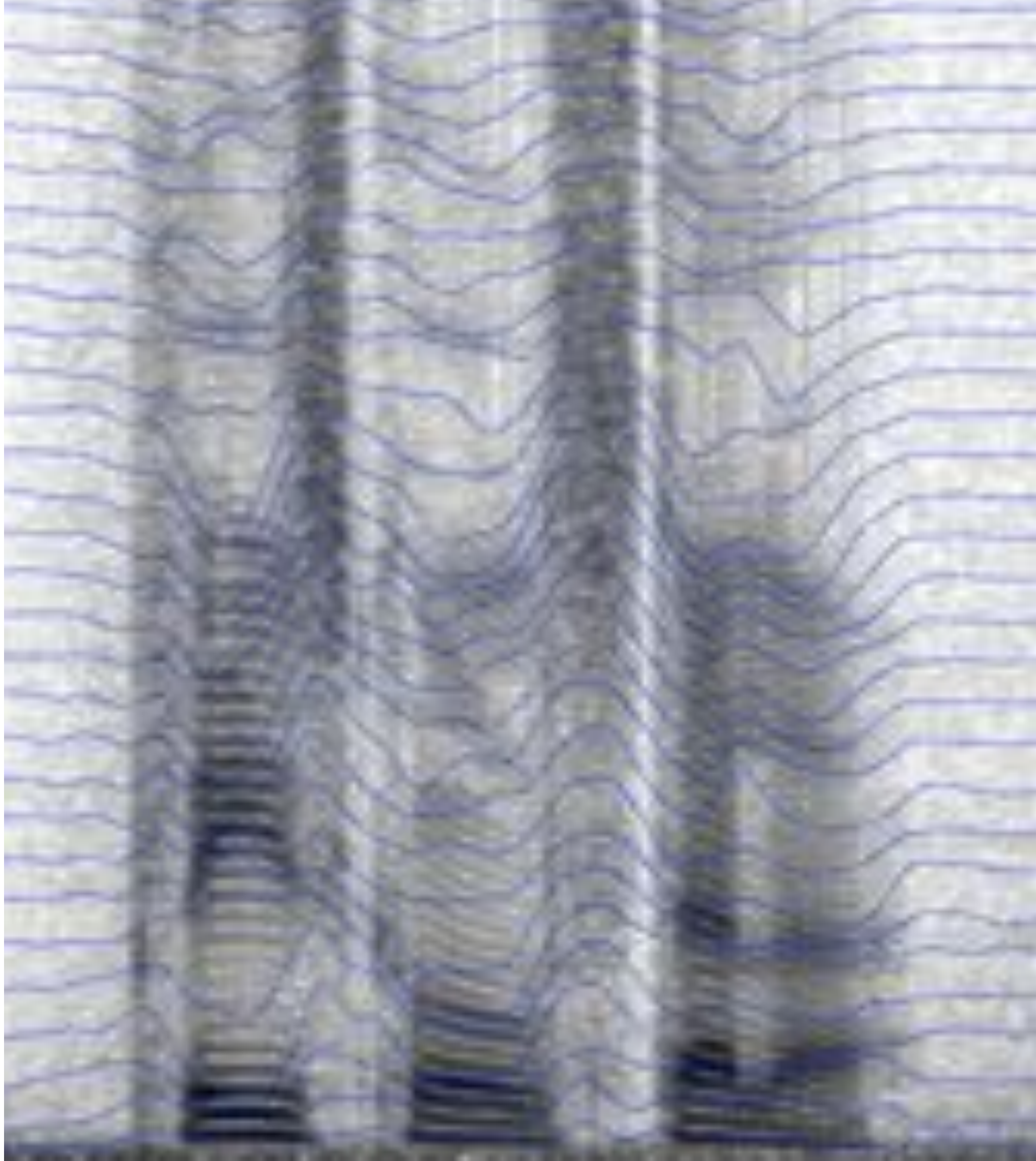
Speech Synthesis

Simon King
University of Edinburgh



Statistical parametric speech synthesis

- text-to-speech as a sequence-to-sequence regression task
- our first model: regression tree + Hidden Markov Model

