UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTMENT OF COMPUTER SYSTEMS AND COMPUTATION

MASTER'S THESIS

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Applying Machine Learning technologies to the synthesis of video lectures.

Master in Artificial Intelligence, Pattern Recognition and Digital Imaging.

Santiago Piqueras Gozalbes

Directed by:
Dr. Alfons Juan Ciscar
Dr. Jorge Civera Saiz

September 15, 2014

A mi yaya Lines.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Scientific and technical goals . . . 2
- 1.3 Document structure . . . 3

2 Speech Synthesis **5**
- 2.1 The text-to-speech synthesis process . . . 5
- 2.2 Statistical Parametric Speech Synthesis . . . 6
- 2.3 Open tools . . . 7
  - 2.3.1 HTS . . . 7
  - 2.3.2 SPTK . . . 8
  - 2.3.3 Flite+hts_engine . . . 8
  - 2.3.4 SOX . . . 8
  - 2.3.5 AHOcoder . . . 8
- 2.4 Evaluation . . . 9
- 2.5 Conclusions . . . 10

3 Machine learning techniques **11**
- 3.1 Introduction to machine learning . . . 11
- 3.2 Hidden Markov Models . . . 12
  - 3.2.1 Acoustic modelling with HMM . . . 13
- 3.3 Deep Neural Networks . . . 15
  - 3.3.1 Acoustic modelling with DNN . . . 16
- 3.4 Conclusions . . . 17

4 Corpora description **19**
- 4.1 The poliMedia platform . . . 19
- 4.2 The VideoLectures.NET platform . . . 20
- 4.3 Transcription format . . . 22
- 4.4 Conclusions . . . 23

5 Systems **25**
- 5.1 Overview . . . 25
  - 5.1.1 Training . . . 25
  - 5.1.2 Synthesis . . . 26
- 5.2 Spanish system . . . 28
  - 5.2.1 Data usage and preprocess . . . 28
  - 5.2.2 Linguistic analysis . . . 28

5.2.3 Acoustic models . . . 30
5.3 English system . . . 31
5.3.1 Data usage and preprocess . . . 31
5.3.2 Linguistic analysis . . . 31
5.3.3 Acoustic models . . . 31
5.4 Conclusions . . . 32

6 Evaluation and integration **35**
6.1 Evaluation . . . 35
6.1.1 Experimental setup . . . 35
6.1.2 Results and Discussion . . . 36
6.2 Integration . . . 37
6.3 Conclusions . . . 37

7 Conclusions and Future Work **39**
7.1 Conclusions . . . 39
7.2 Future work . . . 39
7.3 Contributions . . . 40
7.4 Acknowledgements . . . 40

Chapter 1

Introduction

1.1 Motivation

Online lecture repositories are rapidly growing nowadays. Hundreds of platforms host hundreds of thousands of educational videos on practically every subject we may want to learn of. This huge effort, made by universities and innovative educational companies, allows people around the world to acquire from basic to proficiency skills on a wide array of disciplines. Furthermore, many of the multimedia content is being offered to the public free of charge, providing access to education to people on a limited income.

While the idea of a global educational multimedia repository is exciting, there are however some barriers that call to be overcome. The one that inspired this thesis is the language barrier. As it stands, most of the multimedia content readily available is monolingual, driving away potential users. The problem becomes larger when we consider audible content, as in video lectures, which is harder to understand by non-fluent speakers than written content. As a temporal solution, repositories such as Coursera [2] or Khan Academy [5] provide tools to their users in order to allow them to transcribe and translate the content, in a huge collaborative effort. This approach is working for the most popular talks and topics, but it is obviously unsustainable in the long run.

In the last years, the machine learning scientific community has begun to tackle the problem of transcribing and translating these lectures automatically, by using complex Automatic Speech Recognition (ASR) and Machine Translation (MT) systems specifically adapted for this task. These systems can produce subtitle files in a variety of languages, and then the users can select whichever suits their needs. The limited number of speakers (usually one, the lecturer), the relatively good audio conditions and the fact that the topic of the talk is known beforehand have helped the systems to achieve very low error rates, shrinking the gap between machine and human speech recognition.

Regardless of the accuracy, there are two main drawbacks inherent to the subtitle approach. The first one is that the user is forced to split their focus between the video, which usually features either a slide presentation or a video, and the subtitles themselves. The second one is that visually impaired users cannot benefit from the subtitles at all. The aim of this work is to solve both problems by performing the next logical step in this language-adaptation process: to automatically synthesize the speech in the user's native language, by the means of machine learning techniques.

1.2 Scientific and technical goals

The goal of this work is to investigate the current state-of-the-art machine learning techniques, applied to the synthesis of human speech in Spanish and English languages. We aim to produce a system which will receive a subtitle file and will output an audio track, containing the speech signal corresponding to the input text. This audio track can then be presented alongside or embedded in the lecture file as a side track. A modification of the video player will then allow the user to choose what language does he want to listen the talk in.

We aim to produce synthesized speech that is:

Intelligible This is our main goal, as an incomprehensible synthetic voice is a useless one.

Time-aligned We aim to align the synthetic voice with the lecturer's movements. As the user's focus is usually on the lecture slides, this alignment can be performed loosely. Nevertheless, some studies show that big discrepancies between the voice and the speaker's gestures are easily noticed and may distract the viewer [38].

Natural We pursue a natural sounding voice in order to seamlessly integrate the audio track into the video. We pretend to make the user forget they are listening to a synthetic voice, which will help them concentrate on the lecture content.

To help us reach this goals, we have explored novel alternatives to the conventional acoustic modeling approach followed by text-to-speech (TTS) systems. These alternatives are be based on deep neural networks (DNN, Section 3.3). We have carried out a comparison between HMM-based and DNN-based acoustic models for both English and Spanish languages, in order to find out which approach draws us closer to our objective.

Finally, we aim to produce a system that can be applied massively to a repository of video lectures in an automated manner. Such a system needs to be robust and efficient, avoiding audible glitches and large distortions.

1.3 Document structure

This document is divided in seven chapters. Chapter 2 introduces us to the speech synthesis systems basics, with a focus on statistical parametric text-to-speech, as well as open tools to train and use those systems and evaluation measures. Chapter 3 starts with a brief description of the machine learning framework, before detailing two widespread machine learning models (Hidden Markov Models and Deep Neural Networks) and their role in speech synthesis. Then, Chapter 4 details the corpora used in the experiments. Chapter 5 describes the Spanish and English synthesis systems developed in this work. In Chapter 6 we can find the experimentation performed. Finally, Chapter 7 wraps up with the conclusions, future work and contributions derived from this thesis.

Chapter 2

Speech Synthesis

In this chapter the basics of a TTS system are introduced, focusing on statistical parametric speech synthesis. We present the open tools available to train the systems and the problem of performing an objective evaluation of the quality of the synthesized voice.

2.1 The text-to-speech synthesis process

Speech synthesis can be defined as the process of producing an artificial human voice. A text-to-speech (TTS) synthesizer is a system capable of transforming an input text into a voice signal. TTS systems are nowadays used in a wide array of situations, such as in GPS navigation devices, internet services (e.g. RSS feeds or e-mail), as a part of voice response applications, etc.

Usually, the TTS process is divided into two subprocesses, commonly referred as the front-end and the back-end. The front-end deals with the text processing and analysis. This step involves text normalization, such as removing or substituting non-alphabetic graphemes by their alphabetic counterparts (e.g. $\alpha \rightarrow$ alpha), phonetic mapping (assigning phoneme transcriptions to words) and linguistic analysis. Commercial TTS systems often use a combination of expert and data-driven systems to implement the front-end.

The back-end is responsible for transforming the output of the front-end into a speech signal, involving a process often known as acoustic mapping. This mapping can be performed at different levels, such as frame (with or without fixed length), phoneme, diphone, syllable or even word level. After the mapping, the results are concatenated to form the speech signal. Nowadays, there are two main approaches to the back-end of TTS systems, unit selection synthesis and statistical parametric synthesis, both of which are data-driven. Unit selection divides the training data into small units, usually diphones. In order to perform the synthesis, the units are selected from a database based on some suitability score and then concatenated with

the adjacent units.

While US methods are known to produce the most natural sounding speech, statistical approaches have surpassed unit selection in terms of intelligibility [22]. We prefer an intelligible lecture than a natural sounding lecture. This is main reason why we have decided to investigate the statistical approach rather than the unit selection approach. In the next section parametric statistical speech synthesis is described in detail.

2.2 Statistical Parametric Speech Synthesis

Statistical parametric speech synthesis [55] assumes that the voice recordings can be reconstructed with a limited number of acoustic parameters (or features), and those parameters follow a stochastic distribution. The goal of the system is to accurately model these distributions and later make use of them to generate new speech segments. In order to train the models, a wide array of well-known techniques from the machine learning field can be applied, such as the ones presented in Chapter 3.

In order to perform an accurate synthesis, statistical parametric TTS systems combine phoneme information with contextual information from the syllable, word and utterance that surround the phoneme, creating what is known as context-dependent phonemes (CD-phonemes). This contextual information is provided by the front-end module. CD-phonemes often have high dimensionality, which complicates the estimation. Furthermore, many of the CD-phonemes we found at test stage will have not been seen in the training corpora. Our acoustic models will need to deal with this issue.

