Speech Recognition and Text-to-speech Solution for Vernacular Languages

Free software and community involvement to develop voice services

James K. Tamgno
ESMT Dakar, Sénégal.
james.tamgno@esmt.sn

Claude Lishou
UCAD Dakar, Sénégal
claude.lishou@ucad.edu.sn

Aristide T. Mendo'o
ESMT Dakar, Sénégal
mendoo.aristide@gmail.com

Morgan Richomme
Orange Labs Lanion, France
morgan.richomme@orange-ftgroup.com

Seraphin D. Oyono Obono
DUT Durban, South Africa
eyonoobonosd@dut.ac.za

Pascal U. Elingui
ESMT Dakar, Sénégal
elinguiuriel@gmail.com

***Abstract* — This paper summarizes the work performed to study and develop a model Automatic Speech Recognition (ASR) system and a speech synthesis or Text-To-Speech (TTS) system on keywords of the vernacular language Wolof, respectively based on the open source software toolkits Julius and Festival. Much research has been developed in this area. Our goal is to be the first to develop a model for speech recognition and synthesis in Wolof, and also to create different lexicons and knowledge bases of phonetic, acoustic and lingistic feartures in order to introduce other languages.**

Keywords – Speech Recognition; Speech Synthesis; Wolof.

I. INTRODUCTION

Speech technologies – such as automatic speech recognition (ASR) and text-to-speech (TTS) systems – can play a significant role in bridging the "Digital Divide", which is currently preventing the vast majority of developing-world citizens from participating in the Information Age [1]. Most importantly, these technologies can lower the level of sophistication required to access information services, and thereby contribute towards the establishment of a fully inclusive information society. By circumventing language barriers and lessening the impact of illiteracy or disability, these technologies address real needs. Also, given the central role of language in cultural matters, speech technologies can play a significant role in guiding a diverse set of cultures towards the use of Information Technology.

The availability of these technologies creates new opportunities, but in order to realize the full potential of mobile ICT services, important challenges and obstacles must be overcome [2]. Most importantly, speech technologies need to be tailored to the properties of each new language in which they are to be used.

Thus, conscious of the dominant status of the Wolof language in Senegal, and in order to address services in local languages for the customers of its African subsidiary companies, France Telecom through Orange Labs signed a teaching partnership contract with the ESMT. Through this partnership, a research project was initiated, relating to the «Study and integration of a free software based speech recognition and text-to-speech solution for vernacular languages, within the framework of a simple grammar».

The current paper describes how this project was completed; it is organized as follows: Section II reviews some relevant theory and describes the methodology and technologies we used to analyze and to create new speech recognition in Wolof based on the Julius toolkit. Section III presents some theory on speech synthesis and describes the methodology and technologies we used to analyze and to create new speech synthesis in Wolof using Festival, within the framework of a simple grammar. We also describe how the development and the deployment were made. Section IV concludes the paper and outlines our future work.

A. Motivations

The United Nations Conference on Trade and Development (UNCTAD) Information Economy Report 2009 presents Africa as the fastest growing mobile market in the world (UNCTAD 2010). The number of mobile subscribers worldwide are expected to grow to 5.5 billion by the end of 2013 and 70% of them will be in developing countries. Mobile phones are contributing to unprecedented social and economical development on the continent with pioneering initiatives led by international agencies, Non-Governmental Organizations (NGO) and the private sector in agriculture, health, education, banking, citizen media, disaster and humanitarian relief. There is a widespread agreement that Information and Communications Technology (ICT) services, especially mobile ones, have the potential to play a major role in furthering social and rural development in developing economies such as Africa.

B. Vernacular Language Specificities in Senegal

According to National African Language Resource Center (NALRC), the local African language Wolof is understood and spoken by about 90% of Senegalese, while the official language is French. Wolof is a typical African oral language. It means that no formal grammar has been defined. Very few dictionaries [17] have been produced and pronunciation of the same grapheme may be very different thus a standard linguistic approach cannot be used here. Moreover, the

 ISBN: 978-1-61208-127-4

vocabulary is poor and lots of words are directly imported from other languages such as English, French or Arabic.
Illiteracy is counted among the problems that limit the development and progress of ICT in some undeveloped countries. By not writing, reading or understanding certain languages may hinder access to ICTs by some African populations who only communicate in their local language.
With an aim of promoting the African languages and dialects, various initiatives get busy to return the contents and the accessible software in African languages. By not using this language in the development of mobile applications and ICT simply denies access to hundreds of thousands of people to certain telecommunications services and jobs [1].