What you should already know

- <u>Unit selection synthesis</u>
    - how an IFF target cost function uses the linguistic specification, by **querying** each feature individually
    - join cost ensures **continuity** of acoustic features

- <u>Speech signal modelling</u>
    - generalising the source-filter model
    - preparing speech features, ready for statistical modelling

Orientation

- Unit selection
  - selection of waveform units based on
    - target cost
    - join cost

Let's just consider the **IFF** type of target cost, which is based only on the **linguistic specification**

- Speech signal modelling
  - generalised source+filter model

There are several ways to do this, but we need to be able to

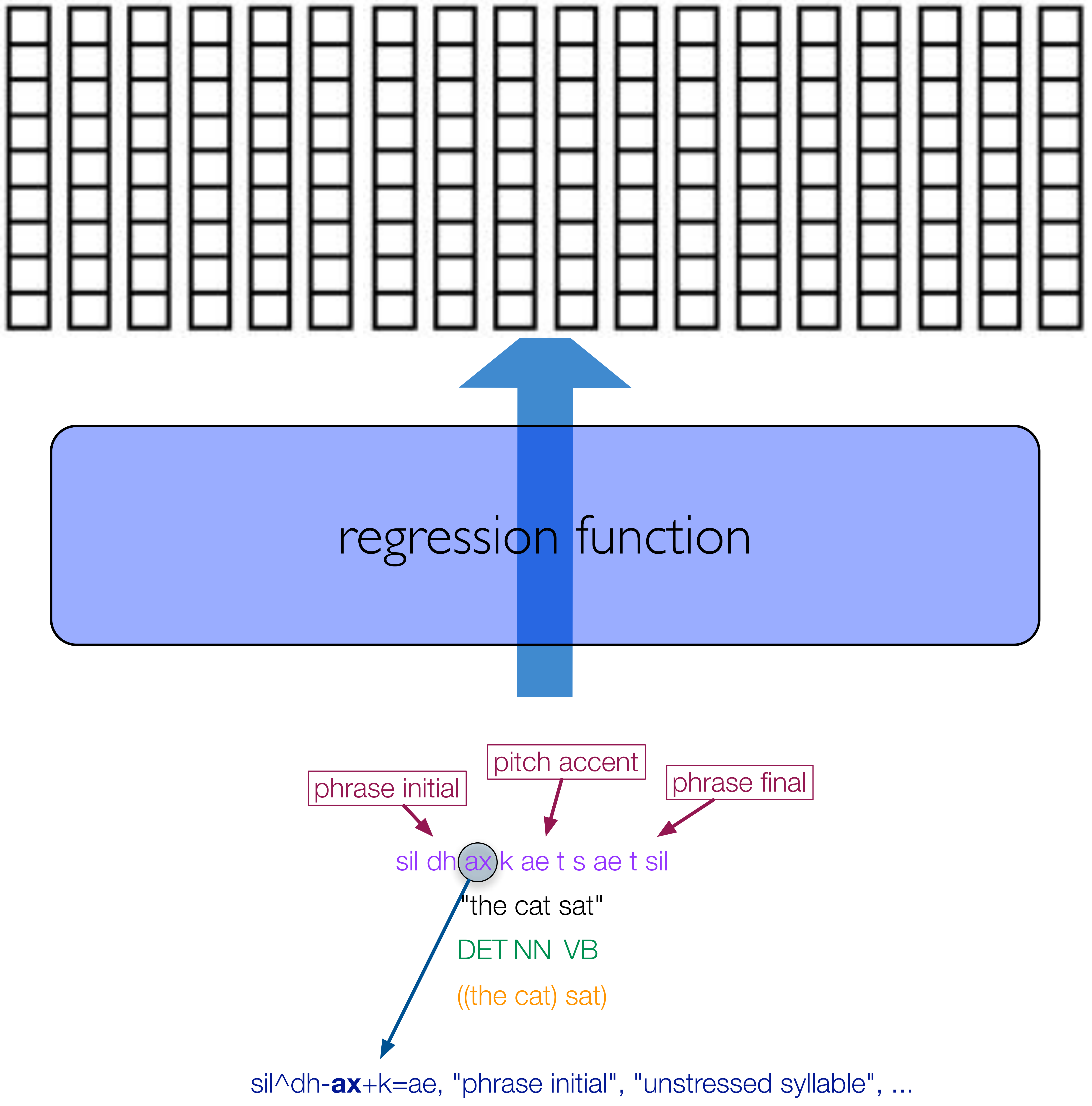
- **separate** excitation & spectral envelope
- **reconstruct** the waveform