The parametrization and reconstruction of the audio signal is performed in a process known as vocoding. The simplest model used assumes a source-filter division: a sequence of filter coefficients that represent the vocal tract, and a residual signal that corresponds to the glottal flow [26]. This model is based in human speech production and assumes that the sounds can be classified as voiced or unvoiced. A voiced sound is produced when the vibration of the vocal cords is periodic, such as in the production of vowels. The voiced segments carry a certain fundamental frequency which determines the pitch. Conversely, an unvoiced sound is produced when this vibration is chaotic and turbulent. A diagram summarizing a simple source-filter model-based decoder can be found in Figure 2.1.

Unfortunately, the separation between voiced and unvoiced does not accurately match reality. Many phonemes are produced by a combination of voiced, quasi-voiced and unvoiced. Performing hard classification results in a metallic, buzzy voice, which sounds far from natural. As way of solution, more advanced vocoders have been proposed in the last years, such as STRAIGHT [21], that include additional parameters to diminish this issue. However, the problem of determining which parameters will

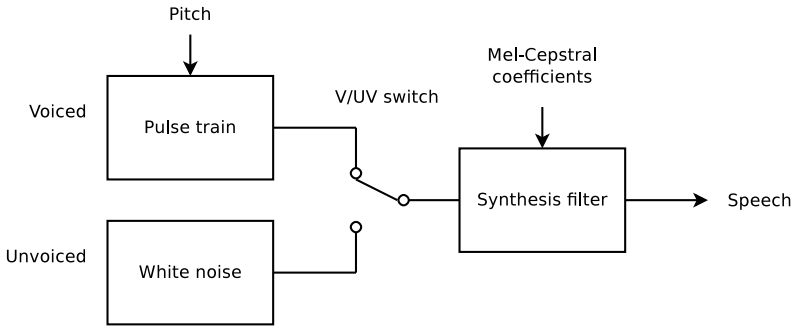


Figure 2.1: A simple source-filter decoder

reconstruct the human voice with high intelligibility and naturalness, while maintaining a set of statistical properties that allow us to learn the acoustic models is still an open one. A comparison of state-of-the-art vocoders can be found in [18].

In order to deal with the discontinuity problems that often arise from a frame to frame generation, dynamic information such as first and second time derivatives are introduced and later used by algorithms that smooth the acoustic parameter sequence. An example of one of those algorithms is the Maximum Likelihood Parameter Algorithm (MLPG) [41]. This algorithm receives a Gaussian distribution (means and variables) of the acoustic features and their time derivatives and outputs the maximum likelihood feature sequence. This procedure improves the naturalness and reduces the noise. On the other hand, it results in a reduction of the high frequencies, causing a muffled voice effect.

2.3 Open tools

There are many open tools available to process and transform the audio signal, extract the acoustic features and train the acoustic models. We present here a list of the tools that have been used at some point or another in this project.

2.3.1 HTS

The HMM-based Speech Synthesis System (HTS) is a patch for the Hidden Markov Model Toolkit (HTK) that allows users to train Hidden Markov Models (see Section 3.2) to perform the acoustic mapping in TTS systems [4]. Over the years, it has seen the inclusion of state-of-the-art methods, such as the estimation of Hidden semi-Markov Models [56], speaker adaptation based on the Constrained Structural Maximum *a posteriori* Linear Regression (CSMAPLR) algorithm [28], cross-lingual speaker adaptation based on state mapping [47], and many more. HTS uses a modified BSD license, which allows its use for both research and commercial applications. It is widely used by many successful research groups, as evidenced by the results of

the speech synthesis Blizzard Challenge, organized yearly by the University of Edinburgh [23].

In this work, we have used HTS in its last stable version (2.2, released July 7 2011) to train HMM acoustic models and Gaussian duration models to use with both HSMM and DNN models. The training demos provided by the HTS team have been used as a base to develop the English and Spanish back-ends.

2.3.2 SPTK

The Speech Signal Processing Toolkit is "a suite of speech signal processing tools for UNIX environments" [37]. It is developed by the Nagoya Institute of Technology and distributed under the a modified BSD license which, just like HTS, allows unlimited personal and commercial use. It comprises a set of tools to perform all kinds of acoustic parameter sequence transformations, vector manipulation and other useful data manipulation programs. SPTK has been widely used in this work.

2.3.3 Flite+hts_engine

"Flite+hts_engine" is a free TTS English synthesis system developed by HTS working group and Nagoya Institute of Technology students [3]. It can perform speech synthesis with HTS trained models. In this work, we have used the front-end linguistic analysis of "Flite+hts_engine" for our English system.

2.3.4 SOX

SoX is a general purpose digital audio editor, licensed under LGPL 2.0. It provides the tools to create, modify and play digital audio files; spectrogram analysis and transforming between audio file formats [39]. We make an extensive use of SoX features in this thesis: concatenate the synthesized segments, perform noise reduction, apply high/low-pass filters, etc.

2.3.5 AHOcoder

AHOcoder is a free, high quality vocoder developed by the Aholab Signal Processing Laboratory of the Euskal Herriko Unibertsitatea, Spain [1]. We have chosen AHOcoder as our vocoder in the TTS systems, based on its permissive license, easiness of use and promising results [13], which prove it can match and even improve the results of other state-of-the-art vocoders.

AHOcoder is based on a Harmonics plus Noise model, instead of an harmonics-or-noise approach that is featured in Figure 2.1. It makes use 3 kinds of acoustic features: Mel-cepstral coefficients (*mfc*), which carry the spectral information; the logarithm of the fundamental frequency ($\log F_0$), which determines the pitch; and the maximum voiced frequency (*mvf*), which provide a separation point for the voiced

segments, where the higher frequencies are considered to be noise. log F_0 and *mvf* features will be referred later as excitation features.

2.4 Evaluation

The evaluation of a speech synthesis system is a complex problem. Concepts as intelligibility and naturalness are hard to measure objectively. This motivates many research problems to perform both objective and subjective evaluation of the results. The voices are listened by experts and non-expert users alike and then scored between 1 and 5 in what is known as a Mean Opinion Score tests [33]. Subjective tests are often expensive and require the collaboration of users not affiliated to the project, and as such, they cannot always be performed. There are many works that deal with the use of objective error measures for TTS evaluation and their relation with the subjective scores [11]. In this thesis, we performed objective evaluation to compare different approaches to the acoustic mapping problem.

We have used 3 different measures to objectively evaluate the quality of the synthesized voices. This measures cannot be considered standard, but they are widely used in other works.

Mean mel cepstral distortion (MMCD). This measure evaluates the quality of the cepstrum reconstruction and has been linked to higher subjective scores [24]. The MMCD between two waveforms is computed as:

$$MMCD(v^{tar}, v^{syn}) = \frac{\alpha}{T} \sum_{\substack{t=0 \\ ph(t) \notin SIL}}^{T-1} \sqrt{\sum_{d=s \in \{0,1\}}^{D} (v_d^{tar}(t) - v_d^{ref}(t))^2} \tag{2.1}$$

where

$$\alpha = \frac{10\sqrt{(2)}}{ln10} \tag{2.2}$$

and v^{tar} is the target waveform, v^{syn} is the synthesized waveform, $v_d(t)$ is the value of the d cepstral coefficient in the frame t. The cepstral distortion is not computed for the silence frames. Notice also the parameter s, which can be 0 or 1 depending on whether the energy of the audio signal is included or not. In this work, we have not included the energy, as the audio recordings were not specifically recorded for the training of a synthesizer. Finally, we assume that the number of frames of the target and synthesized waveforms are the same.

Root Mean Squared Error (RMSE). The RMSE is a standard error used in many fields to compute the difference between the target values of a sequence and the predicted values. We use the RMSE to assess the difference between the pitch ($logf_0$) of the synthesized and original voices.

Classification error %. It is computed as the number of wrongly classified samples divided by the total of observations. We make use of this measure to evaluate the performance of the systems when it comes to Voiced/Unvoiced frame classification.

2.5 Conclusions

We have discussed the problem of synthesizing a voice signal from a given text. We have described the most interesting approach for our purposes, known as statistical parametric speech synthesis. Lastly, we have also reviewed the open tools for speech synthesis and detailed the objective evaluation measures that have been used in this project. It can be seen that speech synthesis is a complex problem, where many decisions will involve trade-offs between intelligibility, naturalness and computational costs. At the same time, the evaluation of the results is not a straightforward issue. These challenges have contributed to motivate this research.

Chapter 3

Machine learning techniques

In this Chapter, we briefly review machine learning theory and techniques particularly relevant to this work. Then we describe two models that are widely used in state-of-the-art TTS systems, Hidden Markov Models (Section 3.2) and Deep Neural Networks (Section 3.3), as well as how they can be integrated into the Speech Synthesis framework to perform acoustic mapping.

3.1 Introduction to machine learning

Machine learning (ML) is a branch of computer science that deals with the problem of learning from the data. The goal of ML is to produce computer programs to solve tasks where human expertise does not exist, or where humans are unable to explain their expertise [7]. A machine learning system makes uses of mathematical models to reach its goal. In this work, we are going to focus on supervised learning, where the system is presented with labeled data (that is, that contains the inputs and the corresponding desired outputs) and the goal is to learn the general rule to map inputs to outputs. Typical problems dealt in supervised learning include:

Classification A certain object or group of objects needs to be assigned a label between a set of potential classes. Classification might be binary (2 classes) or multiclass (more than 2 classes).

