support, from the Signal Processing Laboratory (LaPS) [6].

The LaPS CSR client is an android 2.2 application. Figure 1 explains the system operation: when the start button is pressed, the communication with Julius on the server is requested. Then, the user starts to speak and the system can detect when the user stops to speak. At this point, most of the audio is in the server side since the samples are sent by streaming.

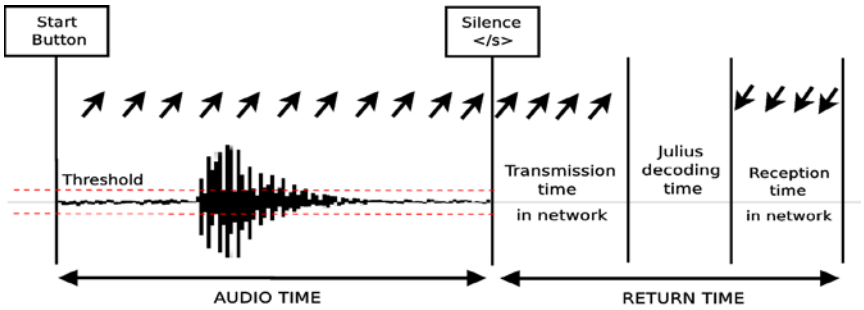

Fig.1 *System Operation*

On the server side, the decoding process begins while the client waits for the result. When the sentence is recognized, it is sent back to the client and shown in the layout.

In general terms, Google system works the same way.

Figure 2 shows the app schematic and its interaction with the FalaBrasil cloud. When the button is triggered on the application, the audio/connection manager tries to communicate with the server via socket. If the communication is established, the microphone stream is open along with three threads: *Audio Sender*, *Silence Detector* and *Sentence Receiver*.

Audio Sender thread receives microphone sample buffers on a list, simulating a FIFO (First In, First Out). The samples are then being sent to the server through the socket's output stream until the user stops talking.

Silence Detector thread simulates a LIFO (Last In, First Out), in which only the last sample buffer to arrive is analysed. When the silence is detected, the microphone stream is closed and thereafter the samples stop being sent to the server.

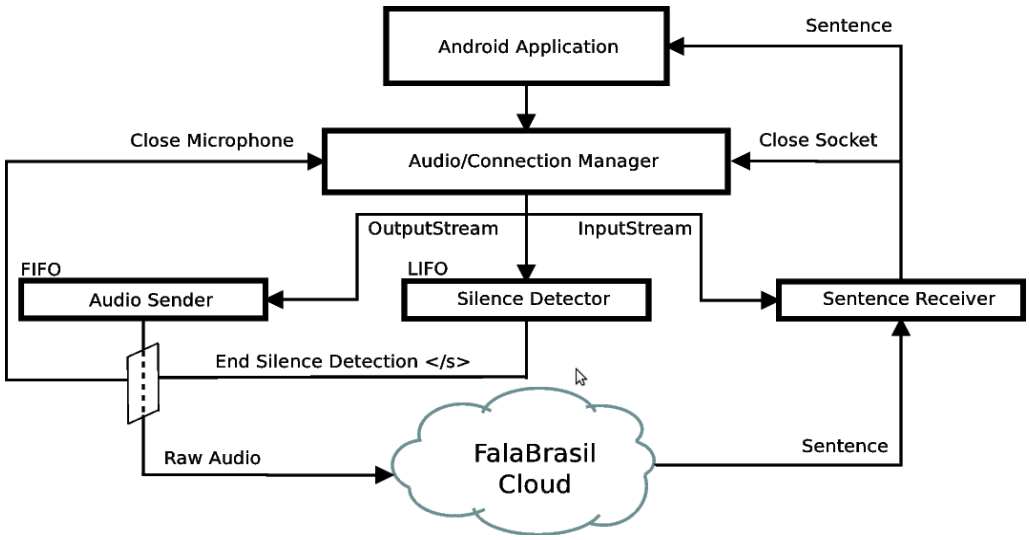

Fig.2. *Android application schematic.*

Sentence Receiver thread listens to the result string from Julius (sentence). When the result is available to the client, the socket is closed on the connection manager and the sentence is shown to the user on the application.

3. Test Environment

The grammar, which contains 235 points of a Brazilian city called Belém, was sampled with N_5=50 random sentences. Each sentence was recorded in a WAV file with 8 kHz sample rate and played to both systems by the same sound box, under the same non controlled acoustic environment. Time and accuracy of the systems were evaluated and compared.

