Phrasing

- Chunking utterances into breath sized pieces
- First approximation: punctuation
  – too little
- Second at content/function words
  – too much

Next week, some inmates released early from the Hampton County jail in Springfield, will be wearing a wristband that hooks up with a special jack on their home phones.

Next week | some inmates released early | from the Hampton County jail | in Springfield | will be wearing | a wristband | that hooks | up with a special jack | on their home phones.

Phrasing

- Banchenko and Fitzpatrick 90:
  - rule driven with punc, POS and syntax
  - balanced phrasing
  - (the boy saw) (the girl in the park)
  - (the boy in the park) (saw the girl)
- Hirschberg and Prieto 94:
  - CART trees
  - 95% for Spanish
- Ostendorf and Veilleux 94:
  - hierarchical statistical model
  - Multilevel breaks.

Taylor and Black 97

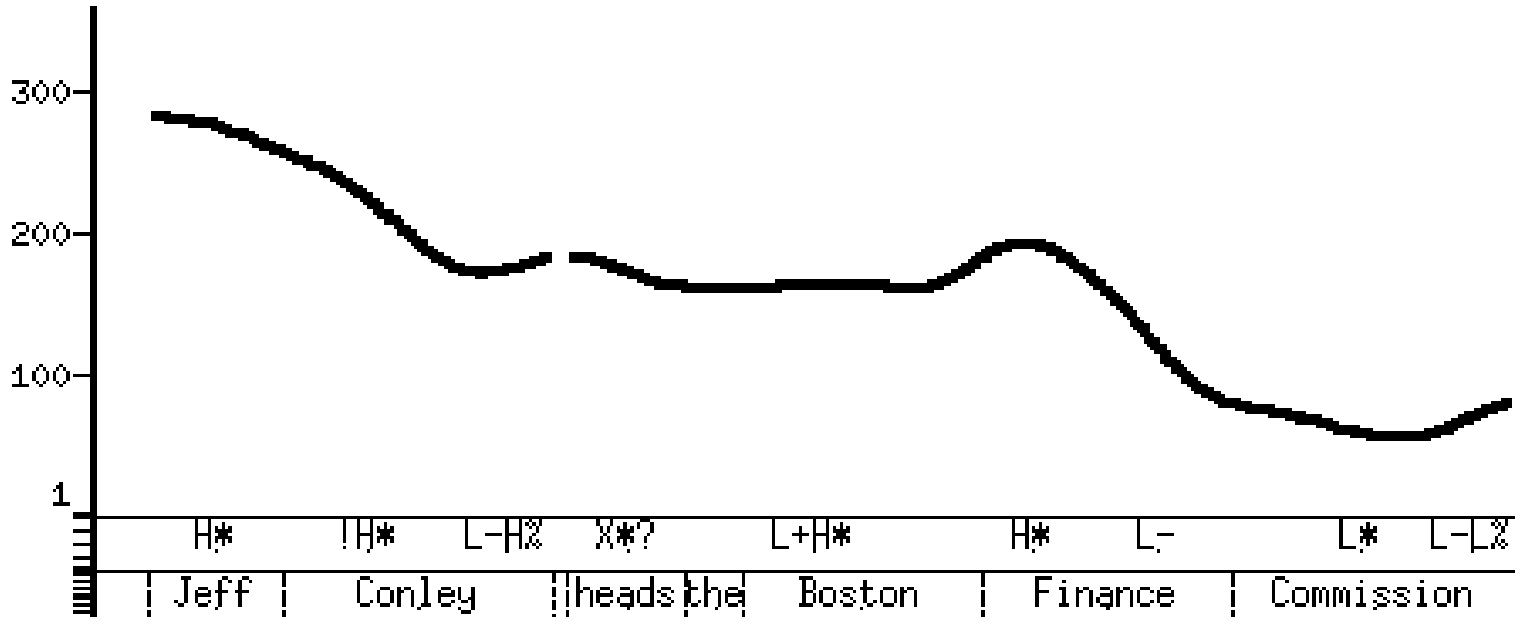
- Keeping balanced phrases
- two part:
  - Predict prob of break at point by CART
  - Base choice on previous break/non-break selections
- $$\prod_{k=1}^{n} \frac{P(B_k \mid B_{k-1}, \ldots, B_{k-N+1}) P(T_{k-N,\ldots,k+1} \mid B_k)}{P(T_{k-N,\ldots,k+1})}$$
- Trained on BBC Radio 4
  - 31,707 words, 6,346 breaks
  - 91% correct with 6-gram

Correctness

When is phrasing correct/wrong?