The weight of vernacular languages in African societies is very important; some countries deal with hundreds of local languages. One of the main problem with these languages or dialects is that, most of the time, there are no written languages (no formal grammar, limited number of dictionaries, few linguists) and have to deal with lots of import from other languages (French, English, Dutch, Portuguese,..). Therefore the procedures for speech recognition and text synthesis have to be adapted and the engineering work must be completed with a complex linguistic work. Additionally industrial actors of the sector have the technology to provide speech recognition and synthesis in any language, but this is not economically profitable. The cost of this technology is too high regarding the potential revenues that are too low. The choice of free software as sustainable and open technology was a possible answer to take into account the specificities of vernacular languages.

C. Approach

In this study, according to the architecture (Figure 10), core issue for investigation may include:

1. Collecting information on theoretical models of platform ASR/TTS and system providing similar services.
2. Analysis of various aspects of the modeling language :
   - Captures words (feature analysis)
   - Converts digital signal of voice into phonemes
   - Attempts to Recognize grapheme/phoneme
   - Finds match in acoustic model database
3. Attempts to make sense of what we are saying
4. Find a Typical architecture of a conversational agent

II. SPEECH RECOGNITION CONCEPTION

An utterance leading to interact with machines is the main idea behind speech recognition. Whatever language one may speaks, it has been progressively made possible.
Looking the way to do so, researches have felt on recognition systems using Hidden Markov Model (HMM). The approach based on HMM represents the current art of the state in open-source field. A new approach has emerged to deal with the challenging problem of conversational speech recognition [3][4].

This part concerning Speech recognition describes theoretical aspect, presents new languages creation methodology, and leads us to a Julius ASR development for Wolof.

A. Review of theory

The matter of the problem is to get one's utterance transcripts into text format.
The goal can be for a given acoustic observation $X=X_1,X_2,\ldots,X_n$, find a the corresponding sequence of words $\hat{W}=w_1,w_2,\ldots,w_m$, with de maximum a posteriori probability P(W/X) . Using Bayes decision rules, this can be expressed by [5]:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\ P(X/Y)P(W) \qquad (1)$$

Since the acoustic observation X is fixed, (1) is equal to
P(X/W) is the probability to observing acoustic observation X given a specific word sequence W. P(W/X) is determined by an acoustic model.
P(W) is the probability of observing W independent of the acoustic observation. It is referred to as a language model.

B. Methodology to create a new language recognized by Julius

B.1. Inside Julius: System Architecture

The language model consists of a word pronunciation dictionary and a syntactic constraint. Various types of language model are supported: word N-gram model (with arbitrary N), rule-based grammars and a simple word list for isolated word recognition.

$$P(w_1/w_2^N) = \frac{P(w_1, w_2, \ldots w_N)}{P(w_2, \ldots w_N)} = \frac{\prod_{i=1}^{N} P(w_i/w_1^{i-1})}{\prod_{i=2}^{N} P(w_i/w_2^{i-1})}$$

N- gram Bayes rule

Given a language and an acoustic model, Julius performs speech recognition task, whether embed in applications or in server-client architecture.
An overview of Julius base system is presented in Figure 1.
Acoustic models should be HMM defined for sub-word units.

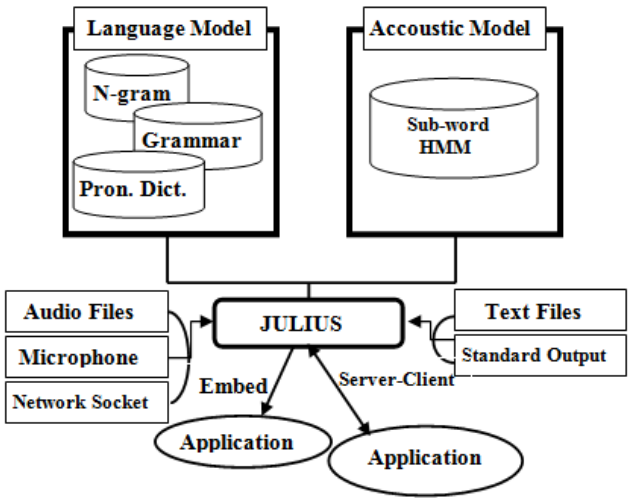

Figure 1. Julius overview [7].

