Expressive Speech Synthesis from Broadcast News

Joaquín Luzón Tuells

Submitted to the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya

In partial fulfilment
of the requirements for the degree in
AUDIOVISUAL ENGINEERING

Supervised by

Igor Jauk and Antonio Bonafonte,
Signal Theory and Communications Department

Degree Thesis

September 28, 2016

A Helena.

Revision history and approval record

Revision	Date
First document revision	19/09/16
Second document revisiion	28/09/19
Final document version	28/09/2016

Document Distribution List

Name	E-mail
Joaquín Luzón	luzontuells@gmail.com
Antonio Bonafonte	antonio.bonafonte@upc.edu
Igor Jauk	ij.artium@gmail.com

WRITTEN BY:	REVIEWED AND APPROVED BY
Date: 28/09/2016	Date: 28/09/2016
Name: Joaquín Luzón	Name: Antonio Bonafonte

Contents

Abstract

Speech Synthesis is the computer process of converting text to voice. This project consists in the synthesis of voices that can tell news with an appropriate expression, since it is important to achieve expressiveness on the generated speech in order to obtain natural sounding voices [1].

Conventional Speech Synthesis systems use as training data audios signals, specifically recorded for voice models training. Nevertheless, in this project the data was obtained from a news TV station, in order to test a different database in the speech synthesis.

An important part of the work done in this TFG has been preparing data later used in synthesis. The audio and its transcriptions were labeled so as to differentiate the expressions recorded: explaining good or bad news, or talking about relevant or trivial topics.

A phonetic segmentation of the database was obtained in order to create the models used in the speech synthesis. After preparing all the audio and transcriptions data, statistical-parametric models were estimated and used to synthesize test voices, in order to evaluate the previous setup work. All the project has been developed in a Linux environment, using Ogmios, AHOCoder and HTS-toolkit as main software.

The results obtained after synthesizing the voices shows that the data preparation process is correct, but the voices synthesized had not the enough quality. This is due to the adaptation of the voices towards heterogeneous samples, originated by the amount of different speakers used to train the models.

Resum

La síntesi de veu es el procés informàtic que transforma text a veu. Aquest projecte consisteix en la sínteis de veus que poden explicar notícies amb una expressió adient, ja que és important obtenir expressivitat en la parla generada per tal d'obtenir veus amb naturalitat expressiva [1].

Els sistemes de síntesis de la parla convencionals utilitzen com a dades d'entrenament veus gravades expressament pel entrenament dels models. No obstant, en aquest projecte s'ha creat una base de dades a partir d'unes gravacions d'un canal de televisió especialitzat en notícies, ja que es volia provar a sintetizar veu amb una base de dades diferent.

Una part important del treball dut a terme en aquest TFG ha sigut preparar les dades despŕes utilitzades en l'entrenament. Les gravacions i les seves transcripcions van ser etiquetades amb la intenció de diferenciar les epxressions gravades: explicant males o bones notícies, o parlant de temes rellevants o trivials.

S'ha obtingut una segmentació de la base de dades per tal de crear els models utilitzats en la síntesi de la parla.

Una vegada preparat els audios i les seves transcripcions, es van estimar models estadístic-paramètrics i es van utilitzar per sintetizar les veu de prova, amb l'objectiu de evaluar el treball de preparació anterior. Tot el projecte s'ha realitzat en un entorn Linux, fent servir *Ogmios*, *AHOCoder* i HTS-toolkit com a software principal.

Els resultats obtinguts desprès de la síntesi mostren que la preparació de les dades es correcta, però les veus sintetitzades no teníen qualitat suficient. Això es deu a

l'adaptacio de les veus a partir d'una base de dades molt heterogènia, degut a la quantitat de parlants diferents contemplats en l'entrenament dels models.

Resumen

La síntesis de voz es el proceso informático mediante el cual se transforma texto a voz. Este proyecto consiste en la síntesis de voces que puedan explicar notícias con una expresión adecuada, ya que es importante obtener expresividad en el habla generada para poder generar voces con naturalidad expresiva [1].

Los sistemas de síntesis del habla convencionales utilizan como datos de entrenamiento voces grabadas expresamente para el entrenamiento de los modelos. No obstante, en este proyecto se ha creado una base de datos a partir de unas grabaciones de un canal de televisión especializado en noticias, ya que se queria probar la síntesis de voz con una base de datos diferente.

Una parte importante del trabajo llevado a cabo en este TFG ha sido la preparación de los datos utilizados en la grabación. Las grabaciones y sus transcripciones se etiquetaron con la intención de diferenciar las expresiones grabadas: explicando buenas o malas noticias, o hablando de temas relevantes o triviales.

Se ha obtenido una segmentación de la base de datos con tal de crear los modelos utilizados en la síntesis del habla.

Una vez preparados los audios y sus respectivas transcripciones, se estimaron los modelos estadístico-paramétricos y se utilizaron para sintetizar las voces de prueba, con el objetivo de evaluar el trabajo de preparación anterior. Todo el proyecto se ha realizado en un entorno Linux, utilizando *Ogmios*, *AHOCoder* y HTS-toolkit como software principal.

Los resultados obtenidos después de la síntesis muestran que la preparación de los

datos es correcta, pero las voces sintetizadas no tenian la calidad suficiente. Esto se debe a la adaptación de las voces a partir de una base de datos muy heterogénea, debido a la cantidad de hablantes diferentes contemplados en el entrenamiento de los modelos.

Acknowledgements

This research would not have been possible without the help and support of Igor Jauk, Antonio Bonafonte and Santiago Pascual, whom I appreciate the dedication and attention given during this months.