- Multiple *acceptable* phrasing exists
- Not all possible phrasing is acceptable:
  - but possible in some context
- Ostendorf and Veilleux 94:
  - same utterance by multiple speakers
  - if predicted utt matches any speaker its correct
- Some choices are arbitrary, some not

Intonation


300
200
100
1
H*
!H*
L-H%
X*?
L+H*
H*
L-
L*
L-L%
Jeff
Conley
heads
the
Boston
Finance
Commission

Intonation

- Predict:
  - accents, boundary tones
  - F0 contour
- More theories than researchers

Intonation Examples

- Fixed durations, flat F0.
- Decline F0
- "hat" accents on stressed syllables
- accents and end tones
- statistically trained

Theory neutral model

(not really neutral)

- Where do accents go?
- Where do boundaries go?
- What shape are they?
- What size, length, position are they

Where do accents go?

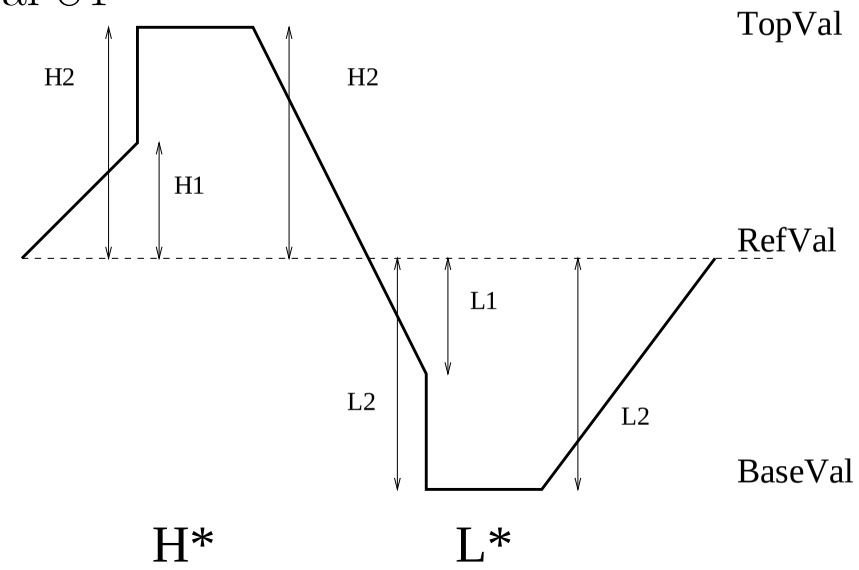
- On the important words.
- First approximation:
  - on stressed syllables in content words
  - 80% correct
- Hirschberg 92
  - hand written rules
  - compound/proper noun
  - phrase position etc
- Festival
  - uses CART on "Hirschberg" features

What shape are they?

- ToBI (Silverman et al 92):
  - Tones and Break Indices
  - Labelling standard *not* computational model
- 6 basic accent types:
  - H*, !H, L+H*, L*, L*+H
- 4 basic end tone types:
  - L-L%, L-H%, H-H%, H-L%
- Break level
  - 1, 2, 3, 4 (larger is bigger break)
- Disadvantages
  - no autolabeller
  - no F0 generator (but ...)
  - Why 6 ?