Structured prediction In this problem, which is closely related to classification, the input object needs to be assigned a certain structured output, such a tree or a string.

Regression Involves the learning of a certain unknown real-valued function f(X).

We are going to focus on the problem of regression, as it is the one that TTS acoustic models need to deal with. A generic machine learning system for a regression

problem can be found in Figure 3.1. As we can see, it is divided into 2 stages. The *training* stage involves the learning of the model parameters with the help of labeled data. The *test* stage allows for the obtention of the model's prediction f'(X), given an arbitrary unlabeled input object X. There are three main steps involved in this process:

1. **Preprocess** The signal is acquired from the object, then filtered to remove noise and prepared for the feature extraction.

2. **Feature extraction** From the processed signal, the relevant information is acquired and a feature vector is computed. It is considered relevant information anything that allows us to predict f(X) more accurately.

3. **Regression** With the feature vector and the trained models, we compute an output prediction f'(X).

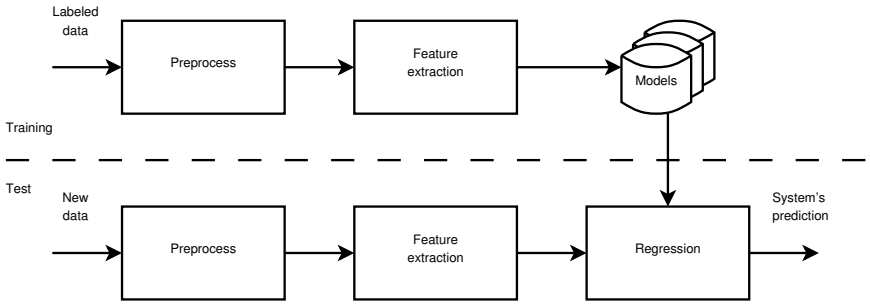


Figure 3.1: A generic machine learning system for regression

3.2 Hidden Markov Models

A Hidden Markov Model (HMM) is a generative model used to model the probability (density, when the variables are continuous) of an observation sequence [19]. It is assumed that the sequence is generated by a known (topology-wise) finite state machine where each state generates an observation with a certain probability distribution. It is called *Hidden* when the states associated to an observation are not visible. An HMM can be characterized with:

Number of states. The usual approach is to include M states, plus 2 special states I and F, that correspond to the initial and final states respectively.

State transition probability matrix. This matrix holds the probability of transiting from a state i to a state j.

Emission probability (density) function. This function is parametrized by a state i and a certain given observation o, and defines the probability (density) of emmiting o given the current state i.

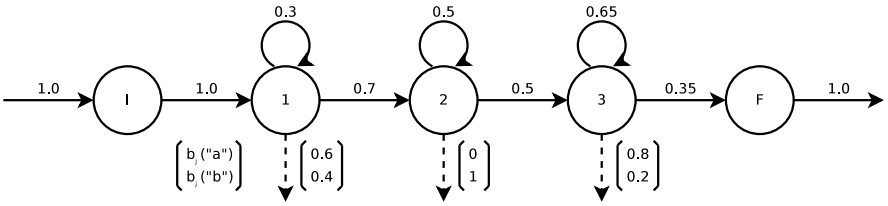


Figure 3.2: A simple HMM with 3 states (not counting I and F) and 2 possible emission values "a" and "b"

We are going to focus on the HMM where the observations variables are continuous, as in the case of acoustic features in TTS. In this case, the usual approach is to employ Gaussian distributions to characterize the emission density function. As the acoustic features are not single valued but vectors, the HMM will feature a mean vector and a covariance matrix for each state. In order to speed up the training, it is common to restrict the covariance matrices to diagonal variance vectors. Finally, instead of a single Gaussian distribution, emission function can be characterized by a Gaussian mixture distribution, which has been applied successfully to other speech-related machine learning tasks [32].

3.2.1 Acoustic modelling with HMM

Over the years, there has been many research and development in statistical parametric TTS that involves the use of HMM to perform acoustic mapping [48, 55]. To perform this mapping, an HMM is trained for each CD-phoneme where the observations correspond to the acoustic features which will later be used by vocoder to reconstruct the voice. As outlined in Section 2.2, training a CD-HMM for each possible combination of text analysis features is unrealistic and would result into poorly estimated HMMs. By way of solution, context clustering techniques at a state-level are used. Clustering is performed by means of binary decision trees. In the training phase, the Minimum Description Length (MDL) criterion is used to construct these decision trees [35]. As the spectral and excitation features have different context dependency, separate trees are built for each one. This approach allows our model to handle unseen contexts, and it is also for the Gaussian duration model. We can see an example of part of a real decision tree of the Spanish system in Figure 3.3.

If we want to use HMM as a generative model, one of the problems that need to be solved is that state occupancy probability decreases exponentially with time, which means that the highest probability state sequence is the one where every state is only visited once. To overcome this limitation, a modification of the HMM model,

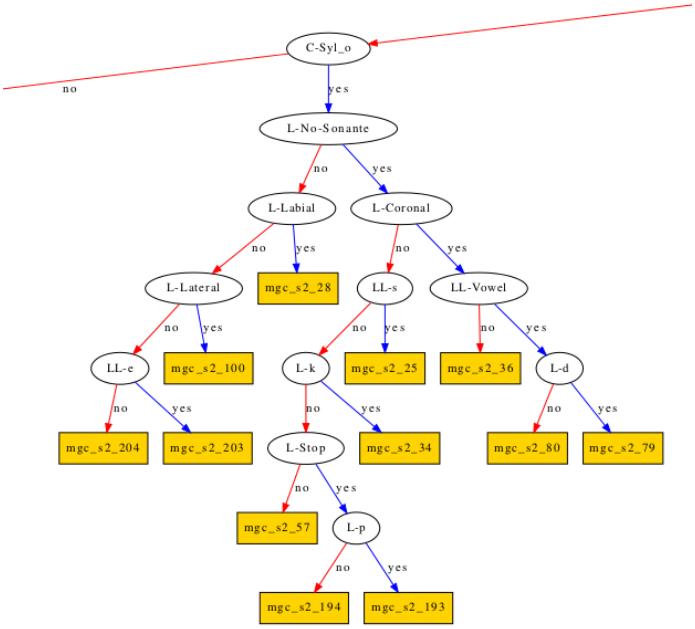


Figure 3.3: A sample of part of a binary decision tree for the first state of the cepstral coefficients of the Spanish HMM system. Notice that most of the decisions depend on the left phoneme (L-*), which reveals strong temporal dependency between adjacent phonemes.

known as **Hidden semi-Markov Model (HSMM)** [9] is preferred. When using a HSMM approach for speech synthesis, state occupancies are estimated with Gaussian probability distributions. This model has been shown to achieve highest scores in subjective tests [56].

In the generation step, first the state durations for each state of each phoneme are predicted by a Gaussian distribution model. Then, we make use of the binary decision trees to select the states and concatenate them into a segment HMM. Finally, the means and variances of the output acoustic feature vector are generated by the segment HMM. However, maximizing the probability of the output sequence would involve emitting the mean value of the current state at every frame, resulting into a segmented feature vector that does not accurately match reality. The MLPG algorithm is used to alleviate this issue. As the MLPG algorithm needs the first and second time derivatives of the acoustic features, the HMM output vector will need to contain them, multiplying the length of the emission vector by 3.

An extra problem emerges from the modelization of the non-continuous features log F_0 and *mvf*. These features are defined in the regions known as "voiced", and undefined in the regions known as "unvoiced". In this thesis, log F_0 has been modeled with a multi-space probability distribution [42], while the *mvf* feature was added as an extra stream and modeled with a continuous distribution, as suggested in [13]. The *mvf* values were interpolated in the unvoiced frames.

3.3 Deep Neural Networks

A neural network (NN) is a discriminative machine learning model composed of neurons that receives an input real-valued vector and returns another real-valued vector. The nodes of a NN are known as neurons. A neuron is composed of one or more weighted input connections and performs a (often nonlinear) transformation into a single output value. NN organize neurons in layers. Every layer is composed by a group of neurons that receive the output of the lower layers. There are no connections between neurons of the same layer.

In Figure 3.4 we can see a diagram of a typical feedforward (i.e. without cycles) network. The input neurons are connected to a hidden layer, which is connected to the output layer. NN with a single hidden layer are considered "shallow", while NN with more than one hidden layer are usually referred as "deep" (DNN). Although it has been known for a while that NN and DNN are capable of approximating any measurable function to any degree of accuracy given enough units on the hidden layer [17], DNN were not widely used until recent years because of the prohibitive computational cost of the training. However, thanks to the advances in their training procedures (such as unsupervised pretraining [12, 16]) and the use of GPUs instead of CPUs [31], which can perform costly matrix operations much faster thanks to their massive parallelism capabilities, DNN and their variants have seen a big resurgence and have been successfully applied to many machine learning tasks [8, 15, 25].

