Module 5

Waveform generation

Roadmap

- Modules 1-2: The basics
- Modules 3-5: Speech synthesis
- Modules 6-9: Speech recognition

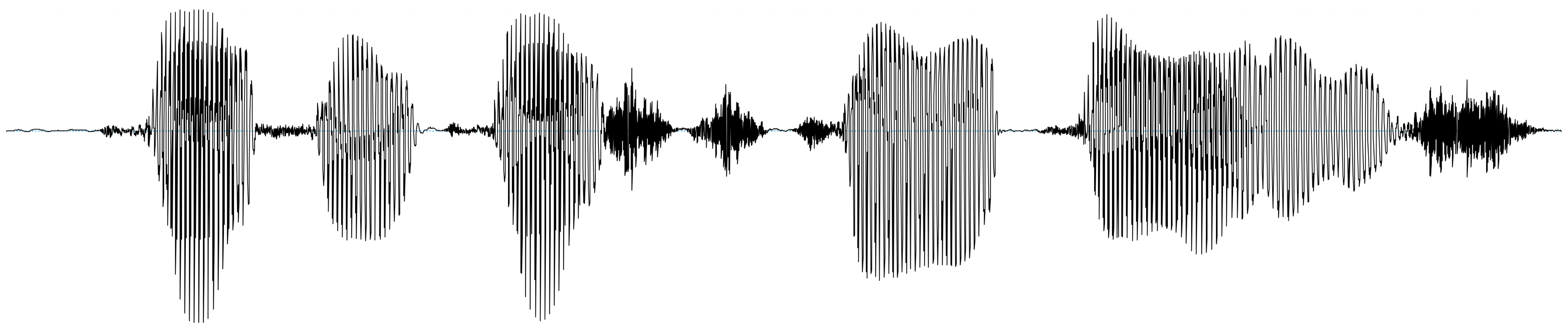
- Block 1 Week 4
  - Module 3: text processing
- Block 1 Week 5
  - Class trip
  - Module 4: pronunciation & prosody
- Block 1 Week 6
  - Assignment Q&A
  - Module 5: waveform generation
- Block 1 Week 7
  - Submission of first assignment

Orientation

- Text-to-speech pipeline architecture
  - Normalise text
  - Predict pronunciation & prosody

  - **Generate waveform**
    - start with recorded speech units
    - manipulate them to
      - join smoothly
      - have the desired prosody

SIL K AA F IY K AA S T S
T UW P AW N D Z SIL



What you should already know

- From the videos & readings
  - Concatenation of waveform fragments
  - Diphone units
  - Waveform manipulation
    - TD-PSOLA
    - Linear predictive model

choosing units that capture contextual effects

i.e., **co-articulation**

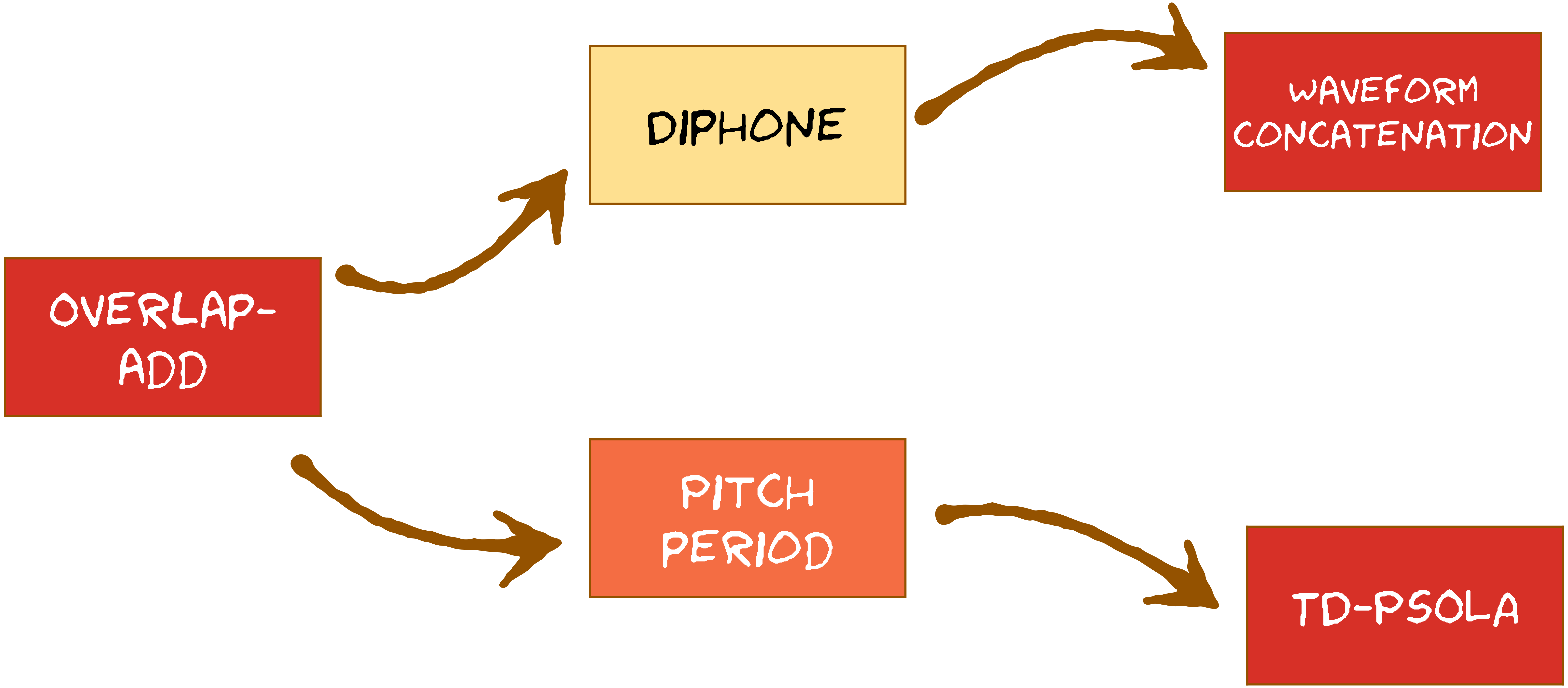
can only modify duration and F0

can also modify the filter / spectral envelope / vocal tract shape

Today's topics - Module 5: waveform generation

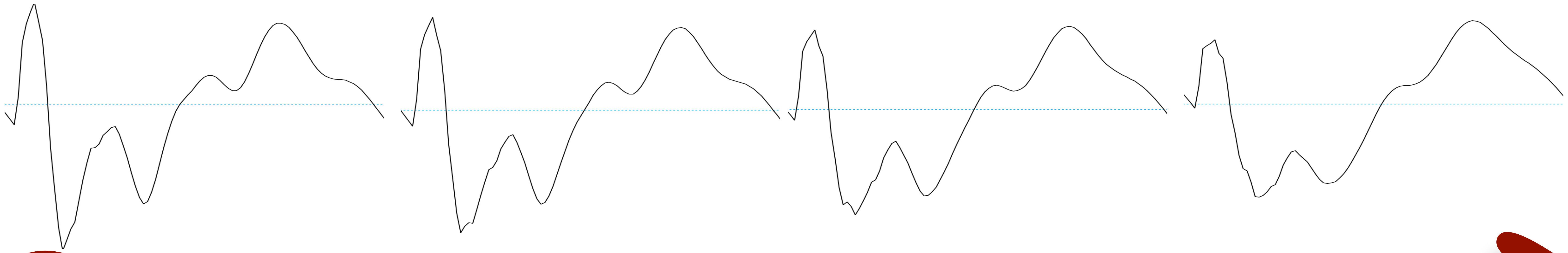
	THEORY					APPLICATION					
	SPEECH			SIGNAL PROCESSING	PROBABILISTIC MODELLING	SPEECH SYNTHESIS		AUTOMATIC SPEECH RECOGNITION			
	SIGNALS	PRODUCTION	PERCEPTION			FRONT END	WAVEFORM GENERATION	FEATURE EXTRACTION	PATTERN MATCHING	HIDDEN MARKOV MODELS	CONNECTED SPEECH
CONCEPTS	TIME DOMAIN	SOUND SOURCE	PITCH	DIGITAL SIGNAL	DESCRIBING DATA	TOKENISATION & NORMALISATION	WAVEFORM CONCATENATION	SERIES EXPANSION	EXEMPLAR	GENERATIVE MODEL OF SEQUENCES	HIERARCHY
	PERIODIC SIGNAL	HARMONICS	COCHLEA	SHORT-TERM ANALYSIS	DISCRETE & CONTINUOUS VARIABLES	PRONUNCIATION	DIPHONE	FEATURES	DISTANCE		SUB-WORD UNIT
	FREQUENCY DOMAIN	VOCAL TRACT RESONANCE & FORMANTS	MEL SCALE	SPECTRAL ENVELOPE	JOINT, CONDITIONAL, BAYES' FORMULA	PROSODY		FEATURE ENGINEERING	SEQUENCE	HIDDEN STATE SEQUENCE	N-GRAMS
MODELS & DATA STRUCTURES	FILTER	RESONANT TUBE	FILTERBANK	IMPULSE TRAIN	GAUSSIAN	FINITE STATE TRANSDUCER		FEATURE VECTOR	SEQUENCE OF FEATURE VECTORS	HIDDEN MARKOV MODEL	
	IMPULSE RESPONSE	SOURCE-FILTER MODEL	PHONEME	PITCH PERIOD	GENERATIVE MODEL	DECISION TREE			GRID	LATTICE	GRAPH
ALGORITHMS & ANALYSIS				FOURIER ANALYSIS	FITTING A GAUSSIAN TO DATA	HANDWRITTEN RULES	OVERLAP-ADD	MFCCS	DYNAMIC PROGRAMMING (DTW)	DYNAMIC PROGRAMMING (VITERBI)	COMPOSITION ("COMPILING")
				CEPSTRAL ANALYSIS	CLASSIFICATION	LEARNING DECISION TREES	TD-PSOLA			BAUM WELCH	APPROXIMATION (PRUNING)