The structure of Julius is illustrated in Figure 2. The top-level structure is the "engine instance"; it contains all the modules required for a recognition system: audio input, voice detection, feature extraction, Language Model (LM), Acoustic Model (AM) and search process.

An "AM process instance", holds an acoustic HMM and work area for acoustic likelihood computation. The "MFCC (Mel-Frequency Cepstral Coefficients) instance" is generated from the AM process instance to extract a feature vector sequence from speech waveform input. The "LM process instance" holds a language model and work area for the computation of the linguistic likelihoods. The "Recognition process instance" is the main recognition process, using the AM process instance and the LM process instance [17].

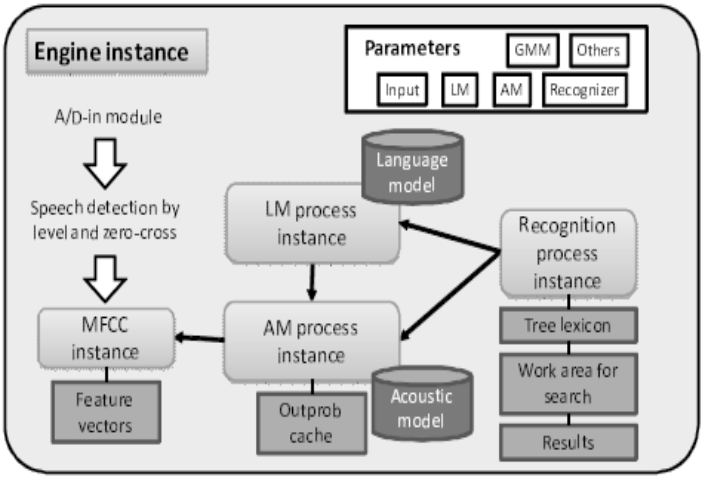

Figure 2. Internal structure of Julius [7].

B.2 Creating a new language

The language model in this context is rule-based grammar. This type of language model is defined in Backus-Naur Form (*BNF*) and compiled with "makdfa.pl", a Perl Julius tool [9]. Since Julius performs rule-based grammar building, we use the HTK toolkit [10] to create a new language in three main steps, as illustrated in Figure 3.:

- Data preparation: All needed data such as lexicon, audio files are collected and compile.
- Training: HMM parameters (mean, variance, etc.) are estimated using HTK tool suite.
- Evaluation: testing and analyzing acoustic model performances.

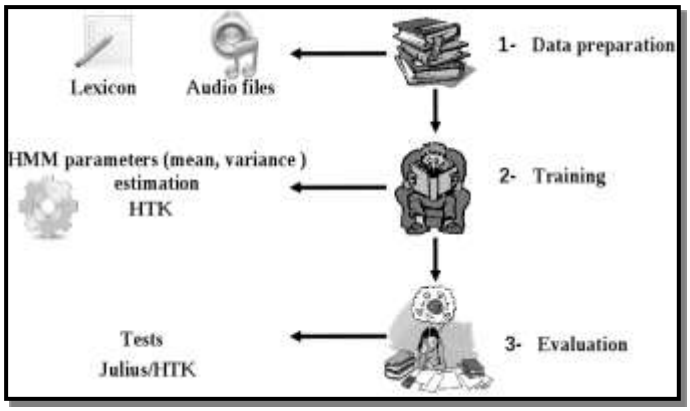

Figure 3. Steps for a new language creation with HTK [10].

C. Developments

New language creation following the three phases is performs by acoustic model maker (Make_AM.sh), which is a program build around Perl and Bash script, running HTK commands. It provides options for data compilation, acoustic model training and evaluation.

Make_AM.sh helps us to create an acoustic model in a chosen language (illustrated by Figure 4).