- Statistical parametric synthesis
  - predict **speech parameters** from **linguistic specification**

A **regression** task!

Orientation

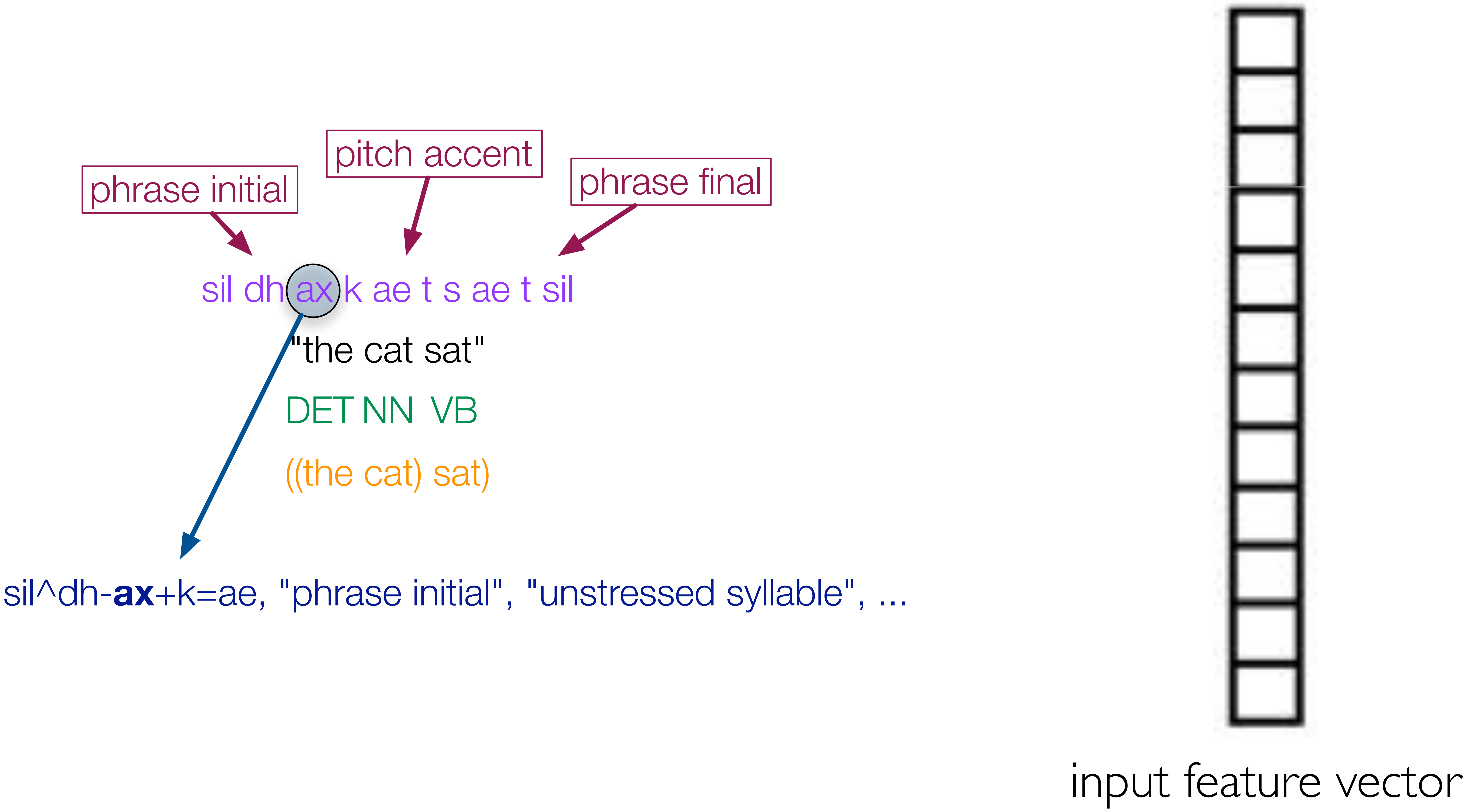
- Statistical parametric synthesis
  - predict **speech parameters** from **linguistic specification**