Today's topics - Module 5: waveform generation



concatenating units

OVERLAP-ADD

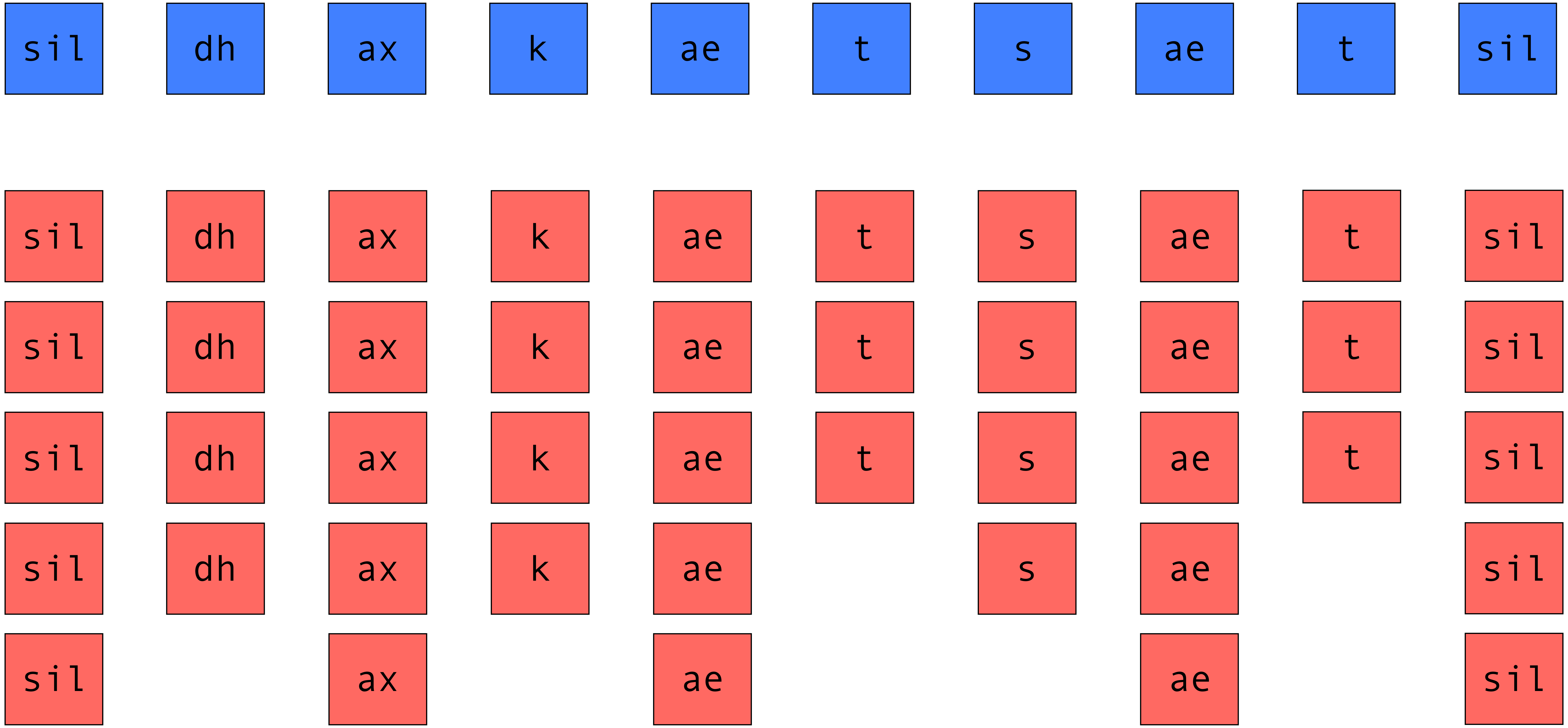


manipulation within units

Speech synthesis - waveform generation

- <u>Extending diphone synthesis to unit selection</u>
- Signal processing for waveform modification
  - Time-domain method: TD-PSOLA
  - Source-filter model-domain method: linear predictive filtering

Which candidate sequence will sound best?



Similarity between candidate sequence and the target sequence

- The ideal candidate unit sequence might comprise units taken from
  - **identical linguistic contexts** to those in the target unit sequence
- Of course, this will not be possible in general
  - so we must use less-than-ideal units from non-identical (i.e., **mismatched**) contexts
- We need to **quantify** how mismatched each candidate is, so we can choose amongst them
- The mismatch 'distance' or 'cost' between a candidate unit and the ideal (i.e., target) unit is measured by the *target cost function*

Join cost

- The join cost measures the **acoustic mismatch** between two candidate units
- A typical join cost quantifies the acoustic mismatch across the concatenation point
  - e.g., spectral characteristics (parameterised as MFCCs, perhaps), F0, energy

- Festival's *multisyn* uses a sum of normalised sub-costs (weights tuned by ear)

Speech synthesis - waveform generation

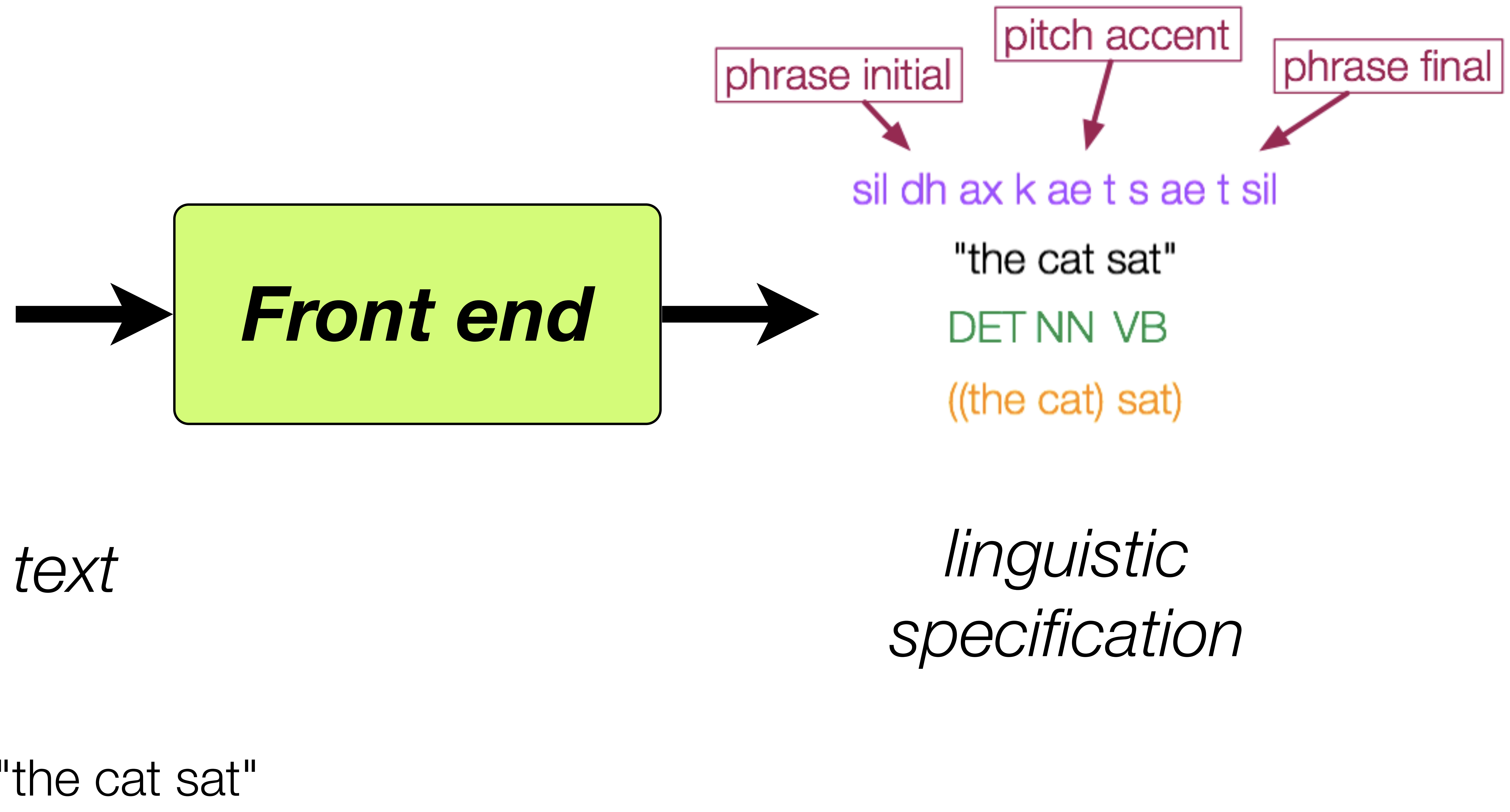
- Extending diphone synthesis to unit selection
- Signal processing for waveform modification
  - Time-domain method: TD-PSOLA
  - Source-filter model-domain method: linear predictive filtering