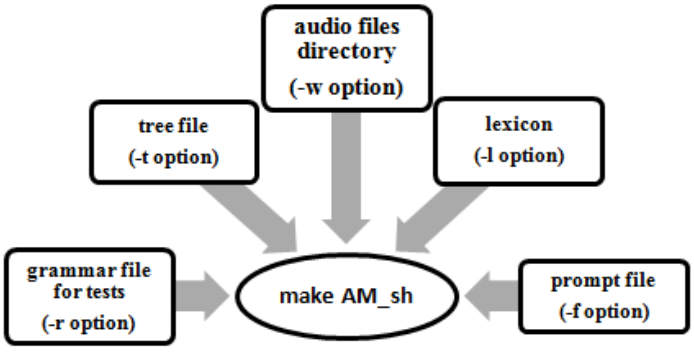

Figure 4. Running Make_AM.sh

Make_AM.sh architecture running on Linux operating system is presented in Figure 5.

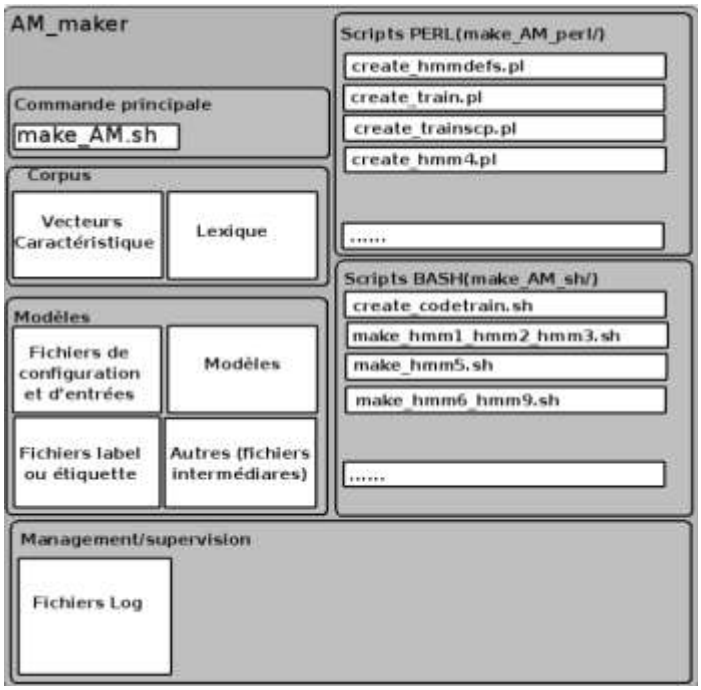

Figure 5. Make_AM.sh architecture

Algorithm: Make_AM.sh.sh

Input: dictionnary ,speech files,
prompt file, tree file, grammar file
Output: acoustic model files.

```
1.  if  required input then
2.      prononciation dictionnary
3.      transcription file
4.      MFCC extraction
5.      flat HMM
6.      silence model
7.      switch model (-p option)  do
8.          case monophone:
9.                    monophone  model
10.         case triphone :
11.                   triphone model
12.     end switch
13.     if  adaptation (-a option) then
14.         acoustic model indepedent of speaker
15.     end if
16.     if  test with HTK(-T option) then test & analyze
19.         else
20.         if  live test with HTK(-H option) then
21.             test with HTK
22.             else
23.             if live test with Julius (-J option) then
24.               test with Julius
25.             end if
26.          end if
27.      end if
28.      return  acoustic model
29.      else   return exception
31.   end if
32. end
```

C.1 Data preparation

For best results, the corpus of acoustic data used for learning (Wolof language) must be performed in a good quality recording studio. The records were made by several Indigenous from Senegal and other regions, for the model to capture most of the nuances of phones [17], to these records, we use Handy Recorder with four channels (H4n).

The age of the speakers selected depends on the intended target of the service. The age of the speakers was between 18 and 55. To proceed we use Make_AM.sh (-f otpion).

C.2 Training

Make_AM.sh passes through all training steps specified in HTK manual (htkbook). Before that, it compiles data in previous phase to prepare feature vectors MFCC and needed files.

C.3 Evaluation and results

We perform Make_AM.sh (-T/-J option) to evaluate the acoustic model built with HTK recognition tool.
We did Wolof recordings of about 123 sentences.