ToBI F0 generation

- Anderson et al 84



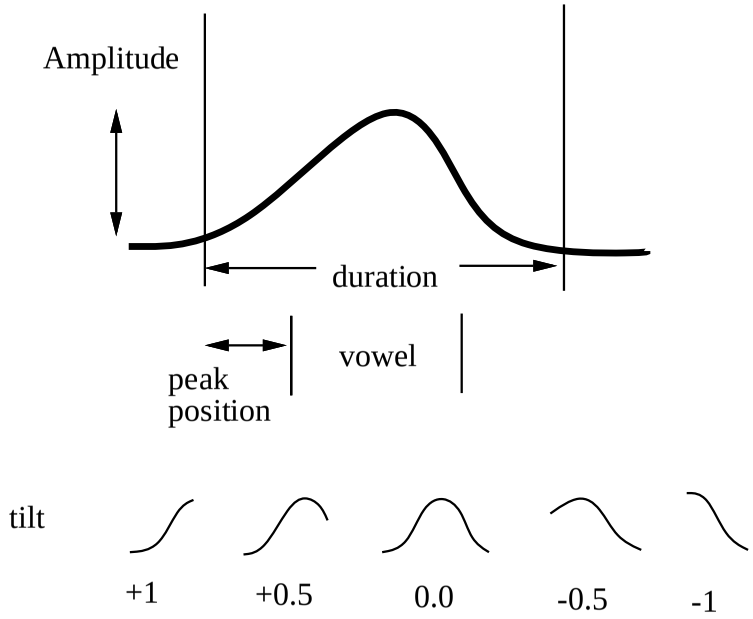
- Three point model (Black and Hunt 96)
  - Linear regression
  - predict start, mid vowel, end on syls
  - smooth result

From the other end ...

- Data driven approach
  - Build models from F0 contours
  - Extract F0
  - Smooth F0
  - Parameterize F0
- Models are good represenation of F0
  - small RMSE error (a few Hz)
- But can parameters be predicted?

Tilt Model: Taylor 97

- F0 derived:
  - accents+boundaries
  - 5 params per accent



Predicting Tilt Params

Dusterhoff (PhD 2000)

- CART for each Tilt param.
- Targeted F0 comparison.
- Comparison with ToBI (LR)
  Tilt: RMSE 32.5Hz and correlation of 0.60
  ToBI: RMSE 34.5Hz and correlation of 0.62
  Dynamical System (Ross): RMSE 33Hz

All on BU FM Radio data f2b

Other Intonation systems

- Fujisaki:
  - physiologically based F0 generation
  - Japanese and German
  - hard to predict
- van Santen
  - six point model
  - over intonation phrases
- Möhler
  - Vector Quantization
  - accent types auto built from data
  - from (sort of) tilt-like parameters
- Malfrere and Dutoit
  - Select natural contours from database

Measuring Intonation

- Multiple acceptable ways
- RMSE and correlation
  - insensitive to small errors
  - can be swamped by uninteresting parts
  - what about "microprosody"
  - Absolute, log, zscores
- Human perception tests
  - expensive to run
  - not very exact

Intonation Theory Wishlist

- From "accents to F0"
- From "F0 to accents"
- Easily trainable to new styles

But ...

- Can't really be done in isolation from
  - phrasing, duration and power

Intonation Summary

- Position of accents on syllables
- Type of accents/boundaries
- F0 contour generation

Duration

- Duration for each phones:
  - fixed (100ms)
  - average
  - statistically modelled
  - natural
- Overall speaking rate
  - global figure
  - need duration contour

Festival approach

- Collection of 153 features per segment
  - phonetic feature plus context
  - syllable type, position
  - phrasal position
  - *no* phone names
- domain:
  - abosolute, log, or
  - zscores ((X-mean)/stddev)
- CART or Linear Regression similar results
  - 26ms RMSE 0.78 correlation

Other duration approaches

- Syllable-based methods
  - Predict syllable times, then segment durations
  - But segment times don't correlate with syllable times
- Sums of Products model:
  - Linear Regression is: $W_0.F_0 + W_1.F_1 + ... + W_nF_n$
  - SoP model is $W_0.(F_0 * F_1 * ...) + W_i.(F_i * F_{i+1}...) + ...$
  - finding the right mix is computationally expensive
  - finding weights is easy
- Other learning techniques:
  - neural nets ...
- None predict varying speaking rate

Building a duration model

- Need data:
  - suitable speech data
- Need Labels:
  - all the labels/structure necessary
- Need feature extraction:
  - Should be *same* format as in synthesis
- Need training algorithm
- Need testing criteria

KDT Database

- KED Timit databases:
  - 452 phonetically balanced sentences
  - *"She had your dark suit in greasy wash water all year."*
- Hand labelled phonetically
- Recorded with EGG
- Collated into festival utterance structures

Building a duration model

Need to predict a duration for every segment
What features help predict duration?