The transformation performed by a single neuron j is described in Equation 3.1.

$$y_j = f(b_j + \sum_i y_i w_{ij}) \tag{3.1}$$

where y_j is the output of neuron j, b_j is the bias, w_{ij} is the weight of the connection between neuron i and j, and f is a non-linear function[1]. Common non-linear functions used in NN are the sigmoid function, the hyperbolic tangent function, the softmax function (for classification problems) and, more recently, the rectified linear function [27]. In this work, will be using the sigmoid function:

$$S(x) = \frac{1}{1 + e^{-t}} \tag{3.2}$$

[1] Linear functions are sometimes used on the output layer

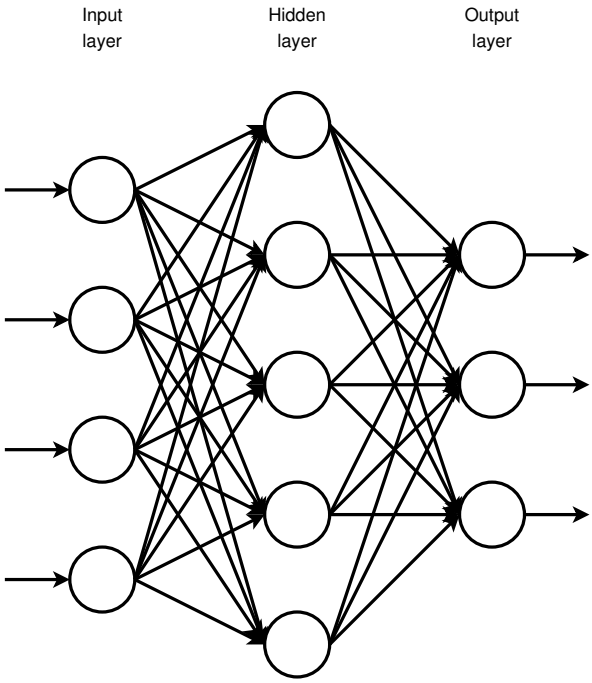


Figure 3.4: A shallow neural network

Please note that the sigmoid function restricts its output to be bounded between 0 and 1, something that must be considered when performing regression of unbounded real values.

3.3.1 Acoustic modelling with DNN

We can perform acoustic modelling with feed-forward DNN, by generating the acoustic parameters frame by frame [54]. While this approach is not new [20], the recent advances presented in the previous section have motivated researchers to take a second look. A diagram detailing the process can be found in Figure 3.5.

Acoustic DNN models receive as an input the information of the CD-phonemes as numeric values, which is then augmented with temporal information of which frame we want to generate, and emit the acoustic features and their time derivatives for the given frame. One of the biggest advantages over the HMM-based approach is that no context clustering is performed, and a single network can model all of the acoustic features at once, using all of the training data available. This results into better generalization.

DNN-based acoustic mapping does not result into the step-wise sequence that

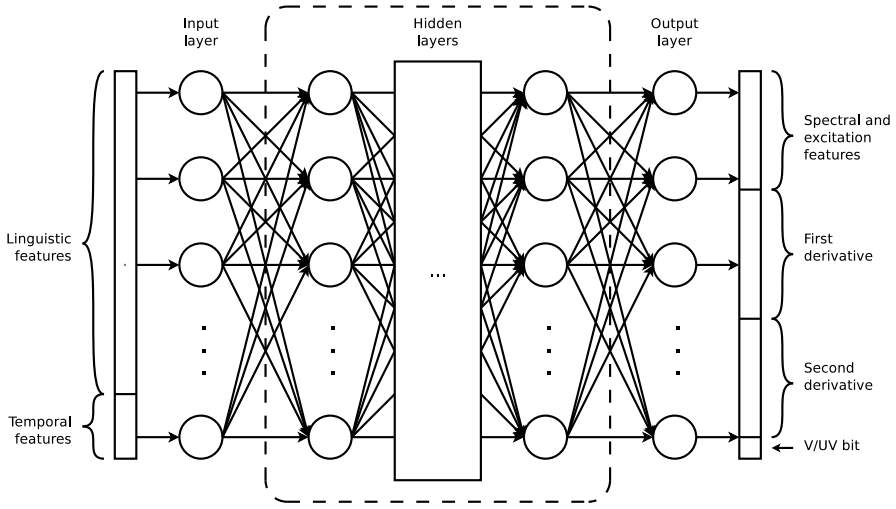


Figure 3.5: A deep feed-forward neural network for speech synthesis.

a maximum likelihood approach for HMM suffer, and so dynamic features are not strictly needed. However, in order to enforce smoothness over time and avoid audible glitches, the DNN will also model the first and second derivatives. By setting the DNN output as the mean vector and computing a global variance from all the training data, we will be able to apply the MLPG algorithm.

The discontinuity problem of the log F_0 and mvf features can be avoided by introducing a V/UV classification bit to the output, and performing interpolation of these acoustic features in the unvoiced frames, an approach known as explicit voicing modelling [52]. When the V/UV bit output is higher than 0.5, the frame is classified as voiced and the value of the features is the same as the network output. When the V/UV bit is lower than this threshold, the frame is considered unvoiced and a special value indicating that the feature is undefined is used instead.

3.4 Conclusions

We have reviewed two approaches to the acoustic mapping problem of statistical parametric speech synthesis systems, and described how they deal with some of the common problems. Chapter 5 will give a detailed explanation of the implementation, while Chapter 6 will provide an objective comparison between both approaches.

Chapter 4

Corpora description

In this Chapter, we describe the corpora used in the development of this thesis. Section 4.1 describes the poliMedia platform and the corpus derived from it, which contains Spanish lectures. Meanwhile, Section 4.2 describes our English corpus, which comes from Videolectures.NET platform. Finally, Section 4.3 briefly describes the format of the transcriptions available.

4.1 The poliMedia platform

The poliMedia (pM) platform is a service created by the Polytechnic University of Valencia for the distribution of multimedia educational content [30]. It allows teachers and students to use a centralized platform in order to create, distribute and access to a wide variety of educational lectures. The platform was created in 2007 and it currently contains more than 2400 hours of video. Furthermore, many of those videos are openly accessible to the public. poliMedia statistics are summarized in Table 4.1.

Tables 4.1: Statistics of the poliMedia repository

Videos	11662
Speakers	1443
Hours	2422

poliMedia video lectures feature a high signal to noise ratio, thanks to the special studio they are recorded on. They also feature a single lecturer, speaking about a certain known topic. These circumstances motivated the use of the repository as a case study in the transLectures project [36]. This project, starting in October 2011, has been providing the pM platform with automatically generated accurate transcriptions and translations for all the videos. These transcriptions are available to the users through the paella video player, and can be edited by them using the transLectures

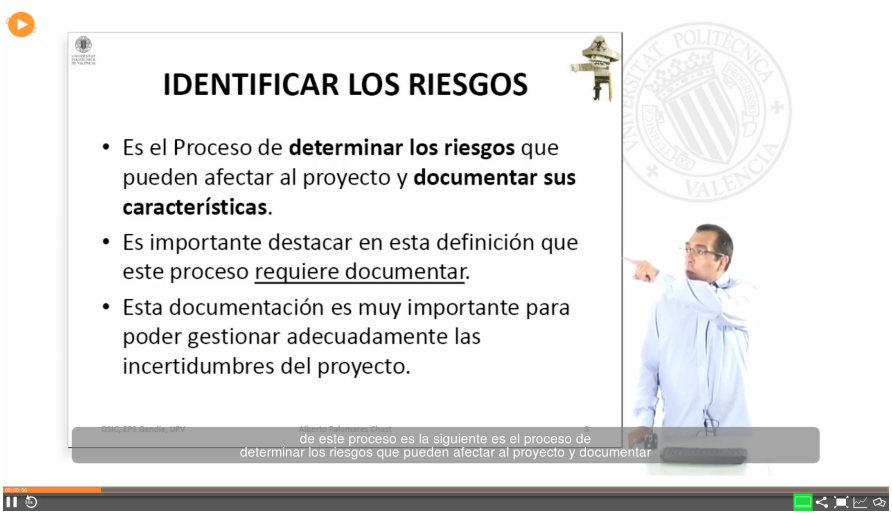


Figure 4.1: A video lecture with subtitles in the paella player

platform [40]. We can see an example in Figure 4.1.

Additionally, the transLectures project has created a training corpus in Spanish composed of over a hundred hours of manually transcribed and revised lectures from the pM repository. The corpus statistics are detailed in Table 4.2.

Tables 4.2: Statistics of the poliMedia corpus

Videos	704
Speakers	83
Hours	114
Sentences	41.6K
Words	1M

We will use this corpus in order to train a TTS system, as the transcriptions are accurate and the acoustic conditions are good enough. However, it is not optimal, as lectures are often noisy (e.g. with coughs and speaker hesitations such as "mmm" or "eee"). It is expected that the high volume of data available will minimize the problems that arise from these circumstances.

