PARAMETRICAL SPEECH SYNTHESIS

SÉBASTIEN LE MAGUER

ADAPT CENTRE, SIGMEDIA LAB, EE ENGINEERING, TRINITY COLLEGE DUBLIN

JULY 8TH, 2020
A couple of rules

- Asking a question
  1. Raise your hand
  2. No acknowledgement at the end of the slide ⇒ unmute and tell me :)

- Breaks
  - About every 30min ⇒ two breaks

- Slides
  - Accessible after the lecture

Some surveys

- Anonymous
- Not recorded
- Idea: having fun
Speech synthesis [4/4]
**WHAT IS IT?**

**Speech synthesis**
Producing speech from something

**Text-to-Speech synthesis (TTS)**

*Text*

Once upon a time...
Speech synthesis
Producing speech from something

Text-to-Speech synthesis (TTS)

Text
Linguistic prediction
Acoustic prediction

Once upon a time...

start-oi-nn
oi-nn-ss
nn-ss-eu
ss-eu-pp
...
Why?

- Dubbing
- Video game
- Story telling
- Simple interact.
- Inter. vocal server
- Telepresence
- Reading assisting
- Expr. assisting
**The architecture of lecture**

- Follow the history
- Presenting problems...
- ... **over-viewing** some solutions

**Your mission**

- **Understand** the problems (and why we have to face them)
- **Understand** the key idea behind the solution(s)
CORPUS BASED TTS [6/6]
GLOBAL ARCHITECTURE

Offline stage

Corpus

speech

text

A. c. param. extraction

Acoustic param.

Desc. feat. extraction

Labels

Training stage

Models

Par. to Speech

Text to synth

Desc. feat. extraction

Labels

Generation stage

Online stage
Text and signal aligned!

- Need of a defined acoustic unit (phone, diphone, . . . )
- Need of an equivalent in the linguistic area (phoneme, diphoneme, . . . )
- **Alignment = linking the acoustic unit to the linguistic unit**

Biases

- The corpus will conditioned strongly the results!
- The tools as well!
The front-end - 1

Goal
"Extract" a symbolic description of the text

Following a standard pipeline
<table>
<thead>
<tr>
<th><strong>THE FRONT-END - 2</strong></th>
</tr>
</thead>
</table>

### Text normalization
Transform non-grapheme (numbers, dates, ...) to their extended orthographic form

### Part-of-speech tagging
Associate to each word their grammatical categories

### Prosodic related modules
- Syllabification (breaks and stressing information)
- ToBI (break and intonation prediction from the syntaxe)

### Letter-to-Sound (or Grapheme-to-Phoneme)
Get the phonetic representation from the text
### Implementations

- Rule based
- Statistical
- + user dictionaries

### Some open source systems

- Festival ([https://github.com/festvox/festival](https://github.com/festvox/festival))
- MaryTTS ([https://github.com/marytts/marytts](https://github.com/marytts/marytts))
ACOUSTIC REPRESENTATION - SOURCE FILTER

Source

Filter

F0

Periodic signal generation

White noise generation

Spectrum Parameters

Vocal tract

speech signal

\( e(t) \)

\( s(t) \)
**MEL SCALE**

**Figure:** HTKBook figure [S. Young et al., (2005)]
SPECTROGRAM VS MEL-SPECTROGRAM
**Mel Cepstrum**

- **Decorrelation**
  - First coefficient = gain!

- **Process**

  - Can be regenerated based on [Fukada et al., (1992)]
A NEW PARADIGM

Fundamental question

What can we do except using the raw signal?
A NEW PARADIGM

Fundamental question
What can we do except using the raw signal?

Foundation of parametrical synthesis
Accept to lose some information to have a better control
### Question

Which one of the two samples is the result of parametrical speech synthesis?

### Samples

- **Sample A:** [Play Sound](#)
- **Sample B:** [Play Sound](#)
<table>
<thead>
<tr>
<th>Unit Selection</th>
<th>Parametrical synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use of the signal</strong></td>
<td><strong>Rendering the signal from ac. params.</strong></td>
</tr>
<tr>
<td>High memory footprint (gb)</td>
<td>Small memory footprint (kb/mb)</td>
</tr>
<tr>
<td>Corpus design crucial</td>
<td>Corpus design important but some errors can be processed</td>
</tr>
<tr>
<td>Rigidity</td>
<td>Flexibility</td>
</tr>
<tr>
<td>High quality (with some glitches)</td>
<td>Muffled/Buzzy speech</td>
</tr>
</tbody>
</table>
ACOUSTIC REPRESENTATION - SOURCE FILTER

Source

Filter

F0

Periodic signal
generation

White noise
generation

Spectrum Parameters

Vocal tract

e(t)

s(t)

Speech
signal
ACOUSTIC REPRESENTATION - MIXED-MODE SOURCE FILTER

- Periodic signal generation
- White noise generation
- Bandpass Filter
- Spectrum Parameters
- Vocal tract
- Speech signal