Parameter	**Description**
Testing audio files	Reusing training audio files
SOURCERATE	parameter not specified configuration files

Table 1. Test environment parameters

Evaluation provides following results:

Recordings carried out at the frequency of	**Percentage of sentences recognized**	**Ajusting SOURCERATE parameter**
16 KHz	100%	Yes/No
48 KHz	88,57%	No
48 KHz	91,43%	Yes

Table 2. Test results

Because Speech Recognition Engines need Acoustic Models trained with speech audio that has the same sampling rate and bits per sample as the speech it will recognize, several reasons can justify the results obtained in Table 2. The different speech mediums have limitations that affect speech recognition, telephony bandwidth limitations, desktop sound card and processor limitations, some VOIP PBX's, such as asterisk, actually represent audio data internally at 8kHz/16-bits sampling rates. Speech Recognition Engines work best with Acoustic Models trained with audio recorded at higher sampling rate and bits per sample. So to made adjustements we decided to collect speech recorded at the highest sampling rate your audio card support, and then downsample the 48kHz/16-bit audio to 16kHz/16-bit audio,

 ISBN: 978-1-61208-127-4

that can be supported by the speech medium as indicated by VoxForge [9], and create Acoustic Models from this. This approach permits us to be backward compatible with older Sound Cards that may not support the higher sampling rates/bits per sample, and also permit us to look to the future so that any submitted audio at higher sampling rates/bits per sample will be usable down the road when Sound Cards that support higher sampling rates/bits per sample will become more common, and processing power increases.

III. TEXT-TO-SPEECH CONCEPTION

A. Reminders on theory

One goal of TTS is to be able to provide textual information to people via voice messages. TTS provides voice output for all types of information that can be stored in databases and information services. Most speech synthesizers are built with modules that perform the 5 steps [15][16] shown in Figure 1.

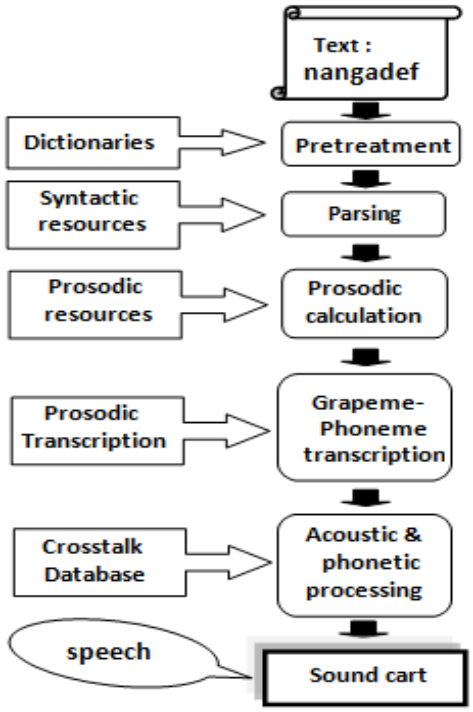

Figure 6. Five steps of Speech Synthesis

B. Methodology to create a new voice in festival

When creating a new voice in Festival [13][14], two scenarios are possible. The language to which the voice will be created is already supported by Festival, or that language is not yet supported. Our study was based on the latter, which requires the providing of:

- Phone set list
- Token processing rules (number etc)
- Prosodic phrasing method
- Word pronunciation (lexicon and/or letter to sound rules)
- Intonation (accents and FO contour)
- Durations
- Waveform synthesizer

But if the language is already supported by festival, we need to consider:

- Waveform synthesis
- Speaker specific intonation
- Speaker specific duration

Another possible solution for building a new voice in festival is to do voice conversion, as is done at Oregon Graduate Institute (OGI) and elsewhere.

For the construction of new voice in Wolof, we did not completely follow the steps described in the tutorial of the festival [17]. Instead, we have, due to a problem matching accented characters (UTF-8) of the Wolof alphabet, bypassed all the steps before the grapheme-phoneme transcription by an additional module written in Perl, composed of different lexicons and Perl functions which do all the steps from preprocessing to grapheme-phoneme transcription. This module produces output correspondence in ARPABET to a Wolof text input.