4.2 The VideoLectures.NET platform

Videolectures.NET (VL.NET) is an free and open educational repository created by the Jožef Stefan Institute, which hosts a huge number of lectures of many different

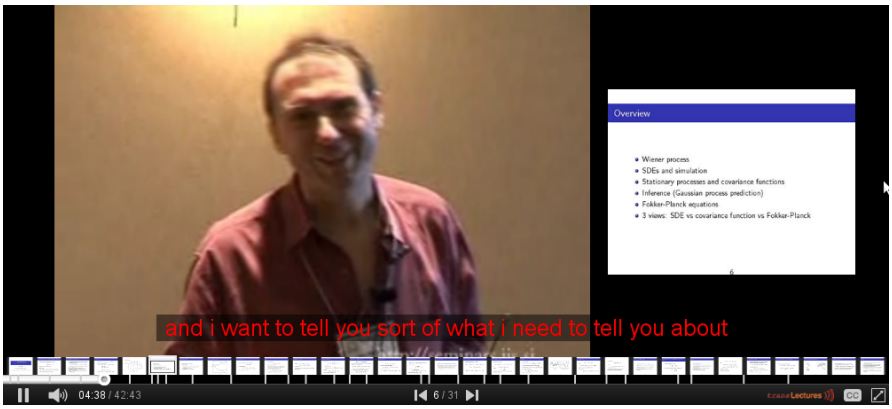


Figure 4.2: A video lecture from VL.NET with subtitles

scientific topics [46]. They aim to promote scientific content, not just to the scientific community but also to the general public. As of September 2014, they provide more than 16000 lectures, 15174 of which are in English. Around a 55% of those talks belong to the topic of computer science, showing that CS is one of the faster fields to embrace the educational revolution today's technologies provide. Many of the videos also provide time-aligned slides, as seen in Figure 4.2. Statistics of the Videolectures.NET platform are summarized in Table 4.3.

Tables 4.3: Statistics of the Videolectures.NET repository

Videos	19106
Speakers	12425
Hours	9545

Unfortunately, Videolectures.NET talks do not share the same acoustic conditions as poliMedia lectures. While pM lectures are recorded in a special studio, lectures from VL.NET are recordings of conferences, workshops, summer camps and other scientific promotional events. As such, more often than not they feature a live audience, which may participate in the talk (e.g. asking questions) and add noise to the audio (e.g. claps, laughs, murmurs). The quality of the microphone(s) used greatly varies between lecturer and it has also a big impact on the final recording.

Videolectures.NET is the other main case study of the transLectures project. Most of the older talks have been transcribed and translated with the best transLectures systems, while newer lectures are expected to be transcribed soon. It is then a good candidate for us to train our systems and to test them in a real setting. In this work, we have used one of the subcorpus derived from the VL.NET repository, which derives from the manually subtitles talks created by video lectures users. This subtitles are

not literal transcriptions, as repetitions, and hesitations are not included, and many lecturer mistakes have been fixed. In order to create a corpus suitable for the training of ASR and TTS systems, the refinement process described in [45] was applied. The final corpus statistics can be found in Table 4.4.

Tables 4.4: Statistics of the VL.NET corpus

Videos	224
Speakers	16
Hours	112h
Sentences	98.7K
Words	1.2M

While the number of hours is similar to the pM corpus, the number of hours per speaker is much higher. As the TTS systems are usually trained for a single speaker, the English will make use of more hours than the Spanish one. This will account for the fact that the acoustic conditions of this corpus are worse than the pM Spanish corpus.

4.3 Transcription format

In this thesis, the corpora used for both Spanish and English systems consisted of video files with their corresponding transcriptions (subtitles). The format of this transcriptions is TTML-DFXP, with the extensions proposed for the transLectures project [44]. We can see below a real example of the start of a DFXP file.

```
<?xml version="1.0" encoding="utf-8"?>
<tt xml:lang="en" xmlns="http://www.w3.org/2006/04/ttaf1"
xmlns:tts="http://www.w3.org/2006/10/ttaf1#style"
xmlns:tl="translectures.eu">
<head>
<tl:d aT="human" aI="UPV" aC="1.00" cM="1.0000" b="0.00" e="657.75"
st="fully_human"/>
</head>
<body>
<tl:s sI="1" cM="1.0000" b="3.06" e="10.72">
Hello, my name is Mónica Martínez, and I am a lecturer at Universidad
Politécnica de Valencia&amp;apos;s Department of Applied Statistics,
Operational Research and Quality.
</tl:s>
<tl:s sI="2" cM="1.0000" b="11.20" e="17.92">
In this lecture, I intend to show you how to build and read
```

```
one-dimensional frequency tables.
</tl:s>
...
```

As we can appreciate, the DFXP holds a variety of information at document level regarding to who made the transcription, the mean confidence measure cM, which will be 1 for human transcriptions and $cM \in]0, 1]$ when the transcription is automatic, the beginning and end times. The rest of the transcription is divided in segments, with a segment id cM, a confidence measure cM, and the beginning and end times (b and e, in seconds). While the DFXP file may contain other information (e.g. alternative transcriptions, confidence measures at word level, etc.) our system does not make any use whatsoever of that info. We assume that the latest alternative available is the best alternative, and synthesize that one.

4.4 Conclusions

We have described the corpora used in the development of this thesis, outlined the corpora characteristics and how they will affect the training of our synthesis systems. We have also detailed the transcription format. A comprehensive report of the use that has been made of the corpora is provided in Chapter 5.

Chapter 5

Systems

In this chapter we describe the systems developed and implemented for this thesis. We begin by giving an overview of the shared parts of the Spanish and English systems in Section 5.1. A detailed explanation of the Spanish system specifics is given in Section 5.2, while the English system is detailed in Section 5.3.

5.1 Overview

5.1.1 Training

In Figure 5.1(a) we can see an scheme of the training process. We describe now the steps carried out in order to train our TTS systems.

Filtering and preprocess We start by extracting the audio from the video file and performing segmentation of the audio according to the temporal marks of the segments in the transcription file. The audio is then resampled to 16Khz and left and right audio channels are mixed to a single one. We also perform a filtering process, where some of the audio segments were regarded as unhelpful and subsequently removed. More details are provided in the language specific Sections 6.1.1 and 5.3.1.

Linguistic analysis In this step, the text is analyzed and a grapheme-to-phoneme conversion is carried out. The objective is to transform the text segment to a list of context-dependent phonemes. We used different tools to perform the analysis in English and Spanish. Please refer to Sections 5.2.2 and 5.3.2 to see the details.

Acoustic features extraction We used AHOcoder *ahocoder* tool to extract the acoustic features from the waveforms. After the extraction, we computed the first and second derivatives with the scripts provided in HTS demo. Finally, for the DNN systems only, we performed linear interpolation of the *lf0* and *mvf* features inbetween the frames they are not defined (unvoiced frames).

Training This step involves the learning of the model parameters from the acoustic and linguistic features. Depending on the model we want to train (HMM or DNN), the procedure greatly varies.

HMM We trained the HMM system with HTS, adapting the HTS' English STRAIGHT demo to our needs. In the case of Spanish, this step involved modifying the clustering questions file to Spanish phonology. We also needed to modify the training script, as the *bap* stream will now model the *maximum voiced frequency* feature instead. The system's output include 3 different models for both duration and acoustic feature models:

- **1mix** Single Gaussian distribution, with diagonal covariance matrices.
- **stc** Single Gaussian distribution, with semi-tied covariance matrices.
- **2mix** Gaussian mixture (2) distribution, with diagonal covariance matrices.

In this work we have used the 2 mixtures Gaussian for the HTS tests, as we found out the quality of the resulting voice was higher.

DNN The training of the DNN involved processing the linguistic analysis output to adapt it to the DNN input format. There are three type of linguistic features: binary, numeric and categorical. Binary and numeric features are provided as is, whereas categorical features are encoded as 1-of-many. All inputs are normalized to have zero mean and unit variance. Meanwhile, the outputs have been normalized to lie between [0.01,0.99] values. The maximum and minimum were extracted from all the training data.

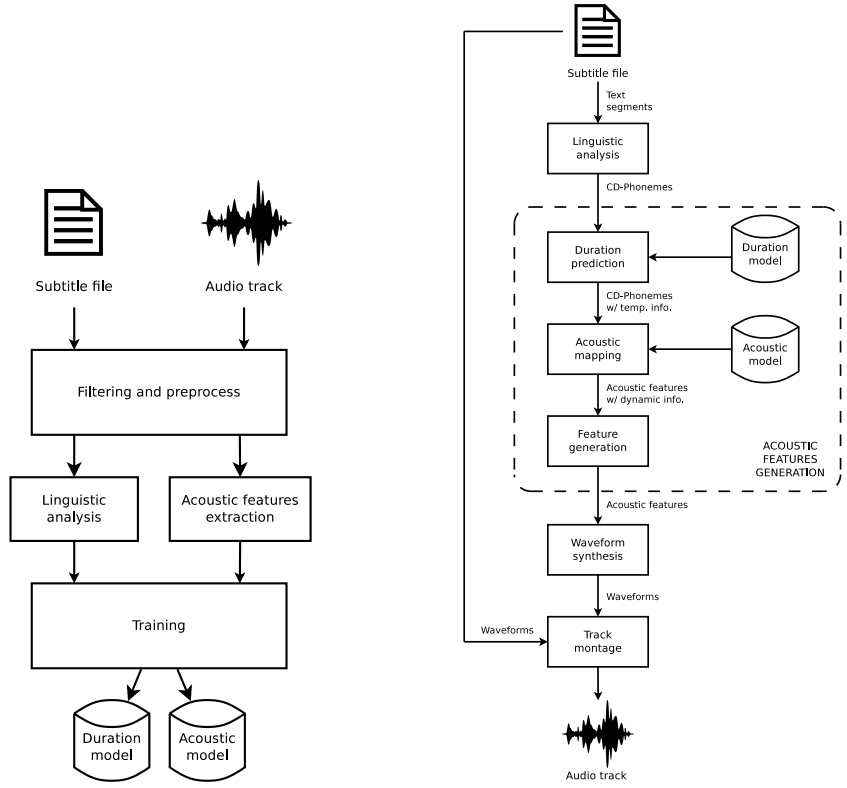
The training was performed with a toolkit developed for the transLectures project, which utilizes the CUDA toolkit [29] to parallelize the training in the GPU. This toolkit was modified to perform regression (as ASR DNN models are used for senone classification) with MSE as the error criterion for backpropagation. Neural networks with more than one hidden layer where pretrained using a discriminative approach [34], and then fine-tuned with a stochastic minibatch backpropagation algorithm [10].