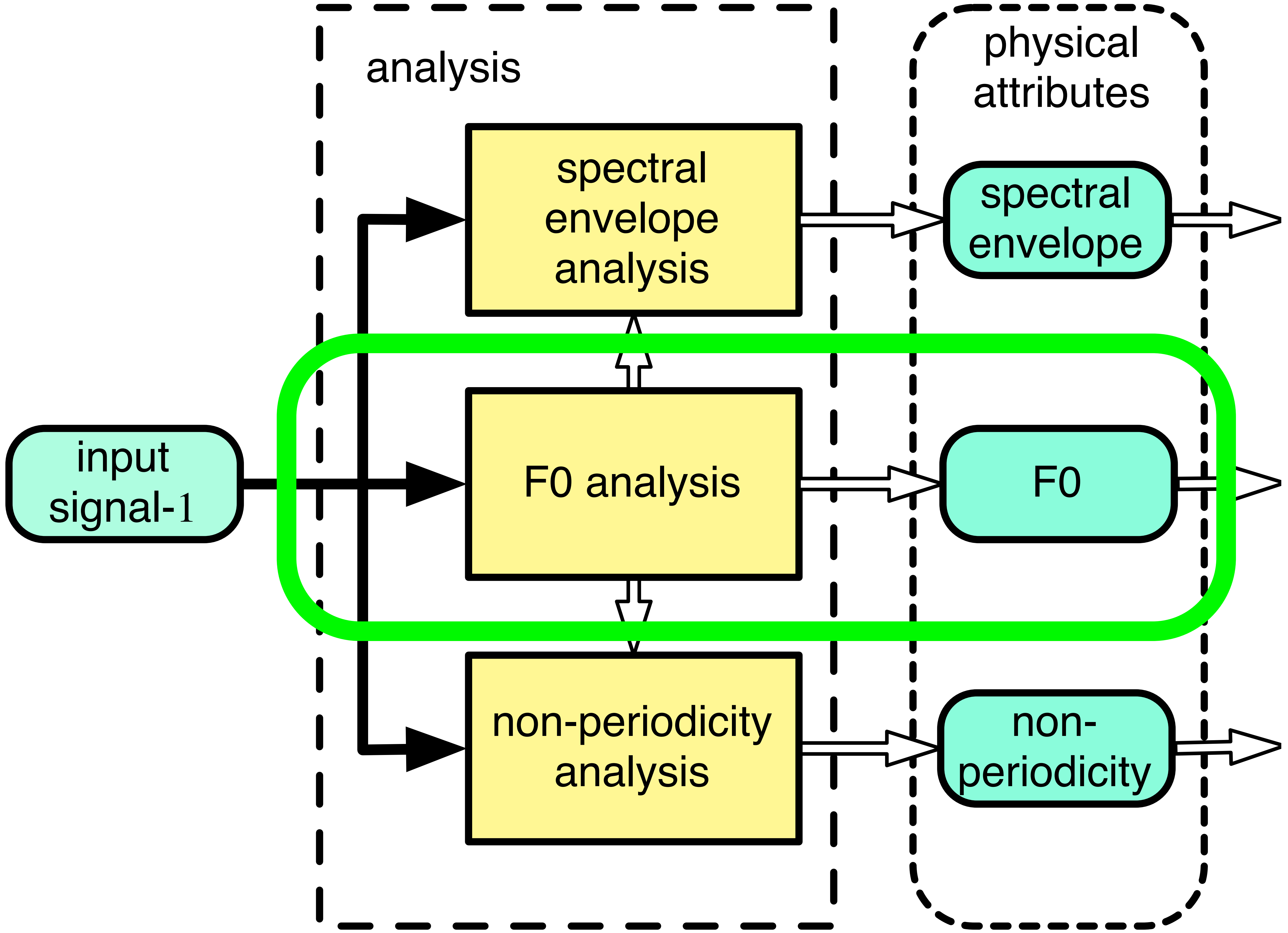
Statistical parametric speech synthesis

- text-to-speech as a sequence-to-sequence regression task
- our first model: regression tree + Hidden Markov Model

What are the input features ? Just the linguistic features !



What are the output features (i.e., speech parameters) ?