- Phone:
  - type: vowel, stop, frictative
- Phone context:
  - preceding/succeeding phones (types)
- Syllable context:
  - onset/coda, stressing
  - word initial, middle final
- Word/phrasal:
  - content/function
  - phrase position
- Others?

Extracting training data

`dumpfeats`

- -relation Segment
- -feats durfeats.list
- -output durfeats.train
- utt0, utt1, utt2 ...

Festival Utterance feature names

- segment_duration
- name n.name p.name
- ph_*:
  - ph_vc
  - ph_vheight ph_vlng ph_vfront ph_vrnd
  - ph_cplace ph_ctype ph_cvox
- pos_in_syl syl_initial syl_final
- Syllable context:
  - R:SylStructure.parent.syl_break
  - R:SylStructure.parent.R:Syllable.p.syl_break
  - R:SylStructure.parent.stress

Full list is in Festival manual

Note features **and** pathnames

Train and test data

Guidelines

- Approx 10% data for test
- Could be partioning or
  - every nth utterance
- For timit let's use:
  - train: utts 001-339
  - test: utts 400-452

```
dumpfeats -relation Segment -feats durfeats.list
     -output durfeats.train kdt_[0-3]*.utt

dumpfeats -relation Segment -feats durfeats.list
     -output durfeats.test kdt_4*.utt
```

0.399028 pau 0 sh 0 0 0 0 0 0 0 0 - f 0 0 0 0 p - 0 1 1 0 0 0
0.08243 sh pau iy - 0 0 0 0 0 0 - + 0 1 l 1 - 0 0 0 1 0 1 0 0
0.07458 iy sh hh - f 0 0 0 0 p - - f 0 0 0 0 g - 1 0 1 1 0 0
0.048084 hh iy ae + 0 1 l 1 - 0 0 + 0 3 s 1 - 0 0 0 1 0 1 1 1
0.062803 ae hh d - f 0 0 0 0 g - - s 0 0 0 0 a + 1 0 0 1 1 1
0.020608 d ae y + 0 3 s 1 - 0 0 - r 0 0 0 0 p + 2 0 1 1 1 1
0.082979 y d ax - s 0 0 0 0 a + + 0 2 a 2 - 0 0 0 1 0 1 1 1
0.08208 ax y r - r 0 0 0 0 p + - r 0 0 0 0 a + 1 0 0 1 1 1
0.036936 r ax d + 0 2 a 2 - 0 0 - s 0 0 0 0 a + 2 0 1 1 1 1
0.036935 d r aa - r 0 0 0 0 a + + 0 3 l 3 - 0 0 0 1 0 1 1 1
0.081057 aa d r - s 0 0 0 0 a + - r 0 0 0 0 a + 1 0 0 1 1 1
0.0707901 r aa k + 0 3 l 3 - 0 0 - s 0 0 0 0 v - 2 0 0 1 1 1
0.05233 k r s - r 0 0 0 0 a + - f 0 0 0 0 a - 3 0 1 1 1 1
0.14568 s k uw - s 0 0 0 0 v - + 0 1 l 3 + 0 0 0 1 0 1 1 1
0.14261 uw s t - f 0 0 0 0 a - - s 0 0 0 0 a - 1 0 0 1 1 1
0.0472 t uw ih + 0 1 l 3 + 0 0 + 0 1 s 1 - 0 0 2 0 1 1 1 1
0.04719 ih t n - s 0 0 0 0 a - - n 0 0 0 0 a + 0 1 0 1 1 0
0.0964501 n ih g + 0 1 s 1 - 0 0 - s 0 0 0 0 v + 1 0 1 1 1 0
0.0574499 g n r - n 0 0 0 0 a + - r 0 0 0 0 a + 0 1 0 0 1 1
0.0441101 r g iy - s 0 0 0 0 v + + 0 1 l 1 - 0 0 1 0 0 0 1 1

