PART V
Text-to-Speech Synthesis (TTS)

TTS: What for?

- Telephone-based applications
  - Telecommunications ($) (Telecommunications ($)
    - Who’s calling
    - Integrated messaging (fax, email, answering machine)
    - Automatic reverse directory
    - Personal telephone attendant
  - Voice access to databases (70% of calls require very little interactivity)
    - Price lists
    - Cultural events
    - Weather report

- Multimedia
  - CDRoms
  - Talking books
  - Interactive games

- Man-machine communication
TTS: What for?

- Help to the disabled
  - Speech impairment
    - Artificial voice
  - Sight impairment
    - Automatic reading of electronic documents
    - Automatic reading of paper documents (with OCR)

TTS = NLP + DSP

(a) Phonetization
(b) Intonation/Duration Generation
(c) Speech Synthesis

To be or not to be, that is the question.

Automatic phonetization

- Dictionary look-up?

To be or not to be, that is the question.

Be b i:
Not n Q t
Or O r
Question k w e s t S @ n
That D { t
The D @
To t U
's s
Automatic phonetization

- More complex than that!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Example</th>
<th>Level</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assimilation</td>
<td>nasality or sonority</td>
<td>word/sentence</td>
<td>pronunciation of neighbors</td>
</tr>
<tr>
<td></td>
<td>assimilation, vocalic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>harmonization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heterophonic</td>
<td>the, record, contrast</td>
<td>word</td>
<td>part-of-speech, meaning (rare)</td>
</tr>
<tr>
<td>homographs</td>
<td>read, est, couvant, portions, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schwa</td>
<td>deletion</td>
<td>sentence</td>
<td>syntactic articulation,</td>
</tr>
<tr>
<td>deletion</td>
<td>table rouge, je ne te le redirai pas</td>
<td></td>
<td>pronunciation of neighbors,</td>
</tr>
<tr>
<td>Phonetic</td>
<td>très utile, deux à</td>
<td>sentence</td>
<td>speaking style</td>
</tr>
<tr>
<td>liaisons</td>
<td>deux, plat exquis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New words</td>
<td>proopiomelancortin</td>
<td>word</td>
<td>spelling</td>
</tr>
<tr>
<td>Proper names</td>
<td>your name here ...</td>
<td>word</td>
<td>morphology, analogy</td>
</tr>
</tbody>
</table>

Intonation

- Why ups and downs?
  - Stress (word level) → Accent (phrase level) (Phonetization↑)
  - Modify slightly → unnatural!

- (a,b) Focus   (c) Finality/continuity
- (d) Grouping, using phrase-level accent

Phoneme Duration

- Not constant
- Not fixed for a given phoneme
- Linked to intonation (longer on accented syllables)
Intonation/Duration

- Prosody relies on each and every level of linguistic competence of the reader:
  - Syntax: Mainly
    "Twas brillig, and the slithy toves Did gyre and gimble in the wabe
    All mims were the borogroves, And the mome raths outgrabe.
    Lewis Carroll, Jabberwocky
  - Semantics:
    "Time flies like an arrow"
  - Pragmatics: Personal implication of the reader in his reading
    'John did not come to New-York by plane'
    = emphasis, contrastive stress, speaking style...
  "Acceptable neutral prosody?"

Challenges: a summary

- Intelligible – Natural – Cost effective
  - Accurate automatic phonetization (=dictionary look-up)
  - Prosody generation (i.e., intonation and phoneme durations) must be "coherent"; easy to produce unnatural prosody
  - Synthesis of phoneme sequences with corresponding prosody
    - Coarticulation!
    - Segmental quality should be maintained after pitch and duration modification
  - Engineering
    - Low design and maintenance cost
    - Low computational and memory cost
    - Easy adaptation to other languages

Contents

- Introduction
- Acoustic speech synthesis (DSP)
  - Model-based (rule-based) approach
  - Instance-based (concatenative) approach
- From text to phonemes and prosody (NLP)
  - Preprocessing
  - Morpho-syntactic analysis
  - Phonetization
  - Prosody Generation

Coarticulation !!!

- Synthesis: be able to mimic coarticulation!
- Recognition: be able to overcome it!
Von Kempelen’s talking machine (1791)

Omer Dudley’s Voder (Bell Labs, 1936)

1. John Holmes’ formant synthesizer (1964)

Rule-based Synthesis

1. John Holmes’ formant synthesizer (1964)
Contents

- Introduction
- Acoustic speech synthesis (DSP)
  - Model-based (rule-based) approach
  - Instance-based (concatenative) approach
- From text to phonemes and prosody (NLP)
  - Preprocessing
  - Morpho-syntactic analysis
  - Phonetization
  - Prosody Generation

2. Diphone concatenation (1977)

Joe Olive’s LPC synthesizer (1977)
Christian Hamon’s PSOLA (1988)

The MBROLA project

T. Dutoit’s MBROLA (1993)

- Based on the same Poisson’s sum formula as PSOLA, but using edited diphones
- Similar overall quality as PSOLA
- Same computational load
- Completely automatic!
  ⇒ can be used to create lots of compatible synthesizers

J’ai été conçu...
Ma voix...