I also want to express my gratitude to Gonzalo and Fran for the support, advices and fellowship. Thank you Florian for your kind help with my English doubts.

Thanks to any colleague that offered encouragement and affection in this years of school, especially Alejandro, Marina, Marc and Jessica. Thanks to all the people involved in the past, present or future of *Revistes Campus Nord. Gràcies Distorsió!*

Of course, I owe my deepest gratitude to my parents, for their love and unconditional support.

List of Figures

List of Tables

1 Introduction

The aim of this project is to study expressive speech synthesis while producing voices with certain expressiveness, making them suitable to read text with an appropriate intention. The main approach of this study is not the intelligibility of the generated voices but the ability to distinguish between different expressive states when interpreting a text.

1.1 Motivation

Speech synthesis research has been focused on creating intelligible speech, setting aside the semantic information on the text transmitted. This situation has derived onto a state-of-the-art high quality speech synthesis in terms of intelligibility.

Adding emotions to an intelligible voice is the key to create natural-sounding speech [2], and this would heavily increase the chances to apply the use of synthesized voices in several areas, such as entertainment (*i.e.* audiobooks or videogames) or medicine (*i.e.* voice prosthesis).

However, since the corpus used as a database for the synthesis in this project is a set of audios from *3/24*, the news broadcasting channel from *Televisió de Catalunya (TVC)*, the goal of voices created is focused on reading distinct type of news. In this case, it is necessary to differentiate between *good news* or *bad news* and distinguish the subject of each one: *sports news, political news, social news, etc.*

1.2 Document Breakdown

In Section 2, State of the art, are explained the differences between the widespread used speech synthesis systems. Besides that, it is described the theory basis and the technology used during this project.

Data Analysis and Selection (Section 3) deepens in the data used to synthesize the resultant speech. First the speech corpus is described and later the steps followed from select suitable audio fragments and until preparing this fragments and them transcriptions so as to obtain the phonemes used on the further speech synthesis.

Section 4 describes the phonetic segmentation. This segmentation consists in indicate, for each recorded phone, its temporal position. This segmentation is fundamental so as to obtain the models that permit to synthesize speech. A first phase of format and lexical adjustments is necessary to finally compute the phones.

Once obtained,the speech and the phoneme segmentation are used in the synthesis training, defined in Section 5. The training has three main sequential stages: Speaker Independent Training, Speaker Adaptative MLLR Training and Speaker Adaptation MAP Training.

At the end of the document, are located Section 6, Results, and Section 7, Conclusions.

As an appendix, are exposed the Work Plan, Time Gantt Diagram and Milestones that were produced to manage this project.

2 State of the art of Speech Synthesis

Speech Synthesis is the computer process that generates voices from written information. *Text-To-Speech Synthesis* (TTS) has a widespread use in commercial systems. In this chapter it is explained the theory used to synthesize the speech. First of all, in section 2.1, a brief overview of speech synthesis systems gives way to a more elaborated explanation of the system used in this TFG. After that, in 2.2, it is explained the method used to transcribe the broadcast audios, later used to set up the corpus needed to carry out the speech synthesis.

2.1 Statistical-Parametric Speech Synthesis

There are several ways to synthesize this speech, such as: articulatory synthesis, based on models of the human vocal tract, simulating the movements of the speech articulators; formant synthesis, based on excitation models.

However, the most widely used method is concatenative unit selection, based on concatenating waveforms selected from large, single-speaker speech databases, in order to create a natural-sounding speech [3]. This speech synthesis techniques lacks on flexibility when it comes to give expressiveness to the generated voices. Another approach is statistical-parametric speech synthesis, that use mathematical models to represent the different sounds and generates speech based on these models. It is proven that they add extra flexibility [2]. It is the case of Hidden Markov Models based Speech Synthesis (HTS), the speech synthesis system used in this project.

This work is a follow up of the one presented in [2], whose main goals were: *(1) in-*

creasing the flexibility of expressive voice creation and (2) overcoming the limitations of speaking styles in expressive synthesis, as described in [2].

This speech synthesis research is based on statistical systems as they generally have more flexibility than concatenative systems [2]. This system has been developed in the framework of Hidden Markov Models based Speech synthesis (HTS). In HTS system, a set of speech parameters are modeled by using one Hidden Markov Model (HMM) for each context-dependent phoneme. Speech signal is generally parametrized in two main vector streams: fundamental frequency, f_0 , and spectral envelope. A better quality synthesis can be achieved by adding more information related to the source in an additional stream.

2.1.1 Hidden Markov Models

As seen in Figure 1, HMM is a finite state machine which generates a sequence of discrete time observations [4]. In case of HTS, a 5-state right-to-left model, with two options for each state and every time unit: to increase one state index or to stay. Notation to refer HMM is $\lambda = (\mathbf{A},\mathbf{B},\Pi)$; where $\mathbf{A}$ are the state transition probabilities, $\mathbf{B}$ the output probability distribution and Π the initial state probabilities.

The speech parameters are the observation sequence, $\mathbf{O} = \{o_0, o_1, ..., o_T\}$, whose distribution is modeled by a Gaussian Mixture Model (GMM). Additionally we can define the *hidden* state sequence, $\mathbf{q} = \{q_0, q_1, ...q_T\}$.