Why do we need to manipulate the recorded speech?

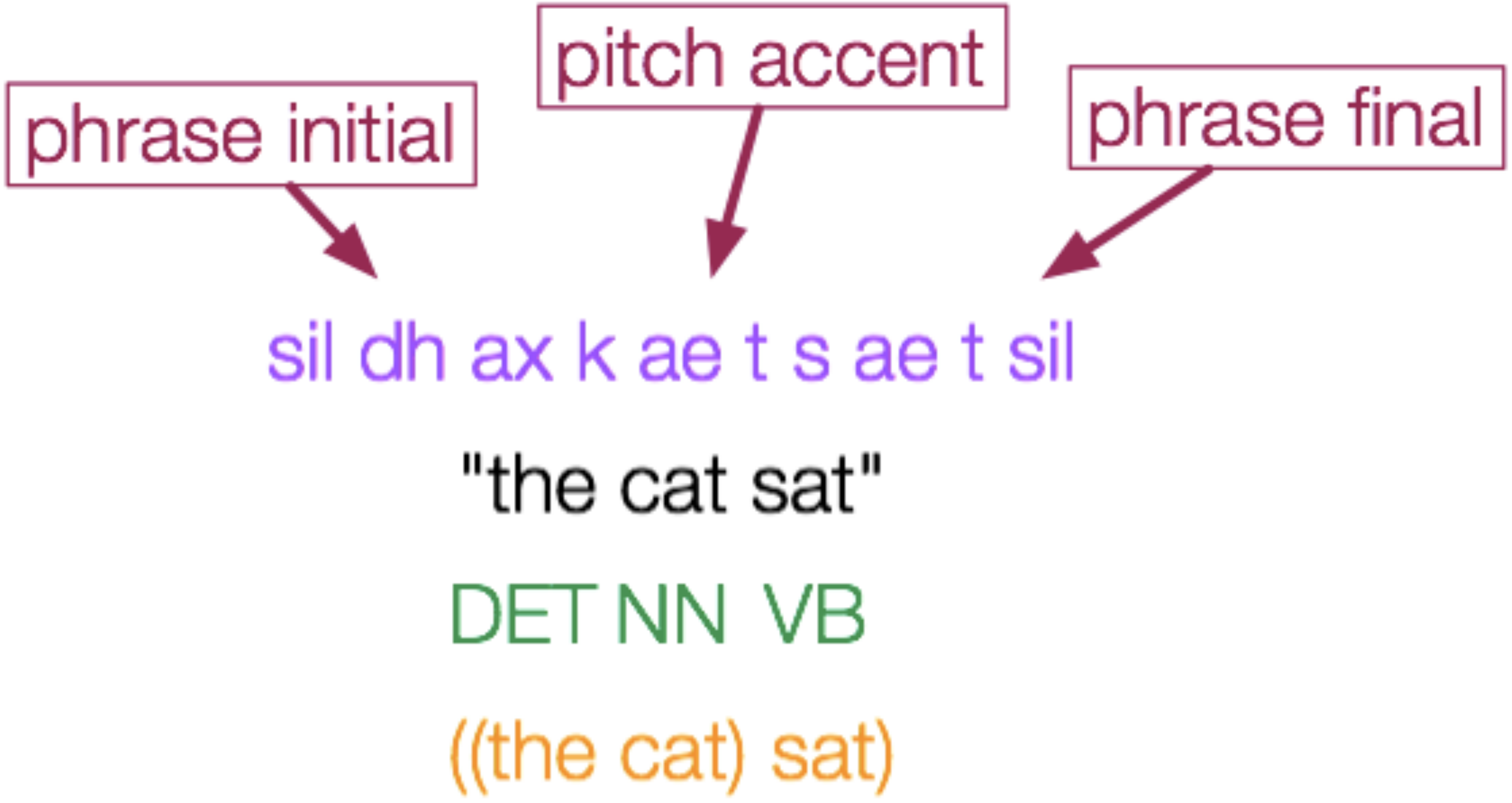
- Diphone synthesis
  - we only have a single recorded example of each diphone
  - so, it won't have the correct F0 or duration
  - we want to to impose the F0 and duration **predicted by the front end**

- Unit selection (full details in the Speech Synthesis course)
  - to disguise the joins by "*lightly* **smoothing**" F0 and the spectral envelope in the local region around each join
  - imposing F0 and duration predicted by the front end is *optional*

What does the front end produce as output?



For diphone synthesis, must predict acoustic properties



Predicted acoustic properties

linguistic specification							
phones	sil	s	ay	m	ax	n	sil
desired duration							
desired F0							

Retrieve recorded diphones from the database


sil_s


m_ax

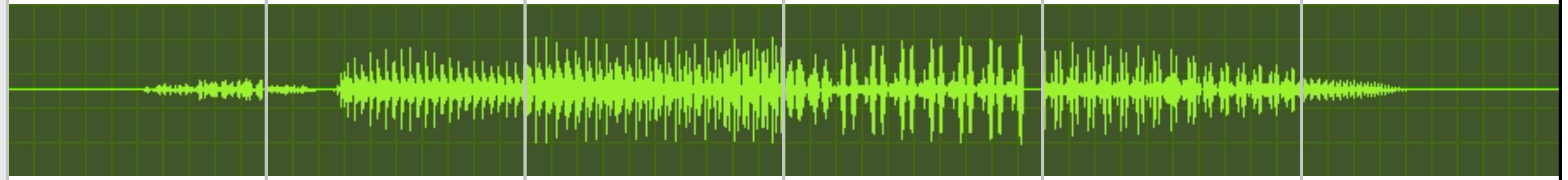

s_ay


ax_n

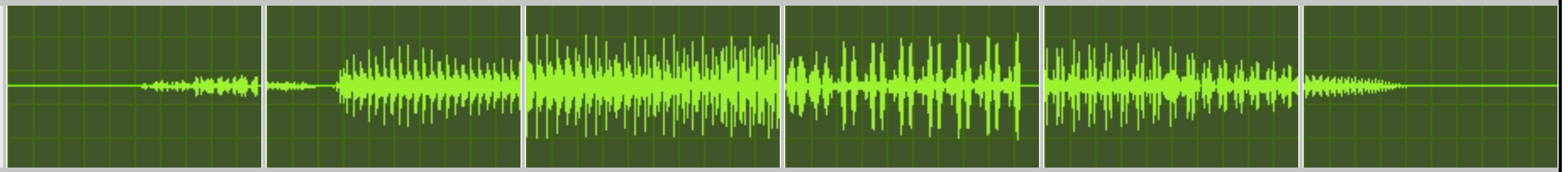

n_sil


ay_m

Retrieve recorded diphones from the database

recorded diphones from the database						
diphones	sil_s	s_ay	ay_m	m_ax	ax_n	n_sil
recorded diphones						
duration						
F0						