Source

Filter
Mel Log Spectrum Approximation (MLSA) [Imai et al., (1983)] [Fukada et al., (1992)]

- Reconstruct more accurate the filter parameters from the MFCC
- Accuracy: maximum spectral error 0.24dB
- Stability guaranteed

Vocoders

- STRAIGHT [Kawahara et al., (1999)] : long time state for the art (matlab)
- WORLD [Morise et al., (2016)] : open source (mainly used currently) (C)
See [Masuko et al., (1996)]

\[ O = W \cdot C \]  \hspace{1cm} (1)

- \( C \) = extracted coefficients
- \( W \) = a windowing matrix
- \( O \) = observations
Figure: based on [Masuko et al., (1996)]
**Inputs and Outputs!**

<table>
<thead>
<tr>
<th>Inputs (from the front-end!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linguistic descriptive features:</td>
</tr>
<tr>
<td>- phonetic</td>
</tr>
<tr>
<td>- position</td>
</tr>
<tr>
<td>- prosodic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic parameters:</td>
</tr>
<tr>
<td>- MGC</td>
</tr>
<tr>
<td>- BAP</td>
</tr>
<tr>
<td>- LF0</td>
</tr>
</tbody>
</table>
HMM-BASED SYNTHESIS [11/11]
Before we start

Speech synthesis / Speech recognition
Lots of concept developed for speech recognition adapted for speech synthesis

For this lecture
Focus only on the baseline system presented in [Zen and Toda, (2005)]
WHAT ARE WE TRYING TO SOLVE - 1

Offline stage

- Corpus
  - speech
  - text
- Acoustic param. extraction
- Desc. feat. extraction
- Labels
- Training stage
- Models
- Par. to Speech

Online stage

- Text to synth
- Desc. feat. extraction
- Labels
- Generation stage
What are we trying to solve - 2

Linguistic scale (variable)

frame rate (fixed)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>pau</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>pau</td>
<td>ao</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ax</td>
<td>pau</td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.02</td>
<td></td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>-9</td>
<td>0.7</td>
<td></td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>0.9</td>
<td></td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.02</td>
<td></td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>-7</td>
<td>0.1</td>
<td></td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.02</td>
<td></td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>
WHAT ARE THE DIFFERENT PROBLEMS

Abstraction
- how to get "knowledge" from the data

Parameter representation
- How to deal with heterogeneous data

Sequence mapping
- How to align sequence of different temporally (phones vs frames)

Sparseness
- How to deal with the fact that the corpus doesn’t contain all possible instances of speech
Gaussian Mixture Model: \[ \sum_{i=1}^{N} \omega_i \mathcal{N}(\mu_i, \Sigma_i) \]
**PARAMETER REPRESENTATION**

- **MSD** = **Multi-Space** Distribution [Tokuda et al., (2000a)]

**Definition**

\[ X = (V, S) \]  \( (2) \)

- **S** = a space
- **V** = a value in the space **S**

**Application for the F0**

\[ MSD(X) = \begin{cases} 
  w_1 N(V(X); \mu, \Sigma), & S(X) = \{1\} \text{ (voiced)} \\
  w_2 \delta_0(V(X)), & S(X) = \{0\} \text{ (unvoiced)} 
\end{cases} \]  \( (3) \)
Speech related adaptation:
- Time constraint ⇒ left-to-right without skip
- Input unit = phone
  - ⇒ 1 HMM per phone
  - ⇒ utterance = N phones = N HMM
- Observable phenomenon = acoustic
- Hidden phenomenon = Articulatory configuration

⇒ the HMM is used as a clock synchronization

References
See lecture 7
HMM = Geometric distribution

\[
Pr(X = k) = (1 - p)^{k-1} p \quad (4)
\]

**PROBLEM:**
- Optimization during synthesis ⇒ 1 frame/state!

**Solution:**
- Gaussian distribution (synth.) [Yoshimura et al., (1998)]
- HSMM (train.) [Russell and Moore, (1985); Zen et al., (2004)]
Input features

- Generally based on linguistic informations...
- ....but you can use what you want
- Only discrete values (booleans, numbers, enumerations, ...)

Main constraint

Have to be available at the synthesis stage
**Problematic**
- English = 53 descriptive features, German = 70 descriptive features!
- ≠ statistical training

**Solution** use a decision tree [S. J. Young et al., (1994)]

Iterative algorithm ⇒
- Criterion (Stop + Split)= Minimum Description Length (based on variance) [Shinoda and Wanabe, (2000)]
PARAMETER GENERATION ALGORITHM [TODA AND TOKUDA, (2005); TOKUDA ET AL., (2000b)]

start-tt-an ... an-tt-end
PARAMETER GENERATION ALGORITHM [TODA AND TOKUDA, (2005); TOKUDA ET AL., (2000b)]

start-tt-an  ...  an-tt-end

Nb segments
PARAMETER GENERATION ALGORITHM [TODA AND TOKUDA, (2005); TOKUDA ET AL., (2000b)]

Diagram showing transitions between states such as start-tt-an, ... , and an-tt-end, with decision points for L-start?, C-Plosive?, N-Vowel?, C-Plosive?, and N-Voiced?.
PARAMETER GENERATION ALGORITHM [TODA AND T. OKUDA, (2005); T. OKUDA ET AL., (2000b)]
PARAMETER GENERATION ALGORITHM [TODA AND TOKUDA, (2005); TOKUDA ET AL., (2000b)]

start-tt-an ... an-tt-end
PARAMETER GENERATION ALGORITHM [TODA AND TOKUDA, (2005); TOKUDA ET AL., (2000b)]

start-tt-an    ...    an-tt-end

[Diagram of a parameter generation algorithm with start-tt-an and an-tt-end as key points.]