The parameters of HMM, are estimated using the Maximum Likelihood (ML) criterion: $argmax_\lambda$ P($\mathbf{O}|\lambda$). As there is no possibility to analytically find the optimization solution that maximizes P($\mathbf{O}|\lambda$), Baum-Welch iterative algorithm is executed until

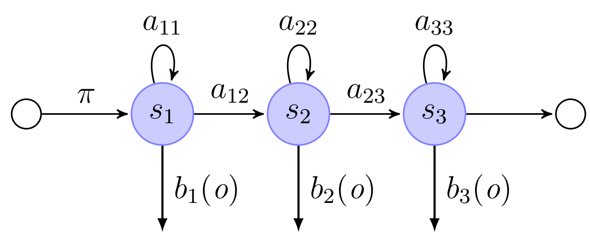

Figure 1: Tri-state HMM are used to model speech parameters

a maximum number of iterations or this algorithm converges into finding the best model λ given the the training data.

Once the model λ has been estimated, it can be used to synthesize the state sequence observations suitable to this model, which is used in the synthesis phase. Is also based on the ML criterion: $argmax_{\mathbf{O}}$ P($\mathbf{O}|\lambda$) [5].

2.1.2 HMM-Based Speech Synthesis

The block diagram of HTS system is showed in Figure 2. In the training part, the first step to parametrize speech signals is to estimate the spectral information. In HTS, this information is obtained by calculating the *Mel-Frequency Cepstrum Coefficients* (MFCC) [6] of every audio frame (usually each 5ms using a 20ms window length). MFCC are used because of the similitude of the the spectrum represented by this coefficients and frequency resolution of the human ear.

Furthermore, the excitation parameters have to be estimated. The f_0 is obtained by a pitch detection algorithm [7]. As f_0 is a variable dimensional parameter (due to its duality voiced/unvoiced regions), Multi-Space Distributions HMMs (MSD-HMM) are used in the modeling stage.

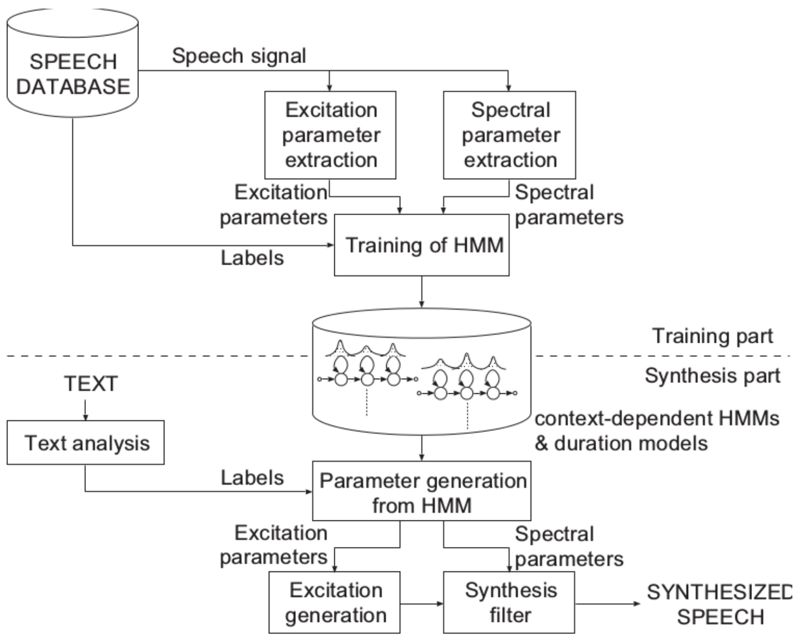

Figure 2: Overview of a typical HMM-based speech synthesis system (from [8])

After the parametrization, for each phoneme, a context-dependent HMM is estimated as seen in Section 2.1.1.

In the synthesis part, first the text is analyzed and there are selected the suitable associated models to each phoneme. The speech parameters are generated as mentioned in Section 2.1.1. Finally, speech is generated using an impulse excitation vocoder as follows: in voiced frames, the excitation is generated as an impulse train where the pulses are separated by the length of the pitch period, and in unvoiced frames, excitation is generated as white Gaussian noise. The mel-cepstral coefficients are generated by the MLSA filter [9].

In fact, in this project we used a high quality vocoder, AHOCoder [10], which is based on a Harmonic Plus Noise (HNM) generation model.

2.2 Audio Transcriptions

In previous sections we have seen that HMM are estimated from speech data, which needs to be transcribed so that each speech segment is used to train the appropriate model. This section describes the audio transcriptions available for the speech recordings.

Audio transcriptions were previously manually written by *TALP* research group. This transcriptions are made using the Transcriber program and saved in its own format (`.trs`). Every audio fragment is transcribed into a *Turn* structure. There is a detailed information about every *Turn*, with labels such as *Event* (if there was an unexpected noise or bad pronunciation), *Time* (of the beginning of the audio fragment, or the Event) or *Speaker*.

In Figure 3, a .trs file is opened with Transcriber software. Detailed information is decoded and presented in the GUI. Nevertheless, there are also present extra information labels in the text, about language *([lang=Spanish])* or prosody changes *([pause])*.