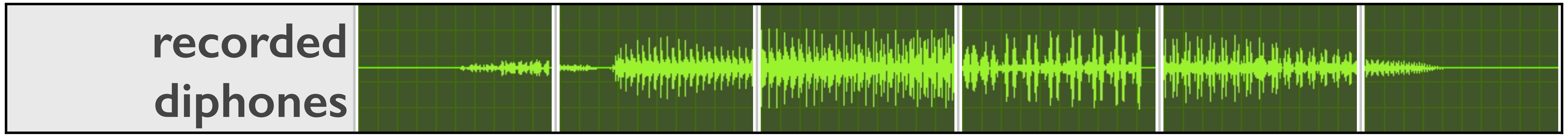
Make a plan for manipulating F0 and duration

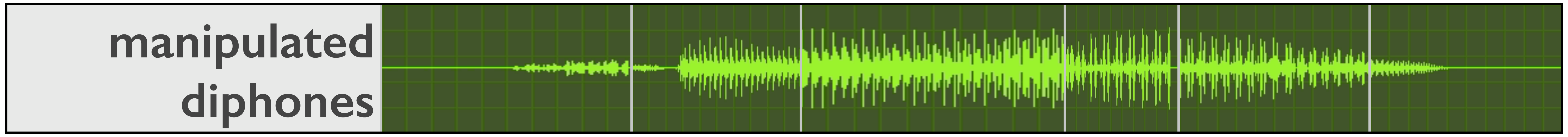
actual vs. desired F0 and duration						
diphones	sil_s	s_ay	ay_m	m_ax	ax_n	n_sil
recorded diphones						
actual duration						
desired duration						
actual F0						
desired F0						

Speech synthesis - waveform generation

- Extending diphone synthesis to unit selection
- Signal processing for waveform modification
  - <u>Time-domain method: TD-PSOLA</u>
  - Source-filter model-domain method: linear predictive filtering

Step-by-step waveform generation: TD-PSOLA version





Speech synthesis - waveform generation

- Extending diphone synthesis to unit selection
- Signal processing for waveform modification
  - Time-domain method: TD-PSOLA
  - <u>Source-filter model-domain method: linear predictive filtering</u>

Using a model of speech to perform manipulation

- Convert speech waveform into
  - **parameters** of a source-filter model
  - e.g., LPC: filter co-efficients + F0 + voicing decision (V/UV)

- Discard waveforms
- **Store model parameters**

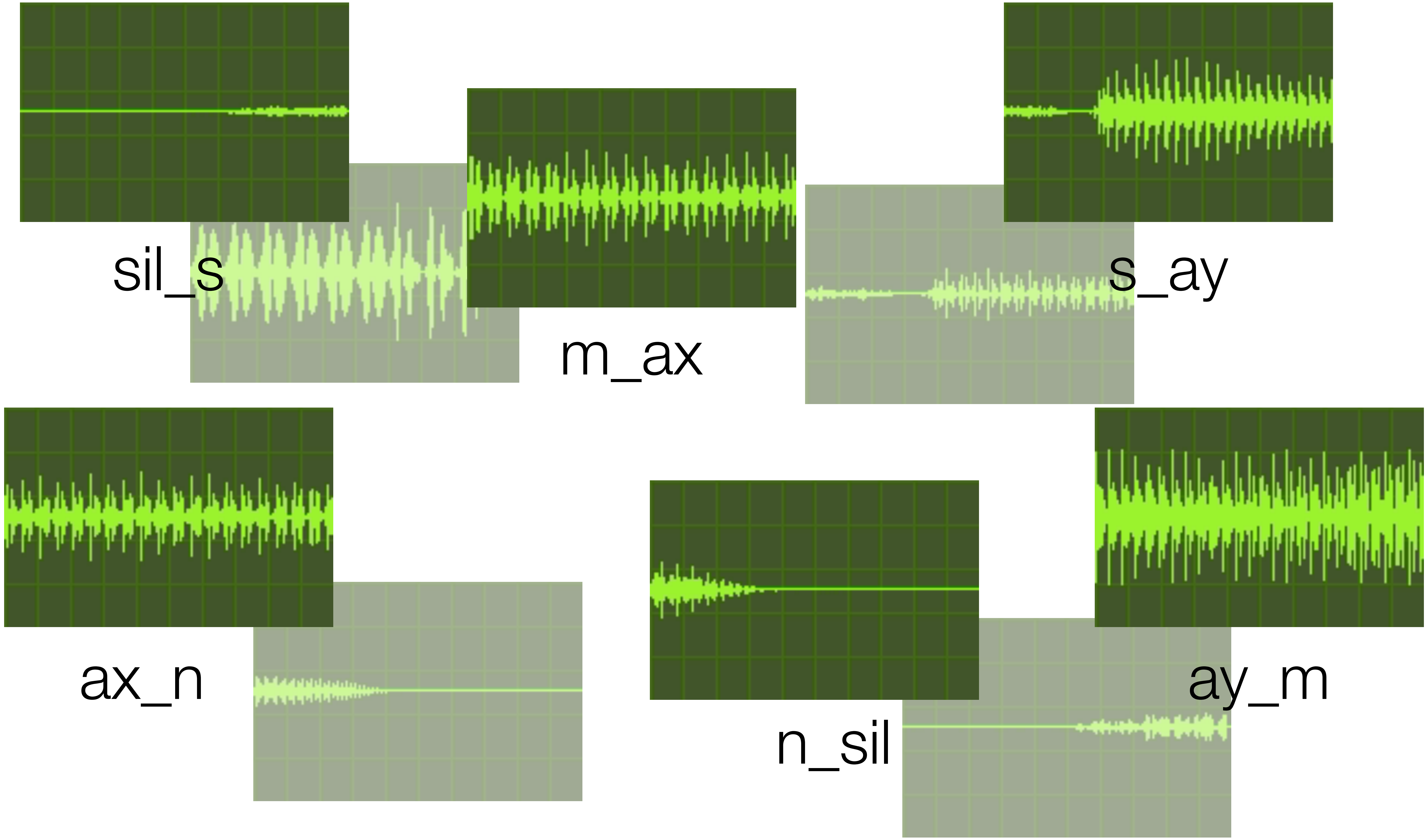
- At synthesis time
  - retrieve model parameters from database
  - modify parameters if required, then **resynthesise**

Step-by-step waveform generation: LPC version

- When building the voice
  - convert recorded waveforms into source + filter
  - source: F0 + voicing decision
  - filter: LPC coefficients

- When generating the waveform
  - manipulate source to achieve desired duration and F0
  - interpolate filter coefficients to match
  - reconstruct waveform from manipulated source + filter

LPC: convert speech into model parameters



LPC: convert speech into model parameters

m_ax

- For each frame
  - fit the filter to the signal (captures the spectral envelope)
    - i.e., solve some equations to find the filter co-efficients
  - inverse filter the speech to obtain the residual
  - store the filter co-efficients and the residual signal (which is a waveform)

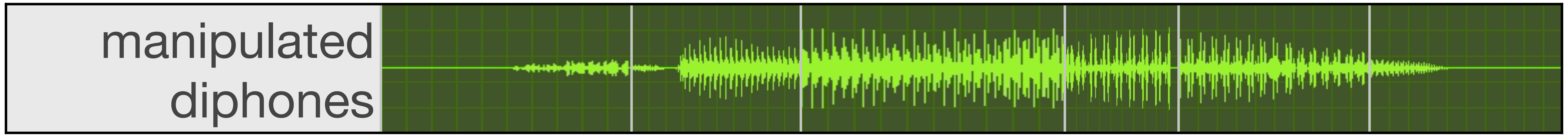
LPC: convert speech into model parameters

$$y[t] = e[t] - \sum_{k=1}^{K} b_k \, y[t-k]$$

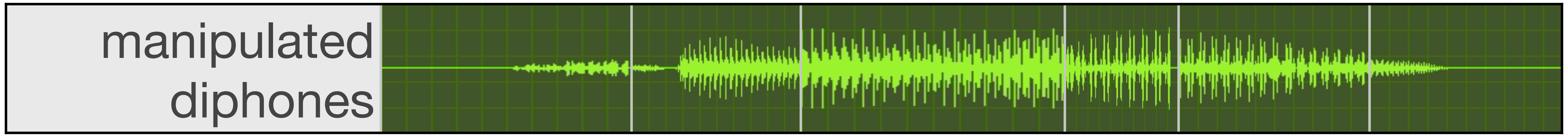
LPC: convert speech into model parameters