5.1.2 Synthesis

In Figure 5.1(b) we provide an overview of the modules that compose our TTS synthesis system. We describe the modules involved in our system from the moment the subtitle file is received to the point the speech output is ready to be embedded.

Linguistic analysis The linguistic analysis performed is the same as the one involved in the training of the system.

Duration prediction The duration of the phonemes (DNN) or the HMM states (HMM) are predicted by the Gaussian duration model. This procedure involves traversing the binary clustering tree of the model until a leaf is selected. Although the duration with the highest probability would be equal to the mean of



(a) System training (b) Synthesis overview

Figure 5.1: Overview of the training and synthesis processes

the Gaussian, in order to keep temporal alignment between the audio and the video, we want to be able to modify the duration of the synthesized segment to match the duration of the corresponding original audio segment. As a solution, to determine the final duration of each state/phoneme we have implemented the algorithm presented in [50].

Acoustic mapping The acoustic mapping process has been thoroughly described in Sections 3.2.1 and 3.3.1. We mention now the tools that our system makes use of.

HMM The HMM mapping is performed with HTS' *HHEd* (make unseen models) and *HMGenS* (feature generation) tools, with Case 1 of the Speech Parameter Generation Algorithm [43].

DNN The DNN mapping is performed with transLectures DNN toolkit.

Feature generation With the acoustic features, their time derivatives, and the vari-

ances (which are generated by the HMM in the case of HMM-based model, and precomputed from all the training data in the case of DNN acoustic model), we apply the Maximum Likelihood Parameter Generation (MLPG) with algorithm [41] to enforce temporal smoothness. We use SPTK's *mlpg* tool for this purpose.

Waveform synthesis We further improve the naturalness of the speech by applying an spectral enhancement based on post-filtering in the cepstral domain [51]. Then we make use of AHOcoder's *ahodecoder* tool to generate waveforms from the acoustic features predicted by the model. The result is the individual audio segments that compose the talk.

Track montage We make use of the timestamps of the subtitle file to compose the audio track of the talk, by alternating silences and voice segments. As some of the voices sometimes carry out a residual noise, which can be easily detected by users wearing headphones, we found out that applying sox's *noisered* tool for noise removal to the full track can help getting rid of the noise at the cost of voice naturalness. The synthesized track is now complete and ready to be embedded.

5.2 Spanish system

5.2.1 Data usage and preprocess

We have extracted a subcorpus from the poliMedia corpus (Section 4.1) to train our TTS Spanish system. This subcorpus features 39 videos with 2273 utterances by a single male native Castillian Spanish speaker. We performed automatic phoneme alignment with the best acoustic model deployed in the transLectures project at month 24 [6]. After the alignment, two segments were removed because of their low probability[1]. The final subcorpus statistics are collected in Table 5.1.

Tables 5.1: Statistics of the corpus for the Spanish TTS system

Videos	39
Speakers	1
Hours	6 (w/o silences)
Segments	2271
Phonemes	305767

5.2.2 Linguistic analysis

We have developed our linguistic analyzer derived from the grapheme-to-phoneme converter used in transLectures project (*syllables.perl*). As Spanish is a highly pho-

[1] We later found out that while transcription was correct, but the temporal alignment of the segments were not.

netic language, the *gtp* conversion can be performed without much loss. The complete list of features the CD-phonemes include is provided in Table 5.2. This information is augmented for the DNN acoustic models with four temporal features of the frame to be synthesized (Table 5.3).

Tables 5.2: Linguistic features of Spanish system.

Level	Feature	Type*
Phoneme	Left-left phoneme identity	C
	Left (previous) phoneme identity	C
	Current phoneme identity	C
	Right (next) phoneme identity	C
	Right-right phoneme identity	C
	Position of the phoneme in the syllable (forward)	N
	Position of the phoneme in the syllable (backward)	N
Syllable	Is left syllable stressed?	B
	No. of phonemes in left syllable	N
	Is current syllable stressed?	B
	No. of phonemes in current syllable	N
	Pos. of current syllable in word (forward)	N
	Pos. of current syllable in word (backwards)	N
	Pos. of current syllable in segment (forwards)	N
	Pos. of current syllable in segment (backwards)	N
	No. of syllables from previous stressed syllable	N
	No. of syllables to next stressed syllable	N
	Vowel in current syllable	C
	Is right syllable stressed?	B
	No. of phonemes in right syllable	N
Word	No. of syllables in left word	N
	No. of syllables in current word	N
	Pos. of current word in segment (forward)	N
	Pos. of current word in segment (backwards)	N
	No. of syllables in right word	N
Segment	No. of syllables in current segment	N
	No. of words in current segment	N

* C=Categorical, B=Binary, N=Numeric

We use 23 phonemes and 2 special symbols to perform the grapheme-to-phoneme conversion. The special symbols are *SP*, to denote silence, and *NIL*, which is added at the start and the end of the segments. The complete list can be found in Table 5.4.

Tables 5.3: Temporal features.

Level	Feature	Type
Frame	Pos. of frame in current segment (forwards)	N
	Pos. of frame in current segment (backwards)	N
Phoneme	No. of frames in current phoneme	N
Segment	No. of frames in current segment	N

Tables 5.4: Phonemes used in the Spanish system.

IPA	ASCII transcription
/a/	a
/b/,/β/	b
/tʃ/	C
/d/,/ð̞/	d
/e/	e
/f/	f
/g/	g
/ɲ/	h
/i/	i
/k/	k
/l/,/lʲ/,/l̪/	l
/m/	m
/n/,/nʲ/,/n̪/	n
/o/	o
/p/	p
/r/	R
/ɾ/	r
/ʃ/,/ʒ/,/ʂ/,/ʐ/	s
/t/	t
/u/	u
/x/	x
/j̰/	y
/θ/	z
-	SP
-	NIL

5.2.3 Acoustic models

Both acoustic models feature a 5ms frame step, with an audio frequency of 16000Hz. The number of cepstral coefficients used is 40. The linguistic information is the same for both models.

The **HMM** system is composed of 5-state, no-skip models with diagonal covariance matrices. A total of 1017 different grouping questions were used for the construction of the decision trees. The α parameter, which controls the number of nodes of those trees, was set to 1.0, while the number of EM iterations in each reestimation was set to 5. The training was performed with a modified version of the English demo training script featured in HTS website.

The best **DNN** system features 169 input neurons, four of which correspond to the temporal features described in Table 5.3, and the rest are linguistic features; 3 hidden layers with 512 neurons each and 127 output neurons (40 mel-cepstral coefficients, 1 lf0, 1 mvf, their time derivatives and the V/UV bit). The pretraining was performed for an epoch each time a layer was added, with a learning rate of 0.08, a batch size of 20 and no weight decay. The fine-tuning process lasted 5 epochs, with a learning

rate of $4e-05$, a batch size of 1600 and 0.003 weight decay.

5.3 English system

5.3.1 Data usage and preprocess

From the VL.NET corpus (Section 4.2), we extracted a subset of 25 videos where the lecturer is a single female native American English speaker. We applied the technique described in Section 4.2 to transform subtitles into transcriptions and then removed the segments with more than 35 points of Word Error Rate. Additionally, any segments with hesitation marks were removed. The rest of the segments where automatically aligned with the phonemes with the best English acoustic model deployed in the transLectures project at month 24 [6]. The final subcorpus statistics are collected in Table 5.5.

Tables 5.5: Statistics of the corpus for the English TTS system

Videos	25
Speakers	1
Hours	13 (w/o silences)
Segments	12791
Phonemes	586844

5.3.2 Linguistic analysis

We use the front-end module of Flite+hts_engine to perform linguistic analysis for the English system. The list of linguistic features given by this system can be seen in Table 5.6. The DNN CD-phonemes are augmented with the same temporal information as in the case of Spanish (see Table 5.3). The 40 phonemes used in the system and their IPA counterparts can be seen in Table 5.4. The *pau* and *x* phonemes are used to mark a silence and the beginning and end of a sentence, respectively.

5.3.3 Acoustic models

The acoustic models trained for English are similar to the ones for Spanish. In particular, the **HTS** system features 5-state, no-skip HMM models with diagonal covariance matrices. The 1483 grouping questions used are the same as the English demo of HTS. The α parameter was set to 1.0 and the number of EM iterations in each reestimation was set to 5.

The **DNN** system consists of 289 input neurons, 2 hidden layers of 1024 neurons and 127 neurons in the output layer. The pretraining was performed for an epoch each time a new layer was added, with a learning rate of 0.08, a batch size of 20 and no weight decay. The fine-tuning process lasted 5 epochs, with a learning rate of

$4e-05$, a batch size of 800 and 0.003 weight decay. The acoustic features generated by the model are the same as in Spanish.