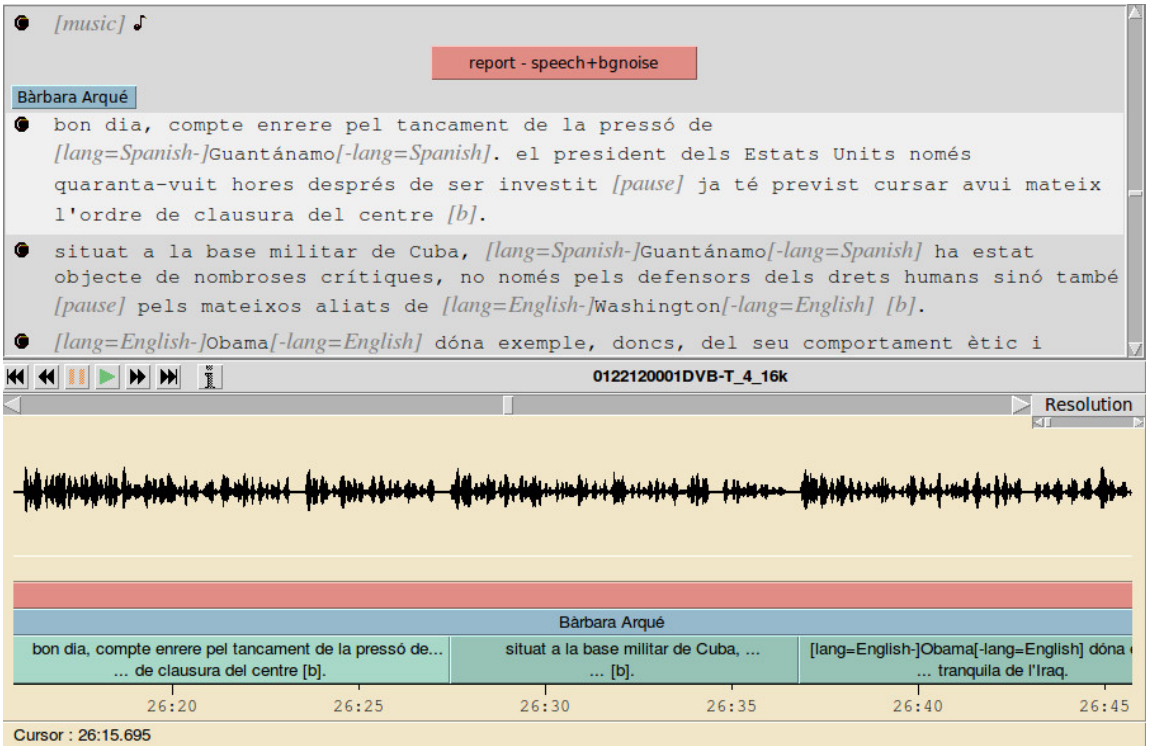

Figure 3: .trs file, opened with Transcriber graphic interface, and its respective audio waveform

After listening to the audio files and reading the transcriptions, new labels were manually added so as to categorize every useful audio fragment, given the expression of the speaker and what were the news about (for more detailed information about labeling process, see 3.2).

<Turn speaker="spk103" mode="planned" channel="studio" startTime="2269.323" endTime="2279.624" >
<Sync time="2269.323"/>el temporal deixa enrere set morts, quatre són els nens
<Event desc="noise" type="noise" extent="begin"/>
que van
<Event desc="noise" type="noise" extent="end"/>
morir ahir a San Boi <Event desc="b" type="noise" extent="instantaneous"/>
en un camp de beisbol. l'Ajuntament de la localitat <Event desc="b" type="noise" extent="instantaneous"/>
ha decretat tres dies de dol <Event desc="pause" type="noise" extent="instantaneous"/>
.
</Turn>

Figure 4: Example of a Turn in a .trs file as plain text, with *Event*, *Speaker* and time labels (*beginTime*, *endTime* and *Sync time*)

Figure 4 shows an original *Turn* structure and Figure 5 shows the modified transcription.

```
<Turn speaker="spk103" mode="planned" channel="studio" startTime="2269.323" endTime="2279.624" >
<Sync time="2269.323"/>el temporal deixa enrere set morts, quatre són els nens SELECT_“BadNews”
<Event desc="noise" type="noise" extent="begin"/>
que van <Event desc="noise" type="noise" extent="end"/>
morir ahir a San Boi <Event desc="b" type="noise" extent="instantaneous"/>
en un camp de beisbol. l’Ajuntament de la localitat <Event desc="b" type="noise" extent="instantaneous"/>
ha decretat tres dies de dol <Event desc="pause" type="noise" extent="instantaneous"/>
.
</Turn>
```

Figure 5: Example of a Turn, with new selection labels

3 Data Analysis and Selection

This chapter explains how was created the database subsequently used to build the speech synthesis system. Figure 6 represents the steps to obtain the final database. Audio and transcription files that set the speech corpus are detailed in 3.1. Also, in section 3.2 and 3.3, are described the labels used to categorize every relevant audio fragment and the criterion to determine which audios are relevant or not. Finally, 3.4 describes the procedures to prepare the selected audio segments and the required text information in order to perform a correct audio segmentation.

