# Limited Domain Synthesis

- ☐ Unit selection gives:
  - high quality
  - but sometimes low quality
  - (currently) difficult to build
- ☐ Limited domain:
  - every synthesis use is in a domain
  - often the domain is restricted

Can you get the advantages of unit selection and avoid the distadvantages

# Should this work?

- If utterances are in domain:
  - good examples are in db
  - less “bad” selections
- Design dbs around domain:
  - quaranteed coverage

# Basic tasks

- ☐ Designing the prompts
- ☐ Recording the prompts
- ☐ Labelling recorded speech
- ☐ Building utterance structures
- ☐ Extract Pitchmarks and MCEP coefficients
- ☐ Build a cluster unit selection synthesizer
- ☐ Testing and tuning

# Designing the prompts

- From a grammar:
  - in Dialog systems generation grammar is known
  - Use probabilistic generation to get coverage
- From data:
  - Find everything that has been said in the system
  - Order it based on frequency
- From thinking about it:
  - what is likely to be said
- Ideally:
  - word coverage
  - bi-gram coverage
  - intonation coverage

# Domains

- Talking clock:
  - very limited set format
  - 24 utterances
- weather reports
  - slot and filler, phrasal
  - 100 utterances
- Communicator
  - full dialog (open ?)
  - actually slot and filler
  - 500 utterances
- Movieline:
  - 54 standard utterances
  - plus times, theatres and movies.

# Talking clock

- – 24 utterances

```
( time0001 "The time is now, exactly five past one, in the morning." )
( time0002 "The time is now, just after ten past two, in the morning." )
...
( time0023 "The time is now, exactly five past eleven, in the evening." )
( time0024 "The time is now, a little after quarter to midnight." )
```

# Preliminaries

```
export ESTDIR=$SPPPDIR/src/speech_tools/
```
or
```
setenv ESTDIR $SPPPDIR/src/speech_tools/
```

```
export FESTVOXDIR=$SPPPDIR/src/festvox/
```
or
```
setenv FESTVOXDIR $SPPPDIR/src/festvox/
```

```
mkdir time_ldom
cd time_ldom
$FESTVOXDIR/src/ldom/setup_ldom cmu time awb
```

Creates directory structure, and copies default files

# Synthesizing prompts

- ☐ To guide speaker
- ☐ For labelling
- ☐ To judge time to record

```
festival -b festvox/build_ldom.scm '(build_prompts "etc/time.data")'
```

Builds, prompt waveforms and labels

# Record database

- Ensure audio levels are ok:
  - xmixer
- Record some examples:
  - listen and look at them

`bin/prompt_them etc/time.data 1`

or

`pointyclicky etc/time.data`

# Autoalign spoken prompts

- Generates cepstrum parameters
- dtw align prompts to speech

```
bin/make_labs promt-wav/*.wav
```

Check it worked

```
emulabel etc/emu_lab
```

# Build utterances

- Build utterances from:
  - synthesized form
  - corrected with actual durations

festival -b festvox/build_ldom.scm '(build_utts "etc/time.data")'

# Pitch marking

- Extract from EGG:
  – but you don't have one of those do you
- Extract from waveform
  – ESPS epoch (proprietary)
  – make_pm_wave

```
make_pm_wave wav/*.wav
```

Check and change params for speaker
(esp for female, but probably all)
See notes on festvox site

# Displaying pitch marking

- ☐ convert to labels
  - `bin/make_pm_lab pm/*.lab`
- ☐ display
  - `emulabel etc/emu_pm time0001`
  - zoom in to voiced section
- ☐ tune
  - switch off filler pm
  - tune pitch range and filters

# Extract MFCC

- ☐ Pitch synchronously

```
bin/make_mcep wav/*.wav
```

# Build Clunit synth

- Build a unit selection synthesizer
- Buckets of params we'll just ignore:
  – take defaults
  – for simple ldom dbs that's ok.

festival -b festvox/build_ldom.scm '(build_clunits "etc/time.data")'

# Build clunit synth

- Load utterances
- Name and sort all units:
  - phone_999 or
  - phone_word_999
- Dump selection features for each unit:
  - mostly phonetic, phrasal
  - no F0 or duration
- Load mcep params
- Build cluster trees with wagon
- Combine trees
- Dump catalog of units

# Test synthesizer

festival festvox/cmu_time_awb_ldom.scm

festival> (voice_cmu_time_awb)

festival> (saytime)

festival> (saythistime "11:25")

- ldom functions generate text:
  - in domain
  - calls SayText to synthesize
  - cannot synthesize out of domain

# Weather example

- Get hourly weather reports from weather.gov
  - For city, state: outlook, temperature and winds
  - sometimes the weather is unavailable
  - sometimes its unparsable
- From templates filled in slots:
  - 100 utterances
- Restrict clunits:
  - used phone_word units not phone units

# Communicator example

- Analysed past 3 months of logs:
  – it changes over time
- Selected based on frequency and coverage:
  – Top 250 utterances
  – another 250 for word coverage
- Delivered in “helpful agent” style
  – mostly phrasal selection
  – can do itineraries
- Restrict clunits:
  – used phone_word units not phone units

# Exercise 8

Due April 23th 12 noon. Do number 1 OR number 2

1. What time is it?
   Build a talking clock using the limited domain synthesis technique.
2. Build a full clunits synthesizer from: "A whole joy was reaping, but they've gone south, you should fetch azure mike."

# Hints 8

1. http://www.festvox.org has a whole chapter of this specific task, 5.6.
2. Don't worry too much about recording quality
3. For non-native speakers, try it, it should still work if you can deliver the prompts.
4. Can you deliver it in a different style voice?
5. The function `(saythistime "11:30")` allows you to test arbitrary times.
6. `(utt.save.wave (saythistime "11:30") "11-30.wav")` allows you to save waveforms
7. Submit three examples, at least one of which should be an example with an error (if possible).

# Hints 8

"A whole joy ..."

1. See list of commands on tutorial web page (its similar to the talking clock but not exactly)
2. See section 12.2
3. Set up as (using your name)

   SPPPDIR/src/festvox/src/unitsel/setup_clunits cmu us awb uniphone
4. Note as there is only one example of each phone, labelling *has* to be correct so you will need to hand correct these.