The *average real time factor* $(\overline{xRT})$, given by Equation 1, was used as time measurement:

$$\overline{xRT} = \frac{1}{N_s} \times \sum \frac{return\ time_i}{audio\ time_i},\ N_s = 50 \qquad (1)$$

This metric was used to provide a fair comparison between the systems, given that some data from Google is “hidden” such as decoding, audio transmission and sentence return time. To evaluate accuracy rate, the word error rate (WER) was calculated according to Equation 2:

$$WER = \frac{D+S+I}{N_p} \qquad (2)$$

where D, S and I are, respectively, number of deleted, substituted and inserted words in a sentence of N_p words. The sentence error rate (SER) was also used as accuracy measurement, indicating the percentage of sentence errors in $N_s = 50$ sentences.

4. Results and Conclusion

It can be stated that the presented system is efficient when compared to Google system. Figure 3 shows that Google (blue histogram) takes on average 0.7 seconds to return a sentence while Julius in LaPS CSR (red histogram) takes on average 0.95 seconds.

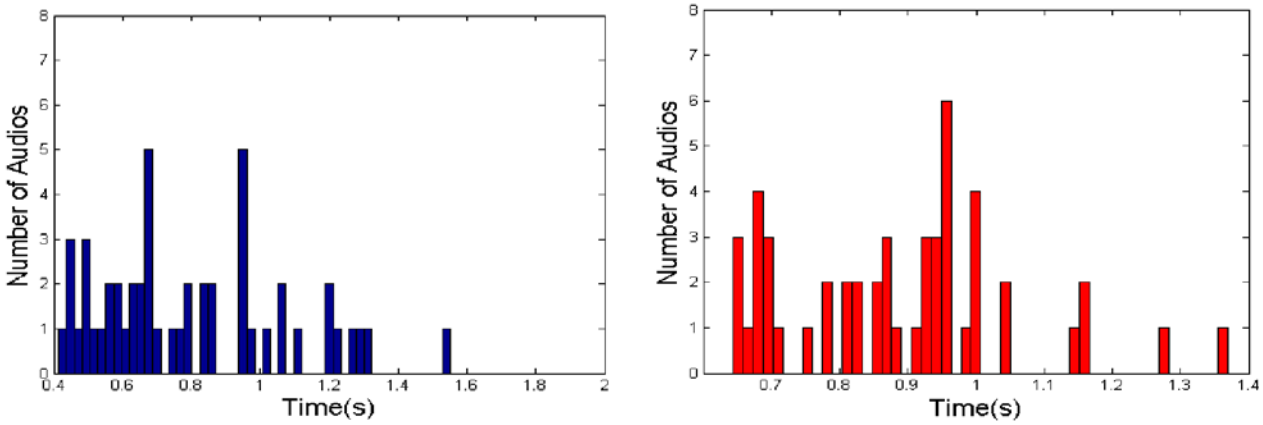

Fig.3. *Return time values for Google (blue) and LaPS CSR (red).*

Table 1. shows accuracy rate and time comparisons for the systems. Google had a better performance in time, as show in $\overline{xRT}$ values. On the other hand, WER and SER rates were lower for Julius. As Julius uses a personalized grammar, phonetic mistakes (noted in Google) can be avoided, such as "Arsenal" recognized as "Arcenal" for example.

Table 1. *Time and accuracy comparisons between the systems.*

System	$\overline{xRT}$	WER (%)	SER (%)
LaPS CSR	0.2279	10.16	16
Google	0.2063	13.33	24

Both systems had a good performance. Google was a little faster than LaPS CSR. However, LaPS CSR was better in terms of accuracy.

For the proposed application, which uses places as sentences, free-context grammars are an advantage. Google does not support personalized grammars and operates in a general dictation mode. Moreover, the API is not a free speech recognition solution.

Thus, it has been proven that is possible to implement a reasonable application-tailored distributed speech recognition system – comparing to the one offered by Google – using only free development tools.

References

[1] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing*, Prentice Hall, 2001.
[2]"*Embedded ViaVoice*," Visited in September, 2009, http://www.ibm.com/software/speech/.
[3] "*Dragon NaturallySpeaking*," Visited in February, 2014, http://www.nuance.com/dragon.
[4] "*Android Developers*," http://developer.android.com/.
[5] Akinobu Lee, Tatsuya Kawahara, and Kiyoshiro Shikano, *Julius - an open source real-time large vocabulary recognition engine*, Proc. European Conference on Speech Communication and Technology, p. 1691–1694, 2001.
[6] "*FalaBrasil Project*", Visited in December, 2013, http://www.laps.ufpa.br/falabrasil/