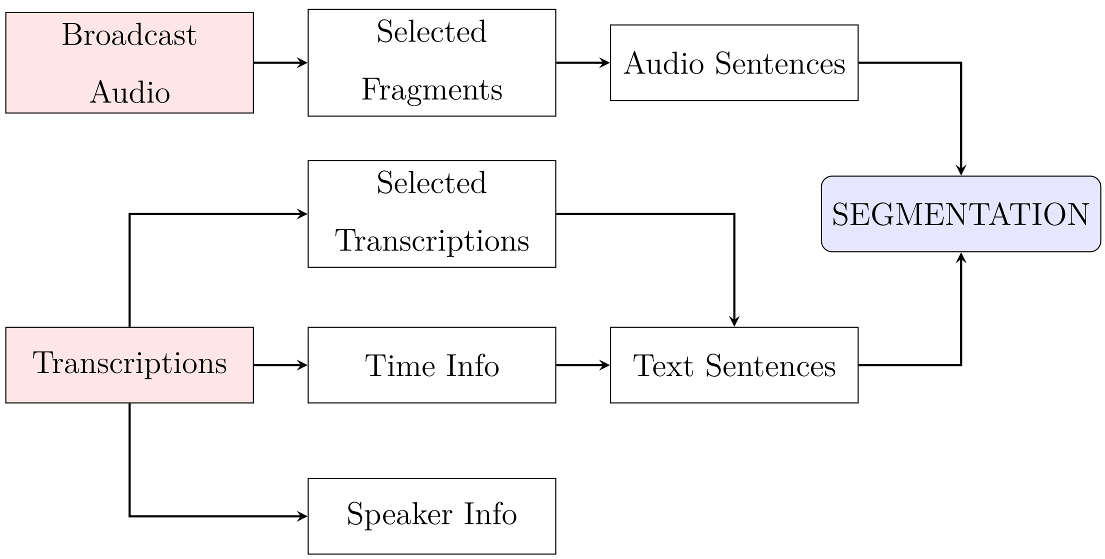

Figure 6: Data Analysis and Selection scheme

3.1 Corpus

The corpus consists of the audio signals and the transcriptions:

- WAVE audio files (.wav, 16kHz, 16 bits, PCM): 132 hours of multiple speakers. As the audio was obtained from the 24 hours news station of *TVC*, most of the sections transmitted were repeated broadcasts. The profitable audio was significantly reduced (as will be seen in 3.4). Besides, only noiseless audio was transcribed, making every noisy audio fragment non-profitable.
- Modified Transcriber files (.trs). Orthographic transcriptions of the selected audio including other labels.

3.2 Data Analysis

The selection of audio fragments (and its correspondent transcriptions) suitable for posterior segmentation was made manually, by listening the .wav files and reading .trs files content (see 2.2). Once an audio fragment was identified as adequate to create the synthesis database (non repeated, noiseless and with orthographic transcription), a label was added to the *Event* in the .trs file. Label categories were:

- `Bad News`. Where a presenter, or reporter, explains bad news, acquiring a very serious intonation.
- `Entertainment`. Where a presenter, or reporter, explains entertainment news. There are usually light and good news, explained with a distended intonation.
- `Good News`. Where a presenter, or reporter, explains good news, also explained with a distended intonation.
- `History`. Voice from a documentary, where the presenter explains information about historic events (specifically about the Cold War).

- `International`. Where a presenter, or reporter, explains international news, with neutral intonation.
- `Neutral`. Where a presenter, or reporter, explain news without any good or bad connotation, acquiring a neutral intonation.
- `Politic`. Some audio fragments contained interventions or statements made by catalan politicians.
- `Political`. Where a presenter, or reporter, explain political news, with neutral intonation.
- `Social`. Where a presenter, or reporter, explain news related to social events or gossips. The intonation is normally distended.
- `Sports`. Where a presenter, or reporter, explain news about sport events.

3.3 Audio Selection

After labeling every appropriate fragment, the audio was manually split into one audio file (`.wav`, 16kHz, 16 bits, PCM) for each fragment. From every transcription, there were extracted three features: the correspondent text to every audio fragment and the time and speaker information of every sentence in the fragment.

Every text extracted from the original `.trs` files had been converted from its original character encoding (UTF-8 or CP-1252, depending on the file) to ISO 8859-1, or ISO Latin-1, for further compatibilities with *Ramses* and *Ogmios* segmentation tools [11] [12].

3.4 Sentence Extraction

Once all audio fragments are obtained and their time, speaker and text information, the fragments are split into single-sentence audio files (`.wav`, 16kHz, 16 bits, PCM). The total length of useful audio files has ended up being slightly over 3 hours from over a hundred speakers.

In a parallel process, each text sentence corresponding to its respective audio sentence is stored in a `.txt` file (ISO Latin-1). After the data is formatted correctly, is time to use the corpus to obtain the phonemes further used on speech synthesis.

4 Audio Segmentation

In this section it is explained the process of phonetic segmentation. Once are prepared the audio sentences and its respective transcriptions, the next step is to execute the segmentation with *Ramses* and *Ogmios* tools.

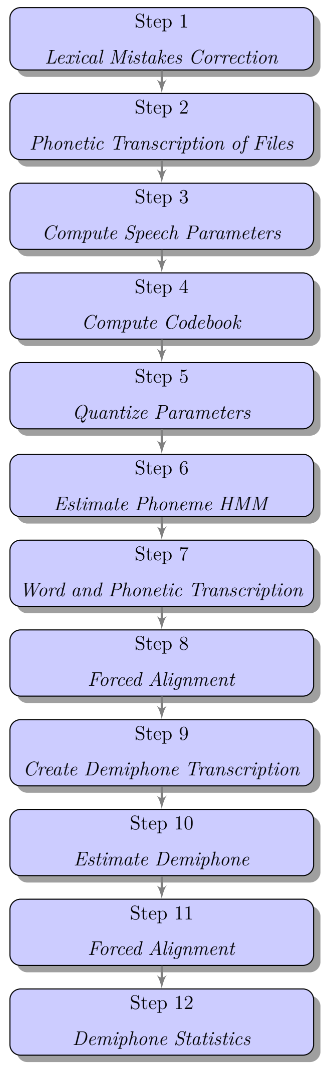

Figure 7: Audio Segmentation Scheme

4.1 Lexical Mistakes and Dictionary

Even though the segmentation is an automatic process, the data has to be supervised and modified in order to correct the transcription, format or linguistic mismatches between files and segmentation tools.

First of all, a revision of every word in the transcription files. Each word is compared with all the words contained in a lexicon. There were several words missing such as neologisms or people names. In this case, words had to be introduced manually in the dictionary. An extra revision of every text file is done in order to ensure that the format is correct.