Diphone-based synthesis

<table>
<thead>
<tr>
<th>Smooth joints</th>
<th>50ms</th>
<th>80ms</th>
<th>160ms</th>
<th>70ms</th>
<th>50ms</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>d</em></td>
<td></td>
<td><em>d</em></td>
<td></td>
<td><em>o</em></td>
<td></td>
</tr>
<tr>
<td><em>o</em></td>
<td></td>
<td><em>d</em></td>
<td></td>
<td><em>o</em></td>
<td><em>g</em></td>
</tr>
<tr>
<td><em>g</em></td>
<td></td>
<td><em>g</em></td>
<td></td>
<td><em>g</em></td>
<td><em>g</em></td>
</tr>
</tbody>
</table>

How to get the best sequence of units for a given utterance? **Viterbi search**

- **Target cost?**
  How to predict which units will sound as they would naturally connected? (should be perceptual)

- **Concatenation cost?**
  How to predict which sequences of units will sound naturally connected? (should be perceptual)
3. Automatic unit selection

Very Large corpus

Unit i-1

 Concatenation cost cc(i-1,i)

Unit i

=0 in case of successive units

sent : "... to bear." sent : "... to bear."
phonet : t U b E@ ... phonet : t U b E@ ...
stress : ^ ... stress : ^ ...
tone : l L ... tone : l L ...
dur : 150 50 85 90 150 ... dur : 150 50 85 90 150 ...
f0 : ... f0 : ...
Formants: Formants

Contents

- Introduction
- Acoustic speech synthesis (DSP)
  - Model-based (rule-based) approach
  - Instance-based (concatenative) approach
- From text to phonemes and prosody (NLP)
  - Preprocessing
  - Morpho-syntactic analysis
  - Phonetization
  - Prosody Generation

From text to phonetics

1. Pre-processing

- Text segmentation into broad segmentation units
- From broad to final segmentation units
  - Sentence end detection
  - Dealing with abbreviations
  - Recognizing acronyms
  - Processing numbers

Simple Regular grammars do most of the job
(Lex – FSTs)
2. Morphological analysis

- Why?
  - Constrain the size of lexicons
  - Morphological features for syntactic processing
  - Morphologically related pronunciation
    - *Nebonstrasse*, *Abb*/*Ab*, *demonstrat*/*demonstrat*/*hathouze*, ...
  - Word-level stress in free stress languages
- How?
  - morphology = highly language dependent
    - English verbs: four to eight forms; French verbs: 37 to 41 forms; compounding much more complex in genrico languages: *hottentottentottentontongtongtontontontontontontontontontontontonton*
  - Typically: regular rules, finite state automata, organized in a language-dependent way

Increasingly: use of graphotactic trained systems
(ex: TNT http://www.coli.uni-sb.de/~thorsten/tnt/)

Or even: brute force: inflected dictionary

3. Contextual analysis: n-grams

Sentence \( \mathbf{W} = (w_1, w_2, \ldots, w_N) \)

All possible sequences \( \mathbf{T} = (t_1, t_2, \ldots, t_N) \)

\( \Rightarrow \) Best sequence of tags \( \hat{T} = \arg \max_T P(T|W) \)

\[
\hat{T} = \arg \max_T \frac{P(\mathbf{W}|T) P(T)}{P(\mathbf{W})} \quad \text{(Bayes)}
\]

\[
P(\mathbf{W}|T) = P(w_1, w_2, \ldots, w_N | t_1, t_2, \ldots, t_N) = P(w_i | t_i) \cdot P(w_{i+1} | t_i, t_{i+1}) P(w_{i+2} | w_i, w_{i+1}, t_i, t_{i+1}) \ldots
\]

\[
= \prod_{i=1}^{N} P(w_i | t_i) \quad \text{(Strong hypothesis 1)}
\]

\[
P(T) = P(t_1, t_2, \ldots, t_N) = P(t_1) \cdot P(t_2 | t_1) \cdot P(t_3 | t_1, t_2) \ldots
\]

\[
= \prod_{i=1}^{N} P(t_i | t_{i-1}, t_{i-2}, \ldots, t_{i-n}) \quad \text{(Strong hypothesis 2)}
\]

Example: « Dogs like to bark », using bi-grams \((n=2)\)

(All possible paths) ...

\[
P(N_{pl}, \text{verb,Inf\_marker,verb}|\text{dogs,like,to,bark})
\]

\[
= P(\text{dogs}|N_{pl}) \cdot P(\text{like}|\text{verb}) \cdot P(\text{to}|\text{Inf\_marker}) \cdot P(\text{bark}|\text{verb}) \cdot \prod_{i=1}^{N} P(w_i | t_i)
\]

\[
= P(\text{inf\_marker}|\text{verb}) \cdot P(\text{verb}|\text{inf\_marker}) \cdot \prod_{i=1}^{N} P(t_i | t_{i-1}, t_{i-2}, \ldots, t_{i-n})
\]

Copyright (c)2002 Faculté Polytechnique de Mons - T. Dutoit
4. Syntactic-Prosodic Phrasing

- Chinks’n chunks

\[ \text{a prosodic phrase} = \text{a sequence of chinks (≈function words)} \]
\[ \text{followed by a sequence of chunks (≈content words)} \]

- Example:
  
  I asked them
  if they were going home
  to Idaho
  and they said yes
  and anticipated one more stop
  before getting home

---

Classification and Regression trees (CARTs)

Predict Color(n) \(\Leftarrow\) SHapes, Sizes \((n,n-1,n+1,n-2,n+2,...)\)?