5.4 Conclusions

In this chapter we have described extensively the inner workings of our TTS systems. We have explained the training and test stages, and detailed the differences between the Spanish and English systems. Although the English TTS task is considerably more complex than the Spanish, we also dispose of a higher volume of data and expand the CD-phonemes with additional linguistic information. The complete systems are fully functional and testing and evaluation studies are already being carried out (see Chapter 6).

Tables 5.6: Linguistic features of English system.

Level	Feature	Type*
Phoneme	Left-left phoneme identity	C
	Left (previous) phoneme identity	C
	Current phoneme identity	C
	Right (next) phoneme identity	C
	Right-right phoneme identity	C
	Position of the phoneme in the syllable (forward)	N
	Position of the phoneme in the syllable (backward)	N
Syllable	Is left syllable stressed?	B
	Is left syllable accented?	B
	No. of phonemes in left syllable	N
	Is current syllable stressed?	B
	Is current syllable accented?	B
	No. of phonemes in current syllable	N
	Pos. of current syllable in word (forward)	N
	Pos. of current syllable in word (backwards)	N
	Pos. of current syllable in segment (forwards)	N
	Pos. of current syllable in segment (backwards)	N
	No. of stressed syllables before current syllable	N
	No. of stressed syllables after current syllable	N
	No. of accented syllables before current syllable	N
	No. of accented syllables after current syllable	N
	No. of syllables from previous stressed syllable	N
	No. of syllables to next stressed syllable	N
	No. of syllables from previous accented syllable	N
	No. of syllables to next accented syllable	N
	Vowel in current syllable	C
	Is right syllable stressed?	B
	Is right syllable accented?	B
	No. of phonemes in right syllable	N
Word	Part-Of-Speech classification of left word	C
	No. of syllables in left word	N
	Part-Of-Speech classification of current word	C
	No. of syllables in current word	N
	Pos. of current word in segment (forward)	N
	Pos. of current word in segment (backwards)	N
	No. of content words before current word	N
	No. of content words after current word	N
	Pos. of current word in segment (forward)	N
	Pos. of current word in segment (backwards)	N
	No. of syllables from previous content word	N
	No. of syllables to next content word	N
	Part-Of-Speech classification of right word	C
	No. of syllables in right word	N
Segment	No. of syllables in current segment	N
	No. of words in current segment	N

* C=Categorical, B=Binary, N=Numeric

Tables 5.7: Phonemes used in the English system.

IPA	ASCII transcription	IPA	ASCII transcription
/ɒ/	aa	/l/	l
/æ/	ae	/m/	m
/ʌ/	ah	/n/	n
/ɑ/	ao	/ŋ/	ng
/au/	aw	/o/	ow
/ə/	ax	/ɪ̈ɪ/	oy
/aɪ/	ay	/p/	p
/b/	b	/ɹ/	r
/tʃ/	ch	/s/	s
/d/	d	/ʃ/	sh
/ð/	dh	/t/	t
/ɛ/	eh	/θ/	th
/ɜ˞/,/ə˞/	er	/ʊ/	uh
/eɪ/	ey	/u/	uw
/f/	f	/v/	v
/g/	g	/w/,/ʍ/	w
/h/	hh	/j/	y
/ɪ/	ih	/z/	z
/i/	iy	/ʒ/	zh
/dʒ/	jh	-	pau
/k/	k	-	x

Chapter 6

Evaluation and integration

In this Chapter we describe the objective evaluation that has been performed in order to compare the HMM and DNN-based approaches to acoustic mapping. Then we show the integration carried out in the transLectures player to allow a user to listen to the synthesized voice.

6.1 Evaluation

We performed objective evaluation of the resulting voice for the HMM and DNN-based TTS systems. We performed this comparison for **Spanish**, as the English system is much more computationally expensive to train (specially in the case of HTS), which limited the parameter tuning and prevented a fair comparison.

6.1.1 Experimental setup

The experimental setup parameters for the experimentation are very similar to the ones reported in Section 5.2.3. In particular, the training was performed with the data described in . For testing, 49 utterances from a video that was not contained in the training corpus were extracted and synthesized. Phoneme durations were set to match those from the natural speech, rather than being generated by a duration model.

For training purposes, audio was extracted from the video and downsampled from 44100Hz to 16000Hz. Every 5 milliseconds, 40 Mel-cepstral coefficients, log F_0 and maximum voiced frequency values were extracted using AhoCoder tools [1]. The *mvf* feature was interpolated in the unvoiced regions for both models, while the log F_0 was interpolated for the DNN explicit voicing. The acoustic feature vectors were then augmented with the information of the first and second derivatives. The textual analysis information was the same for both models.

The HMM system was composed of 5-state, no-skip models with diagonal covariance matrices. A total of 1017 different questions were used for the construction of the decision trees. For comparison purposes, we trained 6 HMM-based systems modifying the parameter α which controls the number of nodes of the decision trees (with $\alpha = 0.5$, 1.0 and 2.0), and the number of EM iterations in each reestimation step (3 or 5). The training was performed using the most recent stable version (2.2) of the HTS system (See Section 2.3.1).

In the case of the DNN-based system, the number of neurons in the input layer was 169, while the number of neurons in the output layer was 127, corresponding to 39 *mfcc* plus energy, log F_0, *mvf*, first and second derivatives and the V/UV bit. Inputs to the DNN were normalized to have zero mean and one variance, while outputs were normalized between 0.01 and 0.99. Different neural network sizes were tested by changing the number of hidden layers (1, 2, 3 or 4) and the number of neurons per layer (128, 256, 512 or 1024).

The sigmoid activation function was used in the hidden and output layers. Neural networks with more than one hidden layer were pretrained using a discriminative approach [34], and then fine-tuned with a stochastic minibatch backpropagation algorithm [10]. The error criterion in both steps was the mean squared error (MSE). The training was performed with a CUDA-based GPU implementation, part of a development version of the transLecture stoolkit.

6.1.2 Results and Discussion

Table 6.1 shows the objective evaluation measures computed for each DNN configuration, together with the results of the best HMM model. Regarding to the DNN configurations, the number of neurons per layer did not contribute as significantly as the number of layers, so only the best result is reported. We can see that DNN-based systems systematically achieve better results in every measure than HMM-based systems. The optimal number of layers is unclear, since the evaluation measures exhibit different behaviour. The V/UV error rate performs better when using simpler architectures, while the spectral features benefit more from a complex architecture.

Tables 6.1: Comparison between HMM-based and DNN-based acoustic models.

System	# layers	RMSE log F_0	MMCD	V/UV Error rate
HMM	-	0.190	6.987	13.35
DNN	1	0.183	6.792	12.08
	2	0.183	6.702	12.27
	3	0.184	6.678	12.36
	4	0.184	6.679	12.42

6.2 Integration

We have successfully integrated a TTS track selector into the transLectures player/editor. If available, the user is presented with a voice icon (see Figure 6.1) that switches between the audio embedded into the video and an external track that contains a synthesized voice. We have also started to synthesize a small subset of lectures from poliMedia repository that have been automatically transcribed and later supervised by an expert. The current implementation does not allow the user to select a track, as only English → Spanish and Spanish → English lectures have been synthesized.

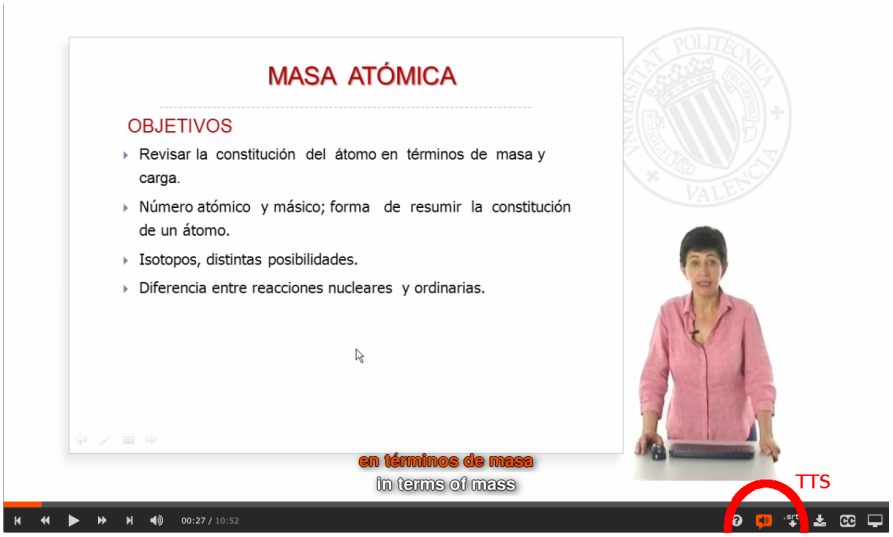


Figure 6.1: A screenshot of the current player with TTS playback capabilities. TTS button being orange shows that the synthesized track is currently being played.

6.3 Conclusions

We have performed an objective comparison of the acoustic models for Spanish. The comparison shows that the DNN-based approach is able to reconstruct the audio wave more accurately. We have also taken the first step towards a subjective evaluation, by integrating the TTS synthesized voice into our video lecture player and synthesizing a small set of lectures with supervised subtitles. Future work will include private and public evaluations by experts and potential users alike.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have tackled the problem of synthesizing subtitles of a video lecture. We have reviewed the speech synthesis state-of-the-art and decided to develop a TTS system which makes use of the statistical parametrical speech synthesis framework. We have analysed the problem of mapping linguistic features to acoustic features extensively, and presented two different approaches, HMM-based and DNN-based. We have developed 2 systems for Spanish and English which can use both HMM and DNN acoustic models. We have performed objective comparison of the voices generated with the Spanish acoustic models. Finally, we have modified our lecture player to include track selection capabilities, allowing users to listen to the synthesized track.