4.2 HMM Demiphone estimation and Forced Alignment

Once the mistakes are corrected, the next step is to transcribe the text in phonemes. At the same time, using the audio files, speech parameters were computed. This parameters are used to compose a codebook which afterwards is quantized. At the time the parameters were correctly computed and quantized, a HMM estimation of every phoneme is done.

After a phonetic transcription of the words in the lexicon, the first forced alignment by means of Viterbi algorithm was done, finding the phone boundaries (transition between models) [13]. This first forced alignment not only finds the boundaries of phonemes but also detects the internal silences in the sentences.

Finally, it was create a demiphone transcription and HMM estimated. Second forced alignment is done in order to obtain the demiphone statistics necessary to start the speech synthesis training.

Once the corpus has been processed in the segmentation stage, next we train the voice models.

5 Speech Synthesis Training

In this chapter it is explained the process of Speech Synthesis, since the extraction of the needed information from the corpus, untill the generation of the models, using HTS software [8] , [18].

5.1 Feature extraction

The first step before starting the speech synthesis, is to analyze and extract the necessary features by using AHOcoder [10]: 40 Mel-Cepstrum coefficients (39 Mel-generaliced cepstral coefficients, MGC, and 1 distortion of band aperiodicities, BAP), $\log(f_0)$ and voiced frequency. Speech analysis conditions are 16kHz sampling frequency and a frame shift of 5ms, in a 20ms window. The limits for f_0 extraction were set between 40Hz and 500Hz.

5.2 Speaker Independent Modeling

This process consists in creating a voice model that does not depend on the speakers characteristics in the database. This model is also called average model.

In the first training phase, models are initialized. The (five-state, left-to-right) HMMs of isolated monophones are estimated by applying Viterbi, using HTK [17]. Models are reestimated more precisely using Baum-Welch (BW).

However, the models in this stage are created without considering the phone situation in a phrase. So as to create an more accurate version of the phones, HMMs are clustered by context. Again, a BW reestimation is done.

Next step is a HMM tree-based clustering process. The decision trees are different depending on the parameters: spectrum, f_0 and duration. This process leads to tied parameter structures that need to untie before reestimate, again using BW.

, duration models are also modeled, clustered, untied and BW reestimated. After all these steps, we obtain the necessary files to execute speech adaptative training synthesis with HTS.

A final step begins computing a Global Variance (GV) [16] of statics feature vectors with the aim of correct oversmoothing effect after other speech parameter generations.

Table 1 shows the explicit list of steps and commands used to train.

Code	Step
MKEMV	Preparing environments
HCMPV	Computing a global variance
IN_RE	Initialization and Reestimation
MMMMF	Making a monophone mmf
ERST0	Embedded reestimation (monophone)
MN2FL	Copying monophone mmf to fullcontext one
ERST1	Embedded reestimation (fullcontext)
CXCL1	Tree-based context clustering
ERST2	Embedded reestimation (clustered)
UNTIE	Untying the parameter sharing structure
ERST3	Embedded reestimation (untied)
CXCL2	Tree-based context clustering
ERST4	Embedded reestimation (re-clustered)
FALGN	Forced alignment for no-silent GV
MCDGV	Making global variance
CONV1	Converting mmfs of speaker independent voice of the hts_engine file format

Table 1: Speaker Independent synthesis steps

5.3 Speaker Adaptative Training (MLLR)

After creating the average model, there are created speaker dependent models, so as to improve the quality of the voice. This technique is based on clustering speech by means of decision tree constructions.

In [14], it is described a transformation $\mathbf{G}^{(r)}$ for each r speech clusters, estimated together with the optimum set of HMM parameters, in order to maximize the likelihood of the training data. It is again a ML problem, this time solved in the framework of Maximum Likelihood Linear Regression (MLLR) method [14] [15].

Table 2 shows specific steps for this training.

Code	Step
REGRT	Building regression-class trees for adaptation
SPKAT	Speaker adaptative training
CONV2	Converting mmfs of SAT voice to the hts_engine file format

Table 2: SAT synthesis steps

5.4 Adaptation Training (MAP)

Later, a Maximum A Posteriori (MAP) adaptation training is performed to compute a final estimation. This approach gives better estimation of model parameters than ML [19]. This estimation is done for each of the categories classified in the labels.

Adaptation stage consists in adapt the means of HMM models at a finer resolution, so as to improve the speaker-dependent model. As HMM output distribution is modeled as M-component Gaussian mixture model (where M is the number of observations), adaptation process transform these Gaussian and center them closer to the parameters of the speaker to adapt.

Table 3 shows the steps followed to execute the MAP training.

Code	Step
MKUN2	Making unseen models
ADPT2	Speaker adaptation
MAPE2	Additional MAP estimation
CONV3	Converting mmfs to the hts_engine file format

Table 3: Adaptation synthesis steps

After training the models, the final process is to synthesize the resultant voices.

6 Results

Once Speaker Independent, MLLR and MAP models are obtained, several voices are synthesized by using *AHOCoder*. The general model was obtained training 3 hours of audio, from more than a hundred speakers (see Section 3.4). Later, this model was adapted using four labeled audio databases, each one with different lenghts:

- `Good News`: 18 minutes of audio recordings.
- `Bad News`: 21 minutes of audio recordings.
- `Sports`: 27 minutes of audio recordings.
- `Neutral`: 42 minutes of audio recordings.

After synthesizing the voices, labels from the initial corpus were used to test the TTS systems. We achieved the synthesis of all voices. However, the expressiveness in this voices is not improved.

This lack of expressiveness in the results could happened due to the heterogeneity of the database used, forcing to average a pitch model from many speakers labeled in the same category. Furthermore, the speakers grouped under the same category were not been differentiated by gender, increasing the pitch rang that forced the averaging.