Step-by-step waveform generation: LPC version

- Retrieve filter co-efficients and residual signals from database
- Construct residual signal for utterance using concatenation
  - can manipulate F0 & duration with PSOLA method
- Interpolate filter co-efficients to be pitch-synchronous
- Pass residual signal through filter
  - update filter parameters once per pitch period



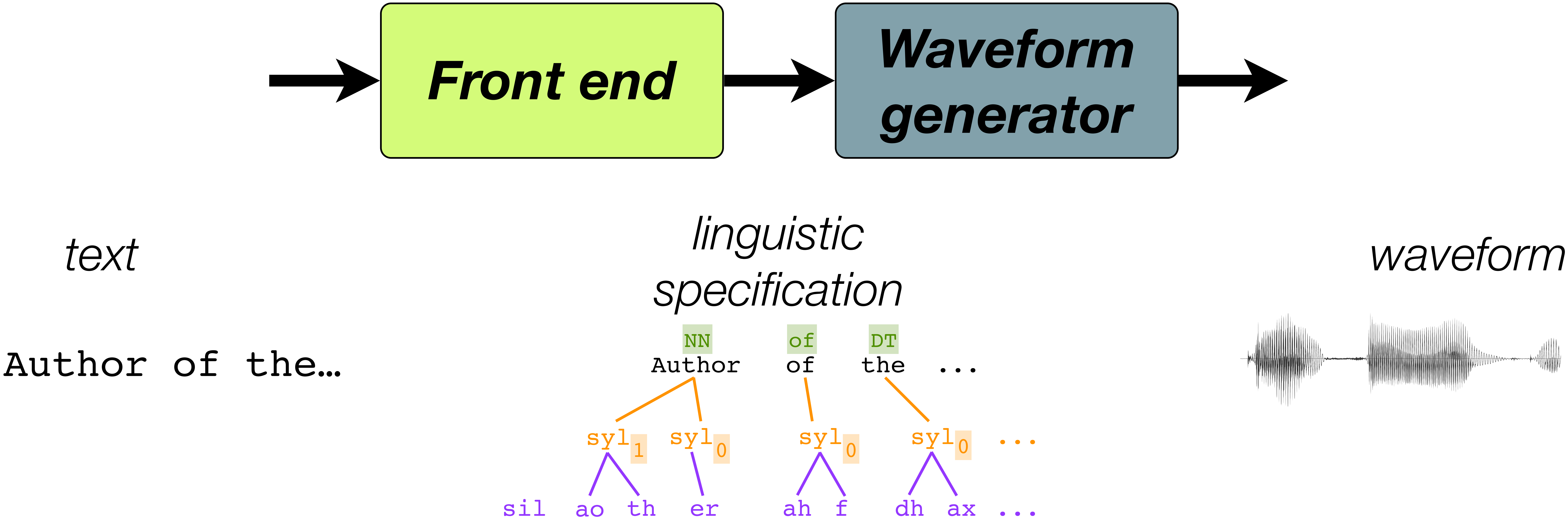
Step-by-step waveform generation: LPC version



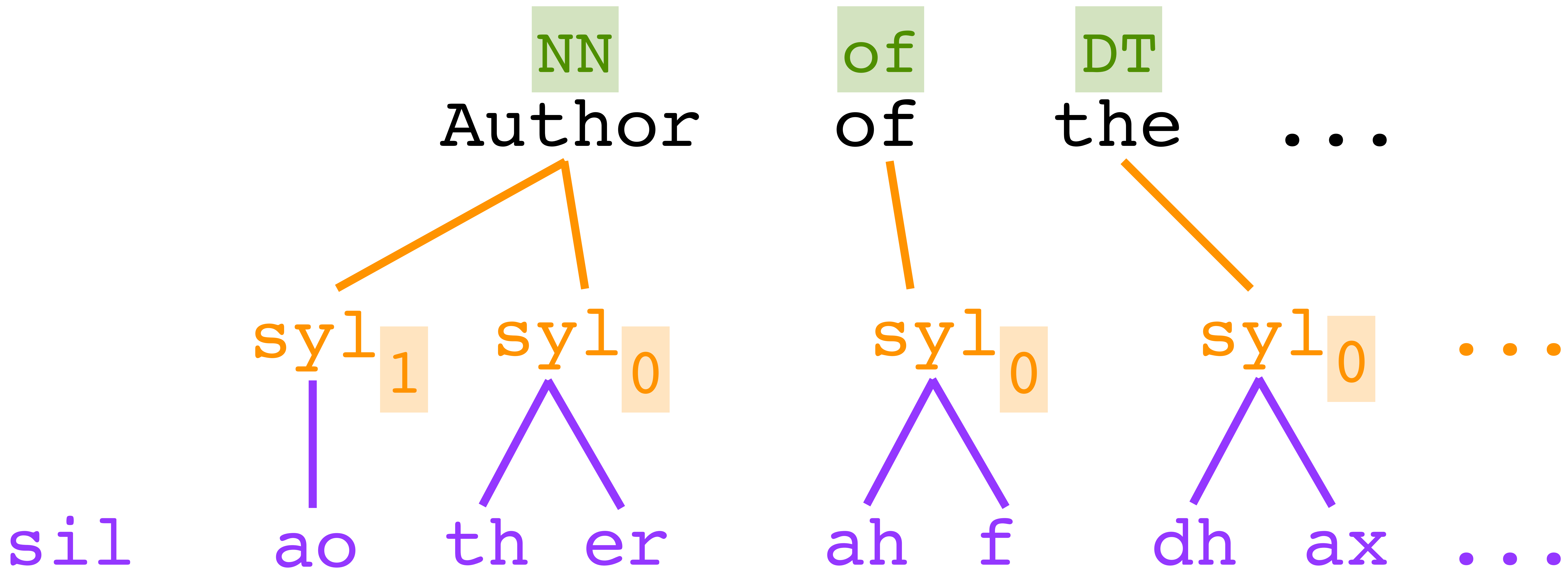
Speech synthesis - waveform generation

- Putting the whole pipeline together

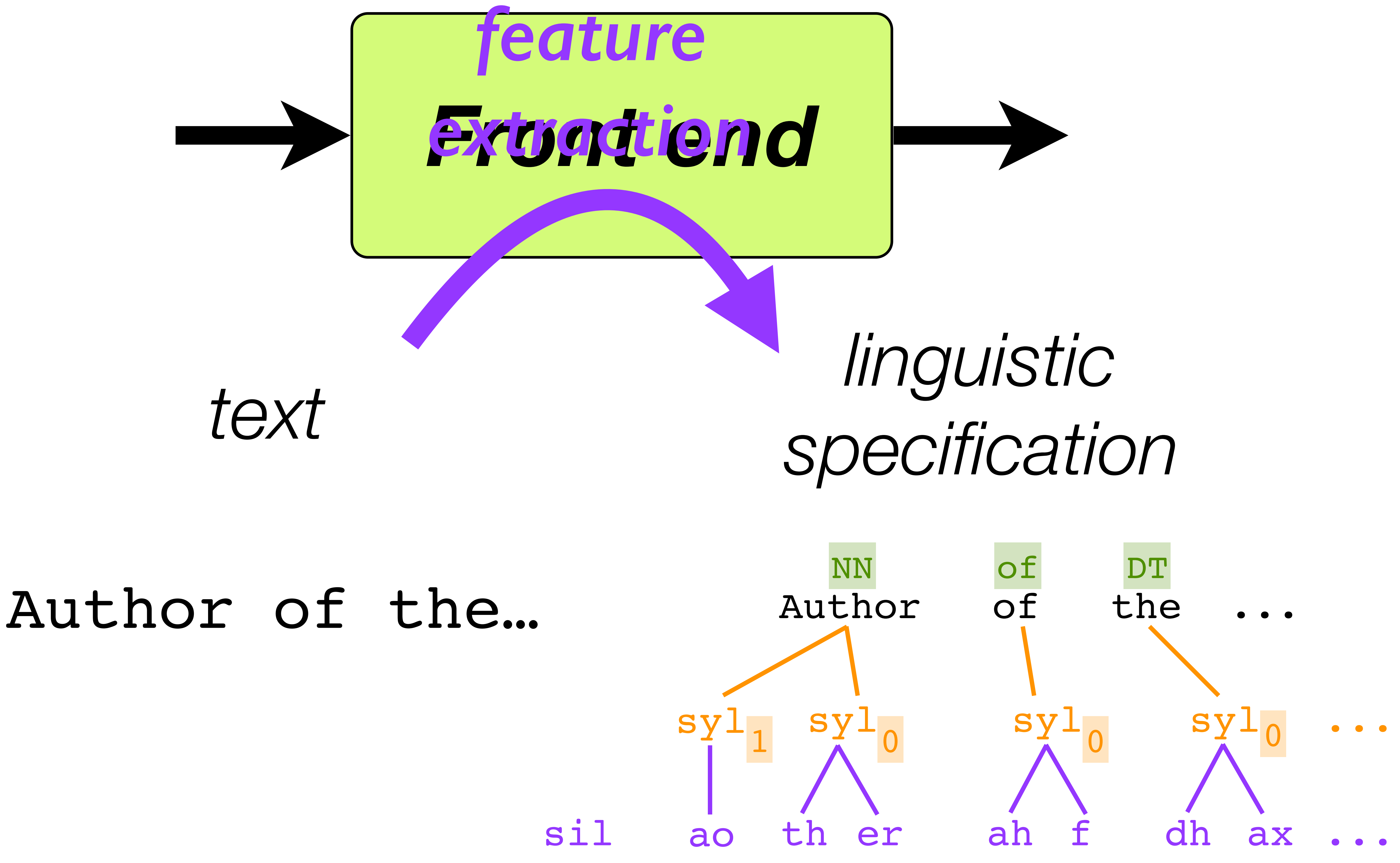
The classic two-stage pipeline of text-to-speech synthesis