7.2 Future work

Our next planned steps include starting subjective evaluation of the synthesized voices for both English and Spanish, as well as integrating the synthesizer into the transLectures platform. We will be working closely with poliMedia maintainers to study the possibility of producing and providing TTS tracks for all or part of the repository. We hope the results of this work can be used to increase accessibility to their excellent educational multimedia content, opening up an array of exciting possibilities.

Additionally, the statistical parametrical synthesis framework opens up many opportunities for improvement. We intend to implement some of the latest advances in TTS with NN, like Deep Mixture Netwoks [53], and carry over the findings in ASR to the TTS systems, such as Recurrent NNs with bidirectional LSTM architecture [14] and DNN adaptation to the speaker [49]. We look forward to continue improving the

voice naturalness and intelligibility.

7.3 Contributions

The scientific publications related to this work are listed below.

- *S. Piqueras, M. A. del-Agua, A. Giménez, J. Civera, and A. Juan* **Statistical text-to-speech synthesis of Spanish subtitles.** IberSPEECH 2014. Submitted.

7.4 Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 287755 (transLectures) and ICT Policy Support Programme (ICT PSP/2007-2013) as part of the Competitiveness and Innovation Framework Programme (CIP) under grant agreement no 621030 (EMMA), and the Spanish MINECO Active2Trans (TIN2012-31723) research project.

Bibliography

[1] Ahocoder. http://aholab.ehu.es/ahocoder.

[2] Coursera. https://www.coursera.org.

[3] Flite+hts_engine, Version 1.05. http://hts-engine.sourceforge.net/.

[4] HMM-Based Speech Synthesis System (HTS). http://hts.sp.nitech.ac.jp.

[5] Khan Academy. https://www.khanacademy.org.

[6] transLectures: Second report on massive adaptation. http://www.translectures.eu/wp-content/uploads/2014/01/transLectures-D3.1.2-15Nov2013.pdf.

[7] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2004.

[8] Ebru Arisoy, Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28. Association for Computational Linguistics, 2012.

[9] Vlad Stefan Barbu and Nikolaos Limnios. Semi-markov chains and hidden semi-markov models toward applications. *Their Use in Reliability and DNA Analysis*, 191, 2008.

[10] Léon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France, 1991. EC2.

[11] Min Chu and Hu Peng. Objective measure for estimating mean opinion score of synthesized speech, April 4 2006. US Patent 7,024,362.

[12] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.

[13] D. Erro, I Sainz, E. Navas, and I Hernaez. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):184–194, 2014.

[14] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[18] Qiong Hu, Korin Richmond, Junichi Yamagishi, and Javier Latorre. An experimental comparison of multiple vocoder types. In *8th ISCA Workshop on Speech Synthesis, Barcelona, Spain*, pages 135–140, 2013.

[19] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997.

[20] Orhan Karaali. Speech synthesis with neural networks orhan karaali, gerald corrigan, and ira gerson. In *Proceedings of the 1996 World Congress on Neural Networks*, page 45. Psychology Press, 1996.

[21] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigné. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech communication*, 27(3):187–207, 1999.

[22] Simon King. Measuring a decade of progress in text-to-speech. *Loquens*, 1(1):e006, 2014.

[23] Simon King and Vasilis Karaiskos. The blizzard challenge 2013. In *Proceedings of the Blizzard Challenge Workshop*, 2013.

[24] John Kominek, Tanja Schultz, and Alan W Black. Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion. In *Proc. of SLTU*, pages 63–68, 2008.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[26] Javier Latorre, Mark JF Gales, Sabine Buchholz, Kate Knill, M Tamurd, Yamato Ohtani, and Masami Akamine. Continuous f0 in the source-excitation generation for hmm-based tts: Do we need voiced/unvoiced classification? In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4724–4727. IEEE, 2011.

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[28] Yuji Nakano, Makoto Tachibana, Junichi Yamagishi, and Takao Kobayashi. Constrained structural maximum a posteriori linear regression for average-voice-based speech synthesis. In *INTERSPEECH*, 2006.

[29] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, 2008.

[30] poliMedia. The polimedia repository, 2007.

[31] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *ICML*, volume 9, pages 873–880, 2009.

[32] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

[33] Flávio Ribeiro, Dinei Florêncio, Cha Zhang, and Michael Seltzer. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2416–2419. IEEE, 2011.

[34] Frank Seide, Gang Li, Xie Chen, and Dong Yu. Feature engineering in context-dependent dnn for conversational speech transcription. In *Proc. of ASRU*, pages 24–29, 2011.

[35] K Shinoda and T Watanabe. MDL-based context-dependent subword modeling for speech recognition. *Journal of The Acoustical Society of Japan*, 21(2):79–86, 2000.

[36] Joan Albert Silvestre, Miguel del Agua, Gonçal Garcés, Guillem Gascó, Adrià Giménez-Pastor, Adrià Martínez, Alejandro Pérez González de Martos, Isaías Sánchez, Nicolás Serrano Martínez-Santos, Rachel Spencer, Juan Daniel Valor Miró, Jesús Andrés-Ferrer, Jorge Civera, Alberto Sanchís, and Alfons Juan. translectures. In *Proceedings of IberSPEECH 2012*, 2012.

[37] SPTK Working Group. Speech signal processing toolkit (sptk), version 3.7. `http://sp-tk.sourceforge.net/`, 2013.

[38] Ralf Steinmetz. Human perception of jitter and media synchronization. *Selected Areas in Communications, IEEE Journal on*, 14(1):61–72, 1996.

[39] Rob Sykes, Pascal Giard, Chris Bagwell, et al. Sox-sound exchange, version 14.3.2. 2011.

[40] The transLectures-UPV Team. The transLectures Platform (TLP). `http://translectures.eu/tlp`.

[41] Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai. Speech parameter generation from hmm using dynamic features. In *Proc. of ICASSP*, volume 1, pages 660–663, 1995.

[42] Keiichi Tokuda, Takashi Masuko, Noboru Miyazaki, and Takao Kobayashi. Multi-space probability distribution HMM. *IEICE Transactions on Information and Systems*, 85(3):455–464, 2002.

[43] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1315–1318. IEEE, 2000.

[44] transLectures. D2.2: Status report on complete transcriptions and translations.

[45] transLectures. D3.1.3: Final report on massive adaptation. To be published.

[46] Videolectures.NET: Exchange ideas and share knowledge. `http://www.videolectures.net/`.

[47] Yi-Jian Wu, Yoshihiko Nankaku, and Keiichi Tokuda. State mapping based method for cross-lingual speaker adaptation in hmm-based speech synthesis. In *Interspeech*, pages 528–531, 2009.

[48] Junichi Yamagishi. An introduction to HMM-based speech synthesis. Technical report, Centre for Speech Technology Research, 2006.

[49] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong. Adaptation of context-dependent deep neural networks for automatic speech recognition. In *SLT*, pages 366–369, 2012.

[50] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Duration modeling for hmm-based speech synthesis. In *ICSLP*, volume 98, pages 29–31, 1998.

[51] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Incorporating a mixed excitation model and postfilter into hmm-based text-to-speech synthesis. *Systems and Computers in Japan*, 36(12):43–50, 2005.

[52] Kai Yu and Steve Young. Continuous f0 modeling for hmm based statistical parametric speech synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(5):1071–1079, 2011.

[53] Heiga Zen and Andrew Senior. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3872–3876, 2014.

[54] Heiga Zen, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *Proc. of ICASSP*, pages 7962–7966, 2013.

[55] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

[56] Heiga Zen, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Hidden semi-markov model based speech synthesis. In *INTERSPEECH*, 2004.

List of Figures

2.1 A simple source-filter decoder 7

3.1 A generic machine learning system for regression 12
3.2 A simple HMM with 3 states (not counting I and F) and 2 possible emission values "a" and "b" 13
3.3 A sample of part of a binary decision tree for the first state of the cepstral coefficients of the Spanish HMM system. Notice that most of the decisions depend on the left phoneme (L-*), which reveals strong temporal dependency between adjacent phonemes. 14
3.4 A shallow neural network 16
3.5 A deep feed-forward neural network for speech synthesis. 17

4.1 A video lecture with subtitles in the paella player 20
4.2 A video lecture from VL.NET with subtitles 21

5.1 Overview of the training and synthesis processes 27

6.1 A screenshot of the current player with TTS playback capabilities. TTS button being orange shows that the synthesized track is currently being played. 37

Index of Tables

4.1 Statistics of the poliMedia repository 19
4.2 Statistics of the poliMedia corpus 20
4.3 Statistics of the Videolectures.NET repository 21
4.4 Statistics of the VL.NET corpus 22

5.1 Statistics of the corpus for the Spanish TTS system 28
5.2 Linguistic features of Spanish system. 29
5.3 Temporal features. 30
5.4 Phonemes used in the Spanish system. 30
5.5 Statistics of the corpus for the English TTS system 31
5.6 Linguistic features of English system. 33
5.7 Phonemes used in the English system. 34

6.1 Comparison between HMM-based and DNN-based acoustic models. . 36