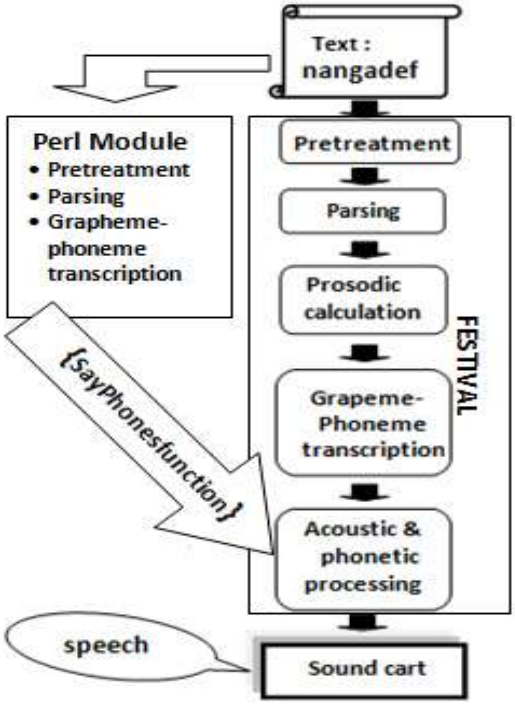

Figure 7. Grapheme-phoneme transcription

The connection with festival takes place using the *festival_client* command, with as parameter the text transcript in ARPABET alphabet by the Perl module. *festival_client* executes the *SayPhones* command on festival server that can synthesize data in ARPABET alphabet, conforming to the phone set list of the chosen language, namely English for our case.

 ISBN: 978-1-61208-127-4

Algorithm: transcription

Input:word list to transcribe=LAenrg()
List of Wolof characters=LCW()
Hash tables, matching graphemes Wolof
<--> graphemes ARPABET
Hash tables special cases, matching graphemes Wolof
<-->graphemes ARPABET
fonction épurer() /*correct or delete all non orthographic elements and no Wolof's character of the word which is crossed to him in parameter */
fonction transcrire() /*give the equivalent ARAPBET of Wolof word which is it given in paramêtre also by taking into account cases individual of transcption linked to the position of phonemes in words */
Output: transcription of the text in ARPABET

1. **Begin**:
2. valeurRetour = "";
3. Silence = "pau";
4. **for** all word of LAenrg() $mot1 € LAenrg()
5. $mot2 = epurer($mot1)
6. $mot3 = transcrire ($mot2)
7. valeurRetour = valeurRetour . Silence /* add a silence at the end of the valeurRetour */
8. valeurRetour = valeurRetour . $mot3 /* add a $mot3 at the end of the valeuRetour */
9. end for
10. **end**

The advantage here is that although working in a language not yet supported by festival, we could with the available diaphone database create a voice in Wolof (or other vernacular language), without having to make recordings.
It can thus build several voices without making great effort.

The diaphone database is extracted from audio recordings of a real person voice. It is important to choose a speaker with a distinctive voice for a better synthesized voice quality.

The recordings are made in the tutorial [17] with the stack of software *speech_tools* of CSTR; but it is completely in command line, then more advanced utilities such as *Audacity* [16] width a Graphic User Interface (GUI) recorder, make records and treatments faster and easier.

C. Building a new voice in vernacular language:

Vernacular languages (Wolof, in this case) are often very poor in vocabulary [17], and are often not formalized. These languages fill their vocabulary gaps by borrowing words in others more advanced languages. This makes it necessary to take into account all the languages of borrowing during the construction of the synthesizer. The construction of vernacular general synthesizer thus requires an abundant lexicon grouping non orthographic characters, exceptions, acronyms and words borrowed from others advanced languages. Figure 8 illustrates the applications talking-Web width festival in a Web page [18].

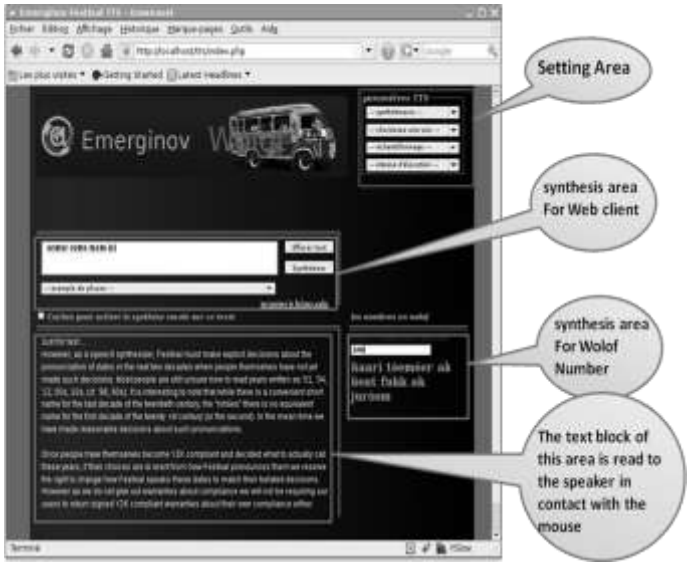

Figure 8. Talking web based on festival [18].