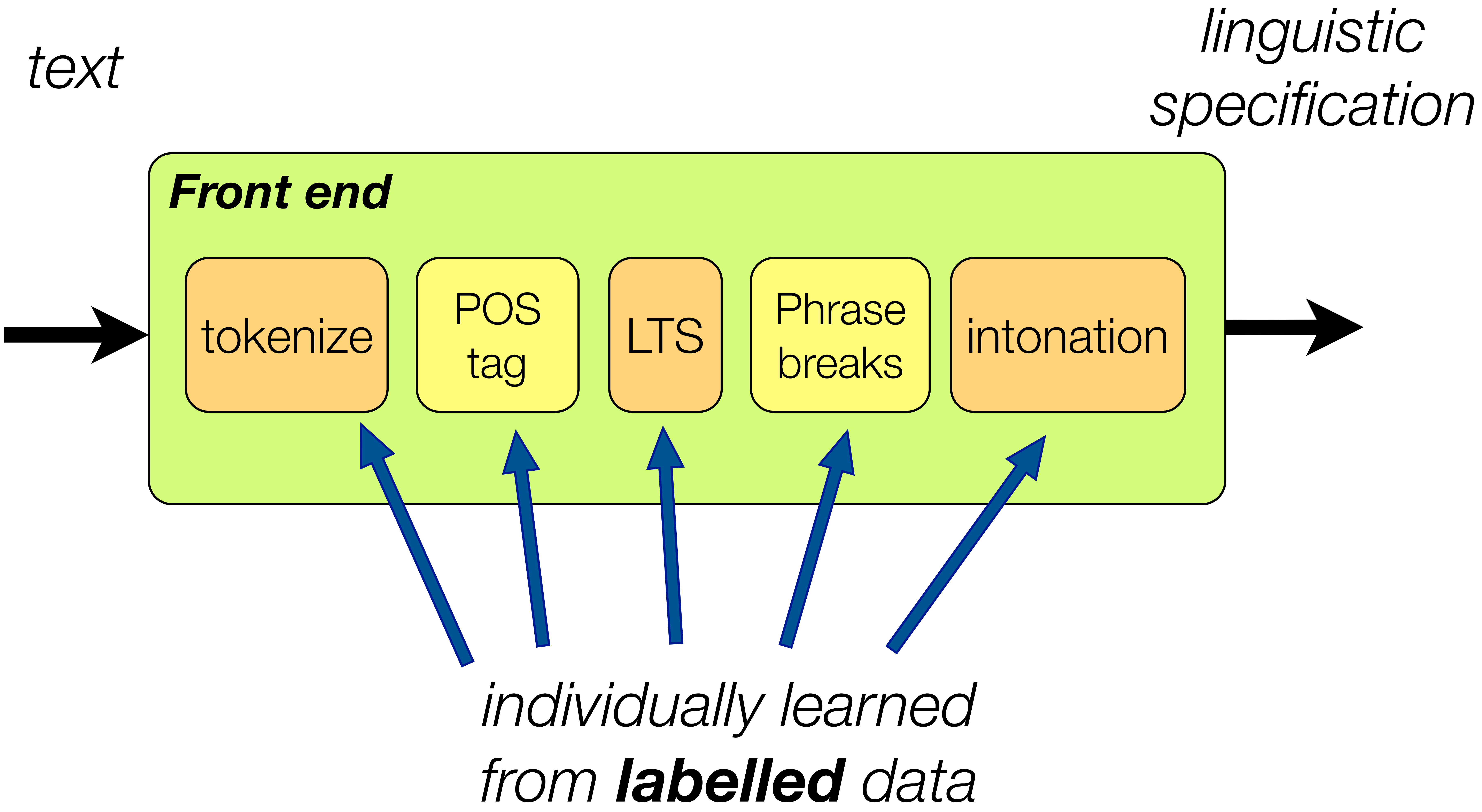
The linguistic specification



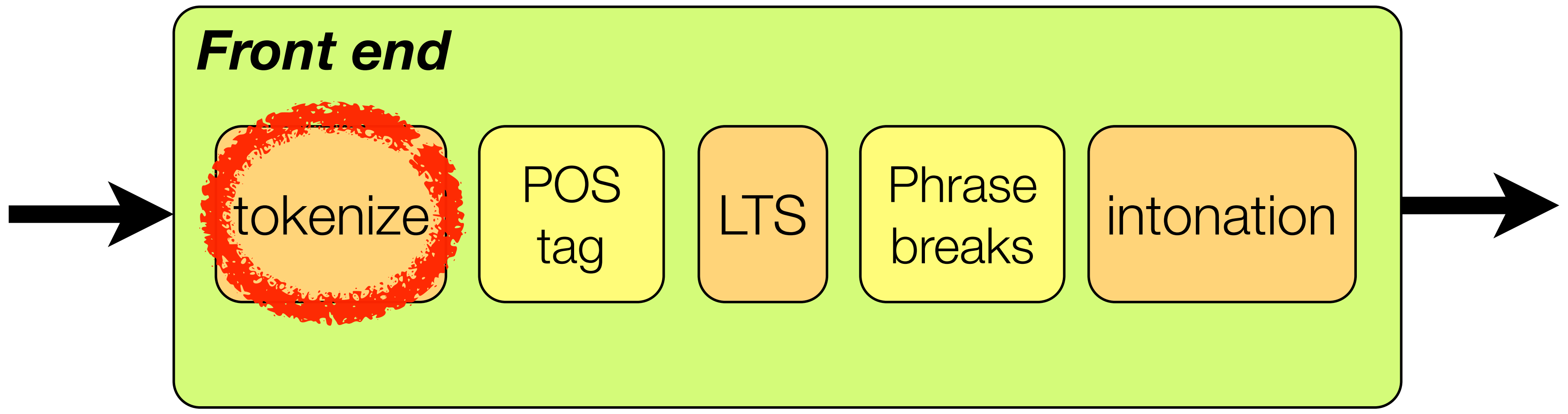
Extracting features from text using the front end



Text processing pipeline



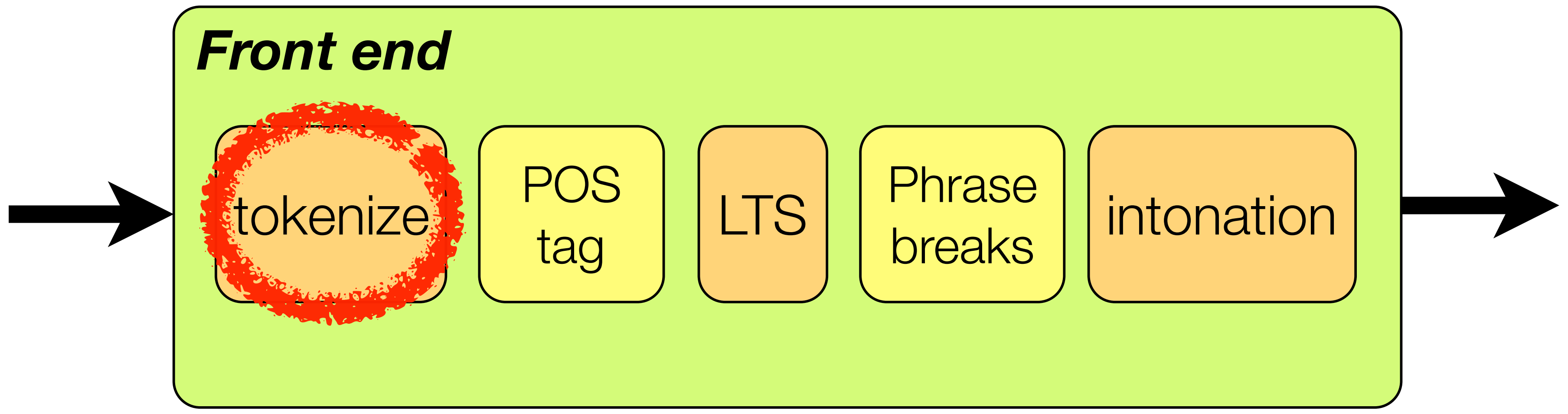
Tokenize & Normalize



- Step 1: divide input stream into tokens, which are potential words

- For English and many other languages
  - rule based
  - whitespace and punctuation are good features