As a possible solutions for the later resarches, we suggest:

- To normalize f_0 for each gender: Masculine and Femenine.
- To divide the data base not only by labels but also by different speakers.

7 Conclusions and future development

In this TFG has been followed the entire process to create voices using Speech Synthesis systems. Nevertheless, the difference between the system used and the conventional Speech Synthesis systems lies in the database used to train the voice models. Unlike the classical approach in Speech Synthesis systems, where the voice database came from recording audios specifically recorded for the modeling, here are used audio recordings from television.

In order to use this data, the audio recordings were selected depending on their noise conditions. After the selection, the resultant audio signals were transcribed and classified according to the information contained in them. This information was categorized depending on the expressions heard in the broadcast news audios.

Categorization of the audio fragments was performed by labeling the transcriptions and later splitting every sentence in the audio files. With this sentences, a phonetic segmentation process was executed.

The data extracted has not the depth of the usual Speech Synthesis systems databases, that count with around 10 hours of audio recordings, but only counts with less than half an hour by most of the categories labeled. Due to this situation, Speech Synthesis training process has followed the adaptation of every category model over a general model.

Nevertheless, the speech synthesized was intelligible, but had neither expressiveness and natural-sounding, compared with other synthesized voices, from conventional text-to-speech systems.

The heterogeneity of the database had propitiate the difficulty to obtain a expressive pitch approach. As we used as training data audio recordings from very different speakers, without distinguishing between genders or audio contents, is very difficult to adapt the models by only using less than half an hour of audio data.

So as to improve the generation of the models, a solution could be to create several databases, distinguishing by genders, gender and labels or create a database from every speaker with relevant time amount of audio recordings. For creating this databases, more broadcast news audio recordings should be obtained and transcribed, in order to obtain more data for the training processes.

Regarding work planning on this project, the time required to obtain the corpus in a correct format had been much higher than predicted at the start of the project. The format that enables to continue with phonetic segmentation and speech training stages had been reached past half the time available to develop the full TFG, making impossible to deepen in the synthesis training.

A wide study on techniques and concepts has been done during the development of this project. I have deepen in HMM theory, specifically in their application in Speech Synthesis systems. My knowledge about Linux terminal and SSH network protocol has been highly improved. Also, I have earned experience in text-processing and HTK-toolkit usage.

As a conclusion for further development, the work done in this TFG enable future researchers to have a database ready to investigate better ways to synthesize speech with statistical-parametric speech synthesis systems.

References

[1] F. Burkhardt and N. Campbell, "Emotional Speech Synthesis", *The Oxford Handbook of Affective Computing, 2014.*

[2] I. Jauk, A. Bonafonte, P. López-Otero, and Laura Docio-Fernández, "Creating Expressive Synthetic Voices by Unsupervised Clustering of Audiobooks", in *ISCA Interspeech, Dresden, Germany, September 6 - 10, 2015.*

[3] A. J. Hunt and A. W. Black, "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database", *Proc. ICASSP-96, Atlanta, May 7-10, 1996*

[4] J. Yamagishi, "An Introduction to HMM-Based Speech Synthesis", *October 2006*

[5] K. Tokuda, T. Kobayashi, T. Masuko, S. Imai, "Mel-generalized cepstral analysis - a unified approach to speech spectral estimation", *Proc. Int. Conf. Spoken Lang. Proc., vol. 3, pp. 1043-1046, 1994.*

[6] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis", *Proc. of ICASSP, pp.1315-1318, June 2000.*

[7] I. Luengo, I. Saratxaga, E. Navas, I. Hernaez, J. Sanchez, I. Sainz, “Evaluation of pitch detection algorithms under real conditions”, *Proc. ICASSP, pp. 1057-1060, 2007*

[8] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based Speech Synthesis System (HTS) Version 2.0", in *6th ISCA Workshop on Speech Synthesis, Bonn, Germany, August 22-24, 2007.*

[9] T. Fukada, K. Tokuda, T. Kobayashzjtt and S. Imait, "An Adaptative Algorithm for Mel-Cepstral Analysis of Speech", *Proc. ICASSP-92, pages 137–140, March 1992.*

[10] D. Erro, I. Sainz, E. Navas, and I. Hernáez, "Improved HNM-based Vocoder for Statistical Synthesizers", *ISCA Interspeech, Florence, Italy, August 28 - 31, 2011.*

[11] A. Bonafonte, P .D. Agüero, J. Adell, J. Pérez, A. Moreno, "Ogmios: The UPC Text-to-Speech synthesis system for Spoken Translation", *TC-STAR Workshop on Speech-to-Speech Translation, Barcelona, Spain, pp. 199-204, June 19-21, 2006*

[12] J. Adell, A. Bonafonte, J. A. Gómez, M. J. Castro, "Comparative Study of Automatic Phone Segmentation Methods for TTS", *Proc. ICASSP, pp 309-312, Philadelphia, 2005*

[13] A. Bonafonte, A. Moreno, J. Adell, P. D. Agüero, E. Banos, D. Erro, I. Esquerra, J. Pérez and T. Polyakova, "The UPC TTS System Description for the 2008 Blizzard Challenge", *2008*

[14] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A Compact Model for Speaker-Adaptive Training", *in Proc. ICSLP, pp. 1137–1140, 1996*

[15] C.J. Leggetter and P.C. Woodland, "Speaker Adaptation of HMM Using Linear Regression", *Tech. Rep. CUED/FINFENG/TR.181, Cambridge University Engineering Department, June 1994*