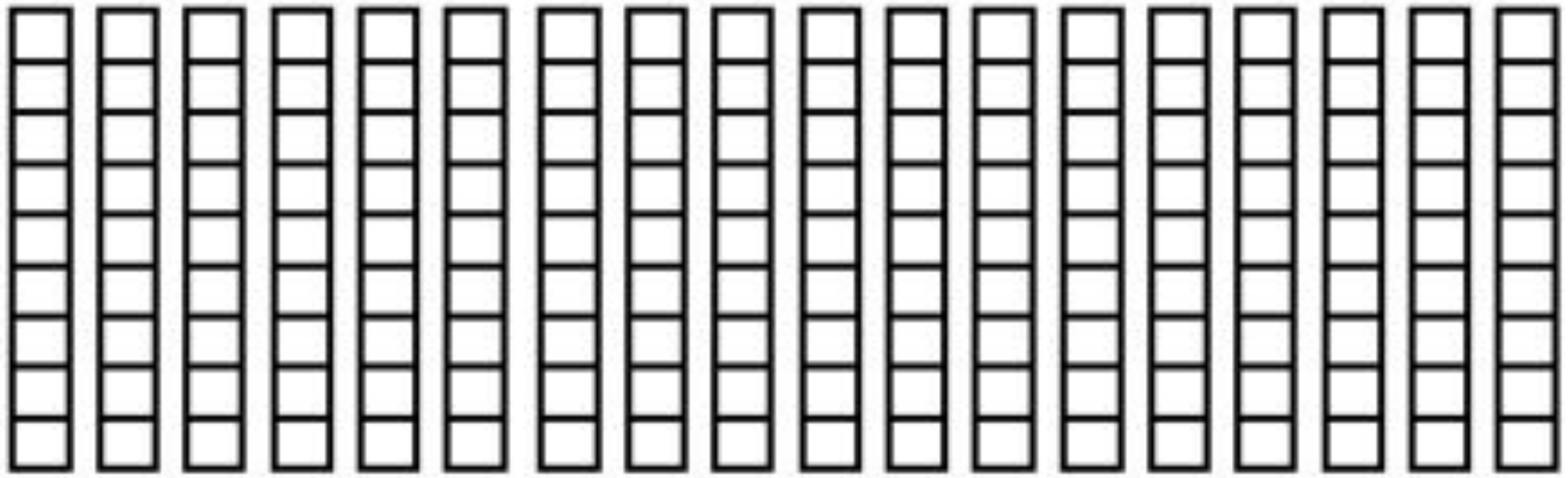


speech parameters

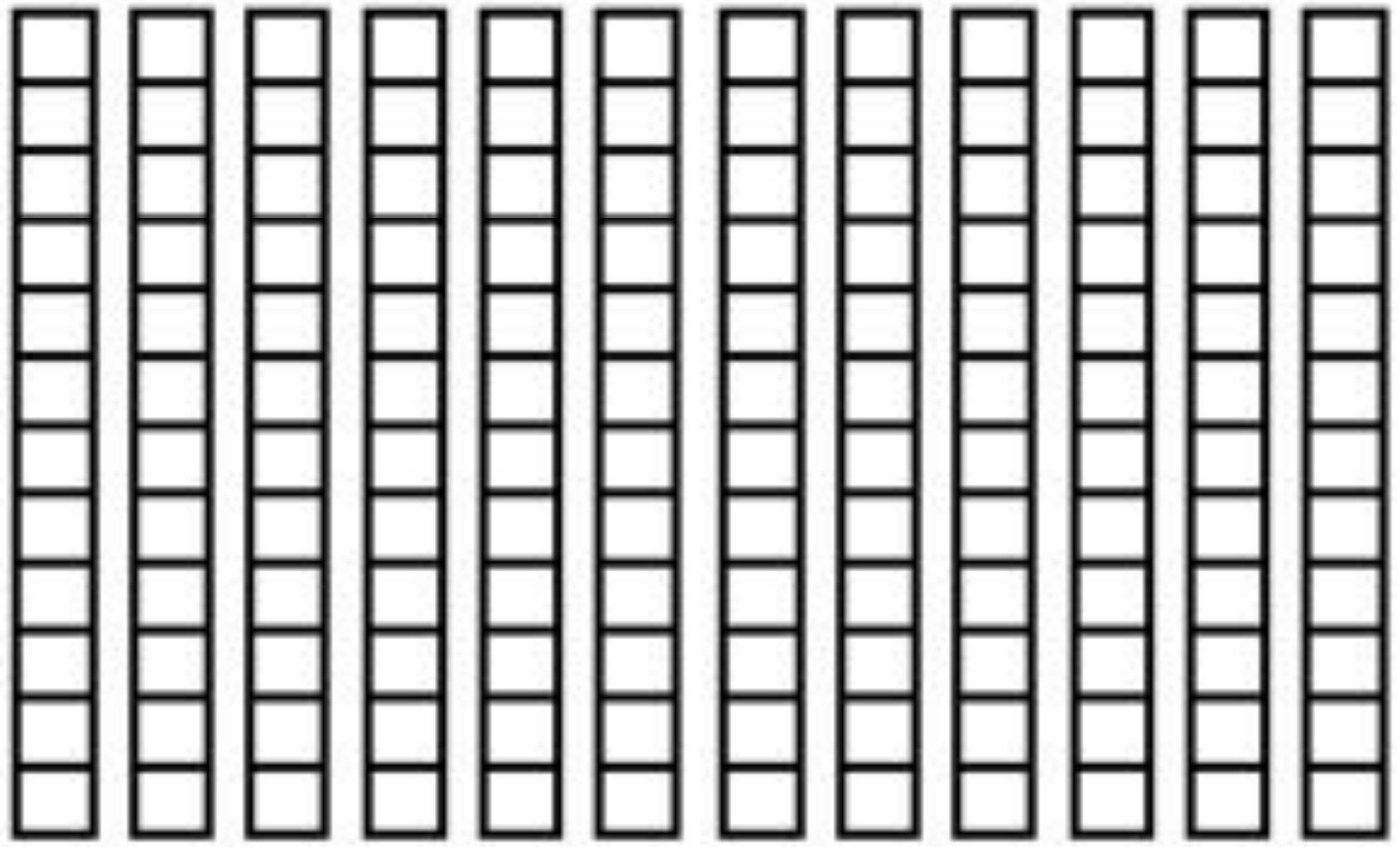
output feature vector

The **sequence-to-sequence** regression problem

output sequence



input sequence



Statistical parametric speech synthesis

- text-to-speech as a sequence-to-sequence regression task
- <u>our first model: regression tree + Hidden Markov Model</u>

Our first model: regression tree + Hidden Markov Model

- Two complementary explanations
  - regression
  - context-dependent models
- Duration modelling
- Generation from the model

Two complementary explanations

- Describing synthesis as a regression task
  - **prediction** of continuous speech parameters from linguistic features
- Practical implementation using context-dependent models
  - **create** *lots* of models: oops! for many, there is **no training data**
  - fix this by **sharing** parameters with existing models ("tying")