- For some other languages, especially those that don't use whitespace
  - may be more difficult
  - other techniques required (out of scope here)

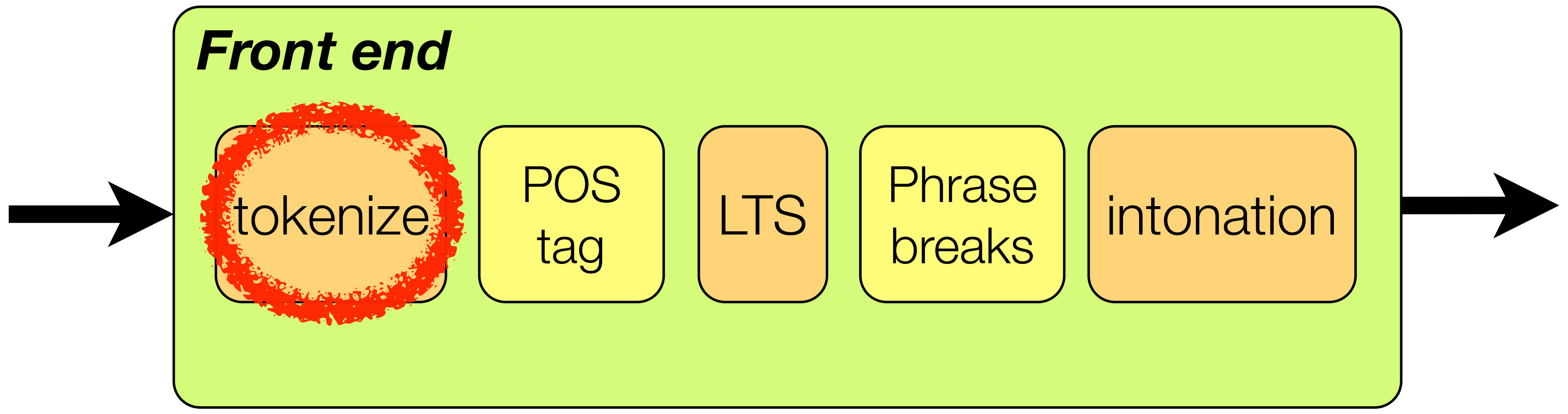


Tokenize & Normalize

- Step 2: classify every token, finding **Non-Standard Words** that need further processing

```
In 2011, I spent £100 at IKEA on 100 DVD holders.

   NYER           MONEY   ASWD    NUM LSEQ
```



Tokenize & Normalize

- Step 3: a set of specialised modules to process NSWs of a each type

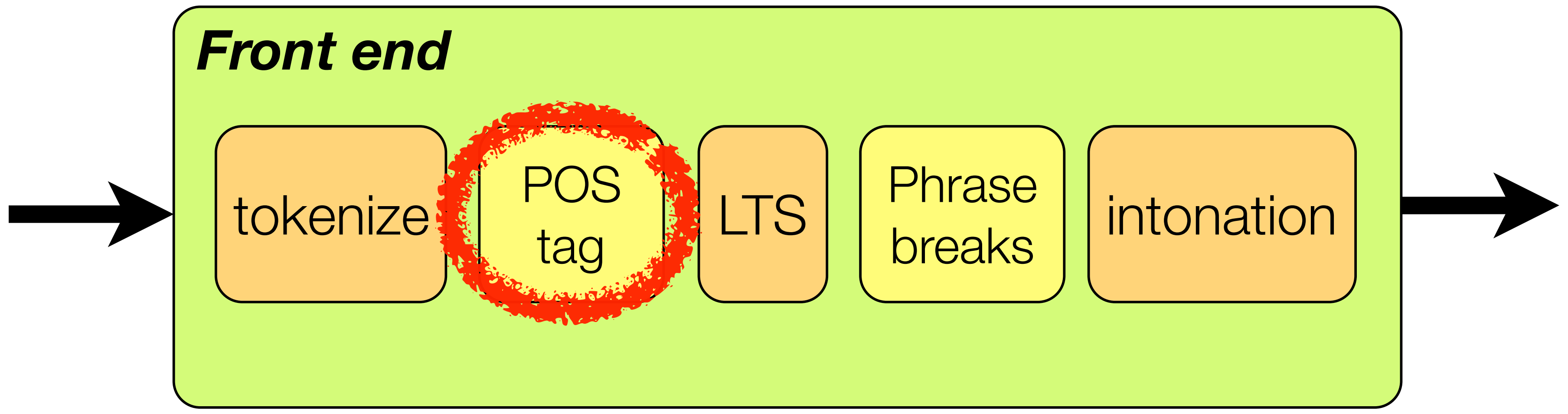
2011 ⇨ NYER ⇨ twenty eleven

£100 ⇨ MONEY ⇨ one hundred pounds

IKEA ⇨ ASWD ⇨ *apply letter-to-sound*

100 ⇨ NUM ⇨ one hundred

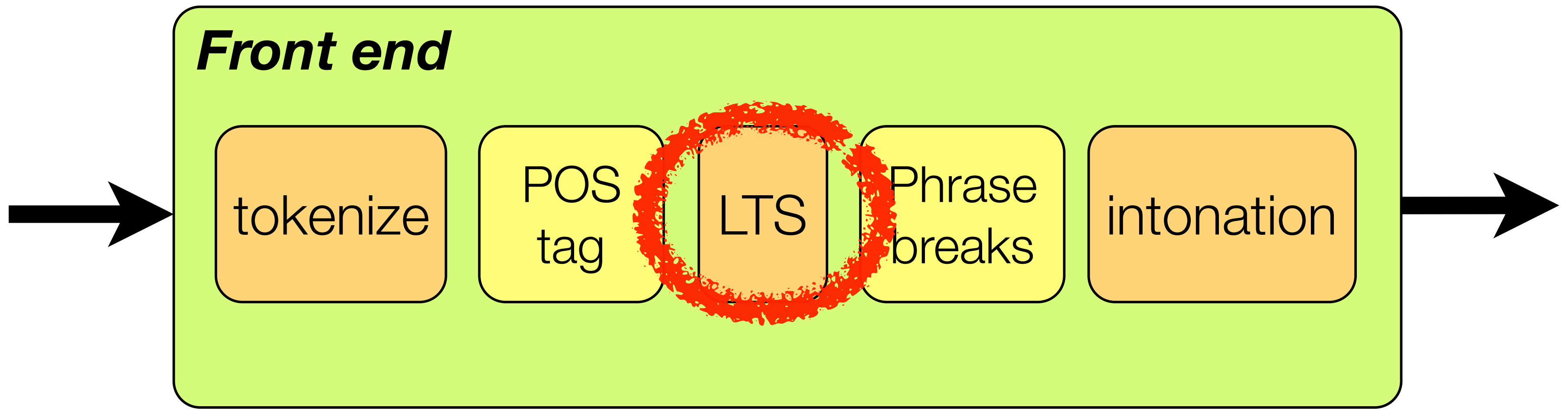
DVD ⇨ LSEQ ⇨ D. V. D. ⇨ dee vee dee



POS tagging

- Part-of-speech tagger
- Accuracy can be very high
- Trained on **annotated** text data
- **Categories** are designed for text, not speech