D. The talking web based on festival

The high level of illiteracy is the factor that leads indigenous to mystify Information Technology and Communication (ICT). We took advantage of open source software festival, to build a JavaScript library, named ajaxForFestival. This library based on Asynchronous JavaScript and XML (Ajax) allows to easily implement Web-talking at any website, this by a set of 12 functions.

Thus, we have described the approach that we used to build a voice based on an African vernacular language (Wolof) on the open source Festival TTS. We also describe the Web service built from speaking of TTS and ajaxForFestival JavaScript library that we built. This helps make Web content accessible to illiterate populations.

E. Web API to build lexicons database

Figure 9. describes the open API allows web users to create lexicons and their transcription in any language of their chose. This helps us build a database of lexicons in any other vernacular language [18].

 ISBN: 978-1-61208-127-4

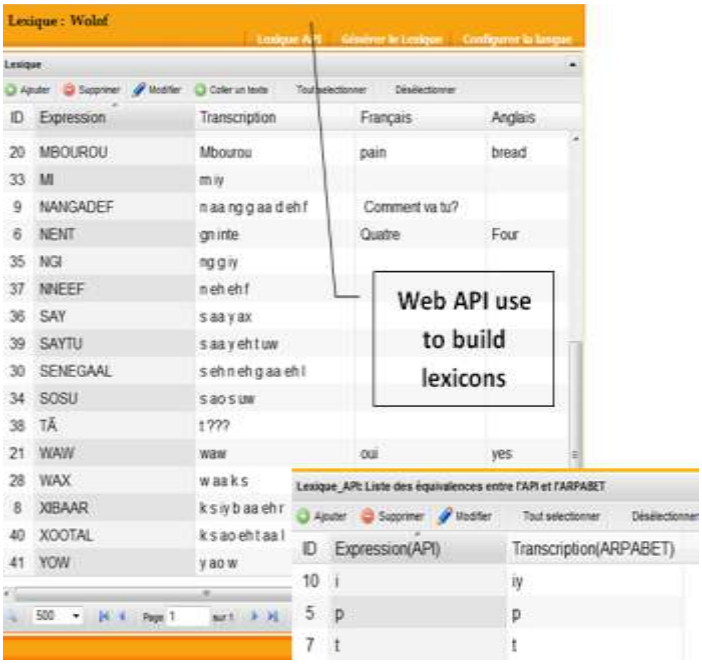

Figure 9. GUI to build lexicons database [18].

All the work done in this project is to ultimately get the architecture shown in Figure 10, where module TTS-ASR implemented, will provide interfaces for dialogue between clients, the administrator and open source tools Julius and Festival. The proposed interfaces should allow the creation of services based on speech recognition and speech synthesis. Customers can access services via various media (SIP, Web, Mobile phone...). The concept behind the services to implement is sending requests to the voice system, which responds with speech synthesis.

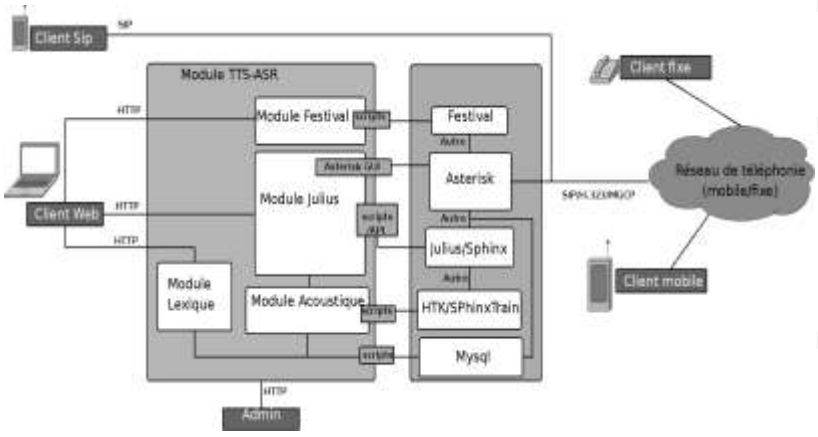

Figure 10. Final architecture

IV. CONCLUSION

The objective of our work was to conduct a theoretical study in the areas of recognition and speech synthesis in the Wolof language on the one hand and implementation of results obtained in the open source software for Julius Speech Recognition and festival for speech synthesis on the other. Today most of our results are already hosted by the platform Emerginov of France Télécom [18].