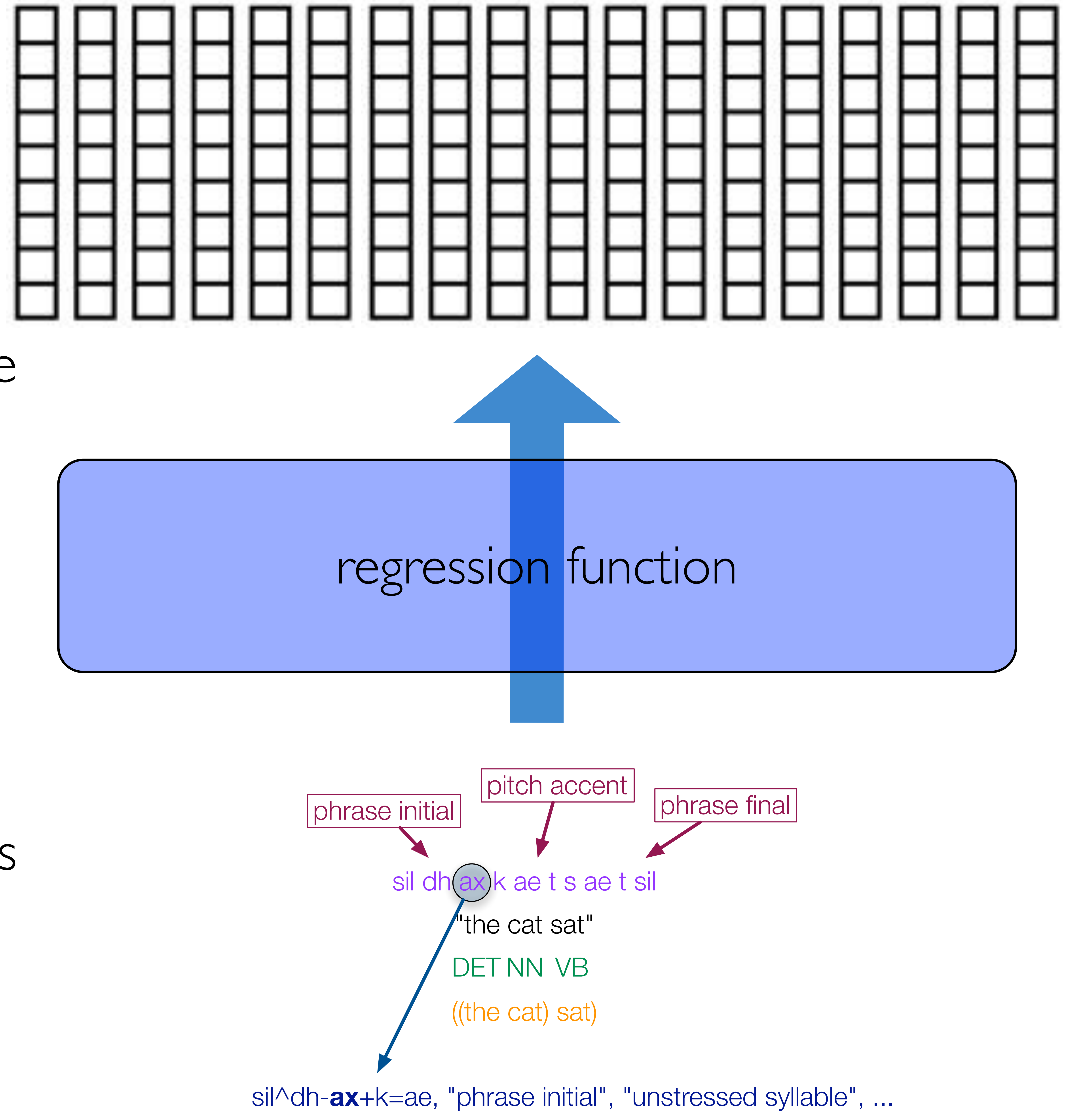
regression

context-dependent modelling

Two tasks to accomplish

- Sequencing
  - progress through the phonetic sequence
  - decide durations
  - create a sequence of frames