[Waveform diagrams showing the progression of signals through the parameter generation process.]
DNN-based synthesis [11/11]
The main problems with HMM synthesis

The abstraction limits: decision tree limits

**Discrete representation**
- Continuous $\Rightarrow$ quantization
- Accurate scale?

**Strong segmentation**
- boundary are *strict* $\Rightarrow$ you are in one cluster!
- non-linear data?

**Scaling issues:**
- some features doesn’t really impact the models
**Fundation of Neural Network - What is a Neuron?**

**Figure:** image taken from Wikipedia

which leads to the formula:

\[ o = \varphi \left( \sum_{k=0}^{n} w_k x_k \right) \]  

(5)
ARTIFICIAL NEURAL NETWORK
WHAT IS A DNN
Why using DNN

- It scales!

- A lot of toolkit and libraries are available
- It is easy to code (but debugging is another story...)
APPLICATION TO TTS [ZEN ET AL., (2013)]

- **Binary features**
- **Numeric features**
- **Duration feature**
- **Frame position feature**

**Input layer**
- Input features including binary & numeric features at frame 1
- Input features including binary & numeric features at frame T

**Hidden layers**
- Text analysis
- Statistics (mean & var) of speech parameter vector sequence

**Output layer**
- Duration prediction
- Input feature extraction
- Parameter generation

**SPEECH**
- Waveform synthesis

**SPECTRAL features**
- Excitation features
- V/UV feature
HMM input features
- Discrete
- Binary/class/discrete values
- Set-based analysis

DNN input features
- Only binary or numerical!
- Can be continuous

How can we convert to DNN compatible
- position features:
- class features:

⇒ This process is named: normalization
THE DURATION PROBLEM

HMM

- subphone = state

DNN First generation

1. HMM predict the duration [Zen et al., (2013)]
2. Associate the state id (duration feature) + frame id (frame position feature)

DNN Current generation

- train a dedicated duration DNN on the forced alignment
- A nice paper [Henter et al., (2016)]

But Why ???

- still the same reason:

  phone is not an ideal unit to capture trajectories
**Input:** Same label files than for the HMM

**Process:**
1. transform discrete values to numerical equivalent or hot vector
2. Predict the sub-unit id (state from hmm, ...) and append this value to the vector
3. Predict the frame id (synthesis and get the duration from HMM) and append this value to the vector

**Result**
Speech is about trajectories ⇒ Recurrent Neural Network
Solution: Introduce a specific unit (LSTM, GRU, …)
  ▶️ we actually don’t need everything ⇒ forget!
  ▶️ Example of an LSTM architecture (multiple variations exists):

Further reading here:
https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714
HOW DOES IT SOUND

Samples

- Sample A: Play Sound
- Sample B: Play Sound
End-to-end synthesis [6/6]
MAIN LOGIC

Starting point
DNN are pretty powerful and scale well

Key question
Can we predict the speech directly from the text using one big DNN?

Historically baseline sequence
- 2016: Wavenet \(\Rightarrow\) neural vocoders
- 2017: Tacotron \(\Rightarrow\) generating directly from the graphemes
Predict directly the raw samples!

- First version: input = phoneme + f0
- Now: used a neural vocoder

Simplification hypotheses

- Application of a $\mu$-law
- Quantization at 256 values
**WAVENET - 2**

**Principle**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input (dil=1)</td>
<td></td>
</tr>
<tr>
<td>Hidden (dil=2)</td>
<td></td>
</tr>
<tr>
<td>Hidden (dil=4)</td>
<td></td>
</tr>
<tr>
<td>Output (dil=8)</td>
<td></td>
</tr>
</tbody>
</table>

**Samples**

Play Sound (A) vs Play Sound (B) vs Play Sound (C)
Principle (fig. from [Shen et al., (2018)])

Samples

Play Sound (A) vs Play Sound (B)
IS IT END-TO-END?

(See Simon King’s talk)
Speech synthesis workshop statistics ([Lozo et al., (2019)])

Industry

- Tacotron 2, Alexa, Deep voice 3, Siri, Cortana, …


Lozo, Carina et al. (Sept. 2019). “The thought collective behind thirty years of progress in speech synthesis”. In:


<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>In:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young, Steve et al.</td>
<td>The HTK Book</td>
<td>July 2000.</td>
</tr>
</tbody>
</table>