The filling of different lexicons and knowledge bases was the most time consuming task and requires some mastery of phonetic features, the acoustic and linguistic of Wolof, and a high degree of concentration. Recognizing this enormous challenge, we have grouped all these characteristics, then we have introduced various web APIs, finally we wrote scripts to automate the filling of the lexicons. This makes the self-evolving platform, because even in production environments lexicons can continue to be met.

Despite the problems we encountered, we are relieved that the end customer France Telecom (Orange Labs) appreciated the positive results we have obtained so far.

In our conceptual approach, we have as much as possible left the door open to other languages that may in future be incorporated into the platform without major problems. There are many possibilities for future development of the system. That is why we believe that the work produced will open doors to a large development in Senegal.

REFERENCES :

[1] P Nasfors. “Efficient voice information services for developing countries”. Master‘s thesis, Department of Information Technology, Uppsala University, May 2007.

[2] E. Barnard, L. Cloete and H. Patel. Language and technology literacy barriers to accessing government services. Lecture Notes in Computer Science. vol. 2739, pp. 37–42, June 2003.

[3] J. Bridle, L. Dengand and J. Picone, “An investigation of segmental hidden dynamic models of speech coarticulation for automatic speech recognition”. Final Report for the 1998 Workshop on Language. Engineering, Center for Language and Speech Processing at Johns Hopkins University, pp. 161-168, July 1998.

[4] J. Ma and L. Deng, “Target-directed mixture linear dynamic models for spontaneous speech recognition. IEEE transactions on speech and audio processing”, vol. 12, pp. 31-38, January 2004.

[5] M. A. Spaans, “On Developing Acoustic Models Using HTK”. Faculty of Electrical Engineering, Mathematics and Computer Science Delft University of Technology, pp. 13-20, December 2004.

[6] T. Kawahara, H. Nanjo, T. Shinozaki and S. Furui, “Benchmark Test for Speech Recognition Using the Corpus of Spontaneous Japanese”. ISCA & IEEE Workshop on Spontaneous Speech Processing and

 ISBN: 978-1-61208-127-4

Recognition (SSPR), Proc. SSPR2003, pp. 135–138, May 2003.

[7] "Julius architecture", http://julius.sourceforge.jp, last accessed on the 8th of March 2011.

[8] A. Lee and T. Kawahara, “Recent Development of Open-Source Speech Recognition Engine Julius”. Nagoya Institute of Technology & Kyoto University, In Proc. APSIPA ASC, pp.131-137, October 2009.

[9] “Collection of transcribed speech for use with Free and Open Source Speech Recognition Engines”, http://www.voxforge.org/, last accessed on the 8th of March 2011.

[10] “HTK website and resources”, http://htk.eng.cam.ac.uk/, last accessed on the 8th of March 2011.

[11] M. Morel and A. Lacheret-Dujour, "Synthèse vocale à partir du texte de la conception à la mise en œuvre" Laboratoire CRISCO, Université de Caen, vol. 42, pp. 193–221, September 2006.

[12] S. Baloul, "Développement d’un système automatique de synthèse de la parole à partir du texte arabe standard voyélé”. Université du Maine, May 2003.

[13] A. W. Black P. Taylor and R. Caley, “The Festival Speech Synthesis System”, Edition 1.4, for Festival 1.4.3, December 2002.

[14] A. W. Black and K. A. Lenzo, Festvox: “Building Synthetic Voice”, Voices version 2.1. http://www.festvox.org/bsv/, last accessed on the 8th of March 2011.

[15] “Festival website and resources”, http://festival.sourcearchive.com/documentation/1.96~beta/files.html, last accessed on the 8th of March 2011.

[16] “Audacity. The Free, Cross-Platform Sound Editor ”, http://audacity.sourceforge.net/, last accessed on the 8th of March 2011.

[17] J. Diouf "Dictionnaire Wolof-Français et Français-Wolof", Edition KARTHALA, August 2003.

[18] "Links to integrated platforms", http://projects.emerginov.org/esmt_Wolof/tts/, last accessed on the 8th of March 2011.

 ISBN: 978-1-61208-127-4