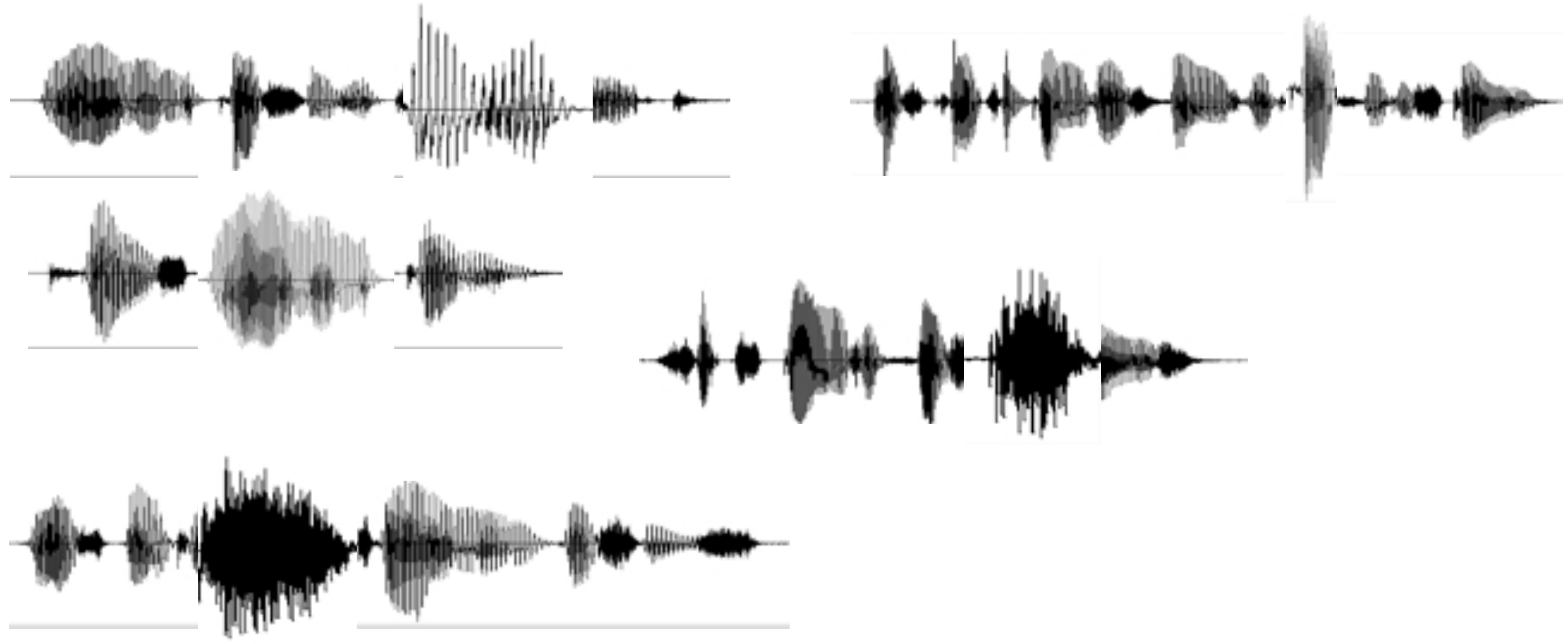
```
NN  Director
IN  of
DT  the
NP  McCormick
NP  Public
NPS Affairs
NP  Institute
IN  at
NP  U-Mass
NP  Boston,
NP  Doctor
NP  Ed
NP  Beard,
VBZ says
DT  the
NN  push
IN  for
VBP do
PP  it
PP  yourself
NN  lawmaking
```



Pronunciation / LTS

- Pronunciation model
  - dictionary look-up, *plus*
  - letter-to-sound model
- <u>But</u>
  - need deep **knowledge** of the language to design the phoneme set
  - human **expert** must write dictionary

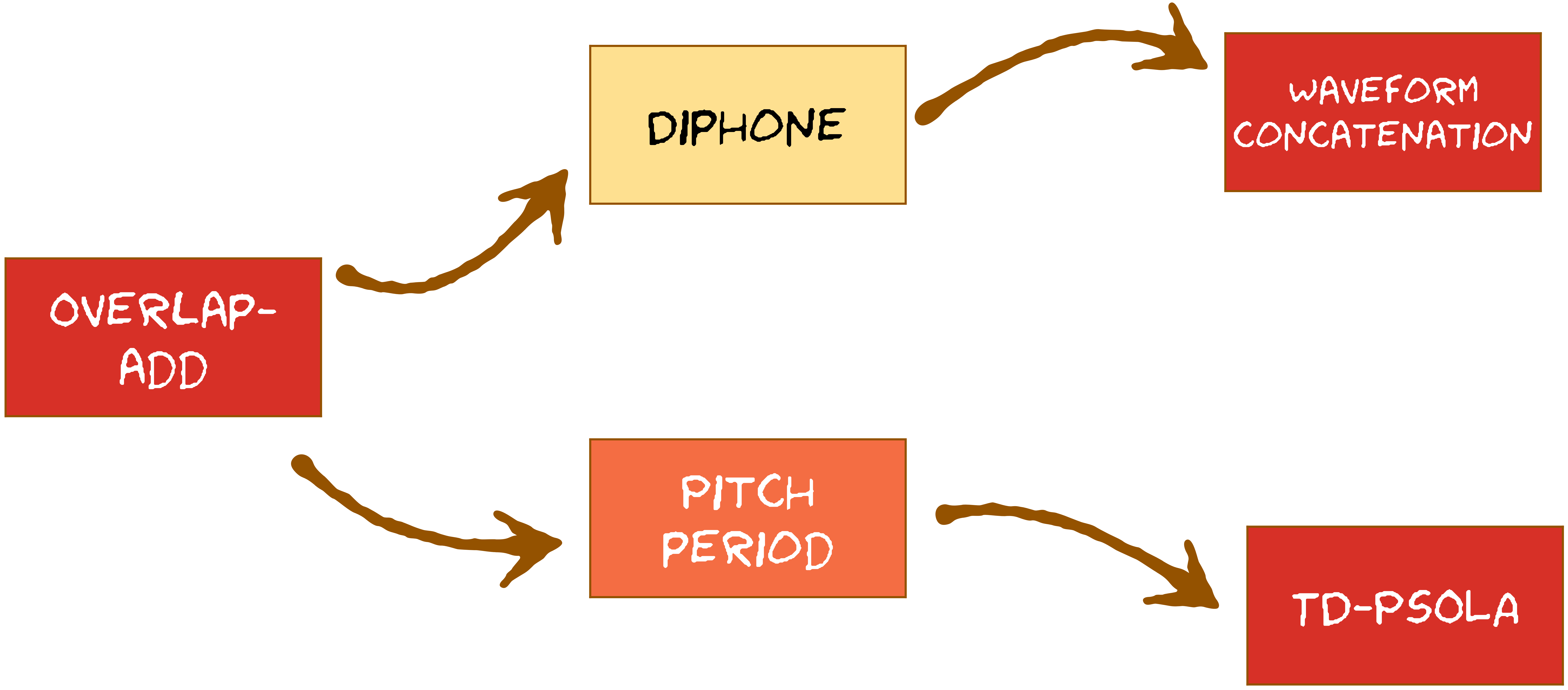
```
ADVOCATING   AE1 D V AH0 K EY2 T IH0 NG
ADVOCATION   AE2 D V AH0 K EY1 SH AH0 N
ADWEEK   AE1 D W IY0 K
ADWELL   AH0 D W EH1 L
ADY   EY1 D IY0
ADZ   AE1 D Z
AE   EY1
AEGEAN   IH0 JH IY1 AH0 N
AEGIS   IY1 JH AH0 S
AEGON   EY1 G AA0 N
AELTUS   AE1 L T AH0 S
AENEAS   AE1 N IY0 AH0 S
AENEID   AH0 N IY1 IH0 D
AEQUITRON   EY1 K W IH0 T R AA0 N
AER   EH1 R
AERIAL   EH1 R IY0 AH0 L
AERIALS   EH1 R IY0 AH0 L Z
AERIE   EH1 R IY0
AERIEN   EH1 R IY0 AH0 N
AERIENS   EH1 R IY0 AH0 N Z
AERITALIA   EH2 R IH0 T AE1 L Y AH0
AERO   EH1 R OW0
```



Key concepts we now understand

- Breaking a complex problem down into simpler steps
- Combining many components into a single architecture
    - representing information in data structures
- The pros and cons of rules vs. learning from data
- Generalising to previously-unseen words or sentences
- Creating new utterances from fragments of pre-recorded speech
- Manipulating the pitch and duration of speech

Today's topics - what we covered



What next?

- Automatic speech recognition

In Modules 6 to 9

- Supported by foundation material on
  - mathematics
  - probability

In next week's foundation class