Build CART model

wagon needs

- feature descriptions:
  - names and types (class/float)
  - make_wagon_desc durfeats.list durfeats.train durfeats.desc
  - and edit output
- tree build options:
  - stop size (20?)
  - held out data ?
  - stepwise
- Change domain:
  - absolute, log, zscores
  - ensure testing done in (absolute) domain

```
wagon -desc feats.desc -data feats.train -stop 20 -output dur.tree
Dataset of 12915 vectors of 26 parameters from: feats.base.train
RMSE 0.0278 Correlation is 0.9233 Mean (abs) Error 0.0171 (0.0219)

wagon_test -desc feats.desc -data feats.test -tree dur.tree
RMSE 0.0313 Correlation is 0.8942 Mean (abs) Error 0.0192 (0.0246)
```

Testing the model

- Use wagon_test on test data:
  - is this a good test set
- On "real" data:
  - Add new tree to synthesizer
  - test it
- Does it sound better:
  - can you tell?

Exercises for April 9th

Do either 1 or 2

1. Build an intonation system that adds small hat accents to all stressed syllables, without declination, except on final syllables, which rise for a question and fall for all other sentence types.
2. Build a duration model using CART for KED Timit database. Use given example features and report the RMSE and Corelation. Modify the features/wagon parameters etc to show (at least) one model that is better (or two that are worse). Run these models and comment how they actually sound.

You need to use the General Intonation system to build this. You'll need an accent assignment decision tree and a Scheme function predict the targets for each syllable.
The accent assignment tree is as mentioned above

```
(set! int_accent_cart_tree
 '
  ((R:SylStructure.parent.gpos is content)
   ((stress is 1)
    ((Accented))
    ((position_type is single)
     ((Accented))
     ((NONE))))
   ((NONE))))
```

And you'll need to write a function that generates the F0 targets, something like

```
(define (targ_func1 utt syl)
  "(targ_func1 UTT ITEM)
Returns a list of targets for the given syllable."
  (let ((start (item.feat syl "syllable_start"))
        (end (item.feat syl "syllable_end")))
    (cond
     ((string-matches (item.feat syl "R:Intonation.daughter1.name") "A
       (list
        (list start 110)
        (list (/ (+ start end) 2.0) 140)
        (list end 100)))
      (
        ;; End of utterance as question
        ;; target for mid point and high end pont
      )
      ( ;; End of utterance but not question
        ;; target for mid point and low end pont
      ))))
```

The condition `(equal? nil (item.next syl))` will be true for the last syllable in the utterance and

```
(string-matches (item.feat syl
    "R:SylStructure.parent.R:Word.last.R:Token.parent.punc") "?")
```

will be true if there is a question mark at the end of the utterance.

You also need to set up these functions as the intonation method

```
(voice_ked_diphone)
(Parameter.set 'Int_Method 'General)

(set! int_general_params
      (list
       (list 'targ_func targ_func1)))
```

Basic data in $SPPPDIR/data/kdt/
Utterances in $SPPPDIR/data/kdt/festival/utts/
Some basic features in $SPPPDIR/data/kdt/dur/feats.basic
For more features (or add your own) see Festival manual appendix
http://estvox.org/docs/manual-1.4.2/

```
dumpfeats -relation Segment -feats .../feats.basic
     -output .../durfeats.train .../kdt_[0-3]*.utt
make_wagon_desc durfeats.list durfeats.train durfeats.desc
```

(edit durefests.desc)

```
wagon -desc feats.desc -data feats.train -stop 20 -output dur.tree
wagon_test -desc feats.desc -data feats.test -tree dur.tree
```

```
(define (Duration_Simple_Tree utt)
  "Duration_Simple_Tree utt)
predicts Segment durations with a simple CART tree (that returns
absolute times in seconds)."
  (let ((end 0))
    (mapcar
     (lambda (s)
       (let ((dur (wagon_predict s simple_duration_cart_tree)))
         (set! dur (* (Parameter.get 'Duration_Stretch) dur))
;        (format t "%s %f\n" (item.name s) dur)
         (set! end (+ dur end))
         (item.set_feat s "end" end)
         ))
     (utt.relation.items utt 'Segment))
    utt))

(voice_ked_diphone)
(Parameter.set 'Duration_Method Duration_Simple_Tree)
(set! simple_duration_cart_tree
      (car (load "/home/awb/data/kdt/dur.tree" t)))
```