SOL: if \(\text{SHape}(n-1) = \text{SHape}(n)\) \(\Leftarrow \) Color(n) = White
else Color(n) = Black

This can be seen as a classification problem:

<table>
<thead>
<tr>
<th>C(n)</th>
<th>S(n)</th>
<th>SH(n)</th>
<th>S(n-1)</th>
<th>SH(n-1)</th>
<th>S(n+1)</th>
<th>SH(n+1)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>S</td>
<td>S</td>
<td>-</td>
<td>-</td>
<td>S</td>
<td>C</td>
<td>...</td>
</tr>
<tr>
<td>B</td>
<td>S</td>
<td>C</td>
<td>S</td>
<td>S</td>
<td>B</td>
<td>C</td>
<td>...</td>
</tr>
<tr>
<td>W</td>
<td>B</td>
<td>C</td>
<td>S</td>
<td>C</td>
<td>S</td>
<td>C</td>
<td>...</td>
</tr>
<tr>
<td>W</td>
<td>S</td>
<td>C</td>
<td>B</td>
<td>C</td>
<td>B</td>
<td>T</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

4. Syntactic-Prosodic Phrasing

- CART tree

\[ st = \text{time to end of sentence} \]
\[ j3 = \text{tag of word on the right} \]
\[ j2 = \text{tag of word on the left} \]
\[ tr = \text{utterance rate (in words/second)} \]

---

Classification and Regression trees (CARTs)

Question which splits into best « purified » sets

« impure » set

« pure » sets
Classification and Regression trees (CARTs)

« Purity » of a set?

= « Entropy » (bits)

= \(-P(\text{Black}) \log_2[P(\text{Black})] - P(\text{White}) \log_2[P(\text{White})]\)

Entropy = \(-(1/2 \times 1) - (1/2 \times 1)\)

= 1 bit

Total entropy after split = 0 + 0 = 0

5. Automatic Phonetization

• Decision trees for predicting phonemes+stress

Example:
s \rightarrow [s] or [z]

s [z] / [anti|hanti] _ [\langle V\rangle]

s [s] / [anti|contre|impr.|prime|tourne|ultra|psycho|télé] _ [l]

s [s] / [vrai] _ [em]

s [s] / [za|para|sinu] _ [e|o|y]

s [z] / [tran] _ [a|h|i]

s [z] / [\langle V\rangle] _ [\langle V\rangle]

6. Prosody generation: patterns

Copyright (c)2002 Faculté Polytechnique de Mons - T. Dutoit
6. Prosody generation: tones

- From text to tones

```
ce personnage grossier, te dérange-t-il
```

| WS | . . o . o . o . |
| SG | (. . .) (. .) (. . .) |
| IG 1 | (. . . /LL) (. . HH) (. . . R/H) |
| IG 2 | (. . . HH) (. . . R/H) |

WS = word stress = lexical stress ≤ Phonetization
SG = stress group
IG = intonation group ≤ Synt.-Pros. Phrasing
(only one stressed syllable)

6. Prosody generation: tones

- From tones to F0 (Hz) by rule:

```
H+ H H H L- l-
```

6. Prosody generation: tones

- From tones to F0 (Hz), corpus-based

- Large speech dba, with known intonation groups, F0 and tones
- For each target intonation group:
  - find a list of similar intonation groups (in terms of tones, number of syllables, position in sentence, etc.) in the dba
  - Select the sequence of intonation groups in the dba which:
    - best represents the target groups
    - AND minimizes intonative discontinuities
6. Prosody generation: tones

If $H, L, H+, L- = \text{stressed syllables}$
$h, l = \text{unstressed syllables}$

($l l l H$) ($l l H$) ($l l L-$)

**Towards corpus-based techniques**

1995-?: The *database* years

- For automatic phonetization (L&H, ENST, Univ. Edinburgh, FPMs)
- For automatic generation of intonation and phoneme duration (AT&T, FPMs, Univ. Aix, Univ. Edinburgh)
- For automatic selection of units for concatenative synthesis (ATR, Univ. Edinburgh, AT&T, FPMs?)

**Conclusion**

- Introduction
- Acoustic speech synthesis (DSP)
  - Model-based (rule-based) approach
  - Instance-based (concatenative) approach
- From text to phonemes and prosody (NLP)
  - Preprocessing
  - Morpho-syntactic analysis
  - Phonetization
  - Prosody Generation
- Towards...