[16] T. Toda and K. Tokuda, “A speech parameter generation algorithm considering global variance for HMM-based speech synthesis,” *IEICE Trans. Inf. & Syst., vol. E90-D, no. 5, pp. 816–824, 2007*

[17] HTK Working group. Htkbook http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml

[18] HTS Engine API http://hts-engine.sourceforge.net/

[19] R. Chengalvarayan nand L. Deng, "A Maximum A Posteriori Approach to Speaker Adaptation Using the Trended Hidden Markov Model", *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, vol. 9, no. 5, pp. 549-557, July 2001*

Appendix: Work Plan, Time Gantt and Milestones

Work Plan

Work Packages and tasks

Project: Project Proposal and Workplan	**WP ref: WP1**	
Major constituent: Research and planning		
Short description: obtain deep python acknowledges and learn to work with some useful libraries/environments for deep learning.	Planned start date: 17/02/16 Planned end date: 01/03/16 Start event: Project Start End event: Document delivery	
Task 1.1: Study of the framework Task 1.2: Study of HMM and Speech Synthesis techniques Task 1.3: Study of BASH and Perl syntax Task 1.4: Planning	Deliverables: Project Proposal and Workplan	Dates: 01/03/16

Project: Data Analysis and Selection	**WP ref: WP2**	
Major constituent: Audio listening and text edtion		
Short description: Listen the audio obtained from broadcasts and tag the transcriptions. Create a database of selected sentences (audio and text transcription) Correct text mistakes made during the process (due to different uses of encoding)	Planned start date: 01/03/16 Planned end date: 22/04/16 Start event: End of preliminary studies End event: Obtain all the required data to start the segmentation	
Task 2.1: Audio listening Task 2.2: Transcription labeling Task 2.3: Creation of text files with selected sentences Task 2.4: Cutting selected audios fragments Task 2.5: Correct text errors	Deliverables: Labeled transcriptions Audio fragments	Dates: 22/04/16

<table>
<tr><td>Project: Audio Segmentation</td><td colspan="2">WP ref: WP3</td></tr>
<tr><td>Major constituent: Audio Segmentation</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Short description:
Perform audio segmentation applying speech analysis techniques
on previously labeled data.</td><td colspan="2">Planned start date: 22/04/16
Planned end date: 15/06/16</td></tr>
<tr><td colspan="2">Start event: Obtaining audio fragments
End event: Obtaining a database of audio segments</td></tr>
<tr><td>Task 3.1: Segmentation applying speech analysis techniques</td><td>Deliverables:
Database of audio segments Audio fragments</td><td>Dates:
15/06/16</td></tr>
</table>

<table>
<tr><td>Project: Speech Synthesis Training</td><td colspan="2">WP ref: WP4</td></tr>
<tr><td>Major constituent: Speech Synthesis</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Short description:
Train Hidden Markov Models using the labeled segments in order
to create expressive voices (first strategy)
and adapt one general model to obtain several
expressive voices (second strategy).</td><td colspan="2">Planned start date: 15/06/16
Planned end date: 01/09/16</td></tr>
<tr><td colspan="2">Start event: Revised audio segmentation
End event: Obtain Expressive Speech Synthesis</td></tr>
<tr><td>Task 4.1: Training Markov Models (HMM) using HTK+HTS tools
Task 4.2: Adaptation of a general model
Task 4.3: Speech Synthesis</td><td>Deliverables:
Voices Synthesized</td><td>Dates:
01/09/16</td></tr>
</table>

<table>
<tr><td>Project: Speech Synthesis Evaluation and Improvement</td><td colspan="2">WP ref: WP5</td></tr>
<tr><td>Major constituent: Improve Speech Synthesis</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Short description:
Develop an interface to present the speech synthesis results to different subjects who will evaluate the expressiveness. Based on obtained results improvements to the system may be suggested and implemented.</td><td colspan="2">Planned start date: 15/07/16
Planned end date: 20/09/16</td></tr>
<tr><td colspan="2">Start event: Obtain first Voices Synthesized
End event: Obtain final Voices Synthesized</td></tr>
<tr><td>Task 5.1: Survey
Task 5.2: Obtain the final synthesized speech</td><td>Deliverables: Improved Voices and User Interface</td><td>Dates: 20/09/16</td></tr>
</table>

<table>
<tr><td>Project: Final Document and Project Defense</td><td colspan="2">WP ref: WP6</td></tr>
<tr><td>Major constituent: Documentation</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Short description:
Focus on the review of all the previous documentation. Prepare the Final Document to deliver before 27/06. Prepare the Project Defense for the week from 11/07 to 15/07.</td><td colspan="2">Planned start date: 05/05/16
Planned end date: 17/10/16</td></tr>
<tr><td colspan="2">Start event: Writing of the Final Document
End event: Project Defense</td></tr>
<tr><td>Task 6.1: Writing of the Final Document
Task 6.2: Revision of the Final Document
Task 6.3: Project Defense</td><td>Deliverables: Final Document</td><td>Dates: 28/08/16</td></tr>
</table>

Milestones

WP	Task	Short title	Milestone / deliverable	Date
1	4	Planning	Project Proposal and Workplan	01/03/16
2	3	Creation of text files with selected senten	Labeled Transcriptions	22/04/16
2	4	Cutting selected audios fragments	Audio Fragments	22/04/16
3	1	Segmentation	Database of audio segments	15/06/16
4	3	Speech Synthesis	Voices Synthesized	20/09/16
6	1	Writing of the Final Document	Final Document	28/09/16
6	3	Project Defense	Project Defense	from 17/10/16 to 21/10/16

Updated Time Plan (Gantt diagram)