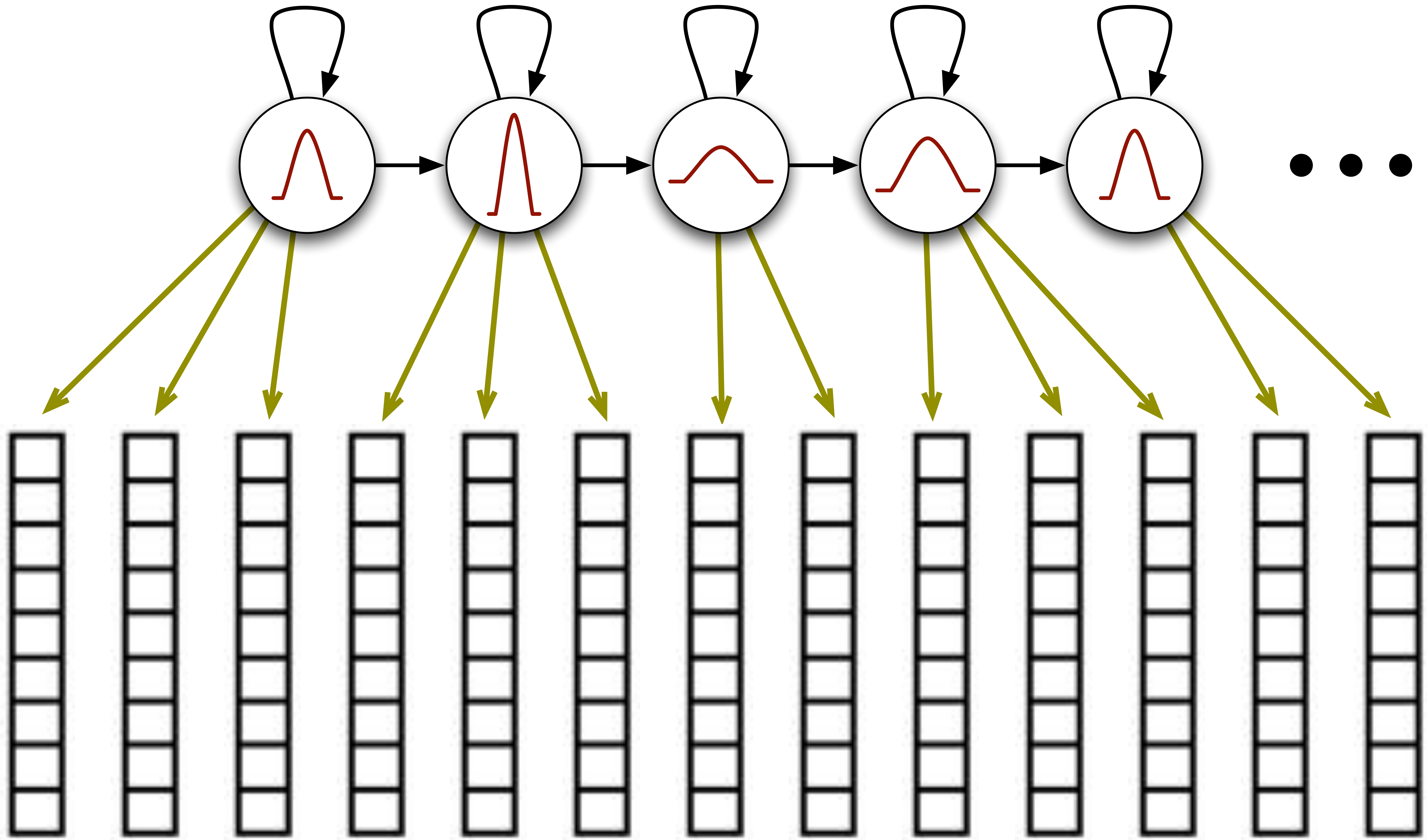
- Prediction (regression)
  - Given the local linguistic specification, predict one frame of speech parameters



Choose suitable machinery for each task

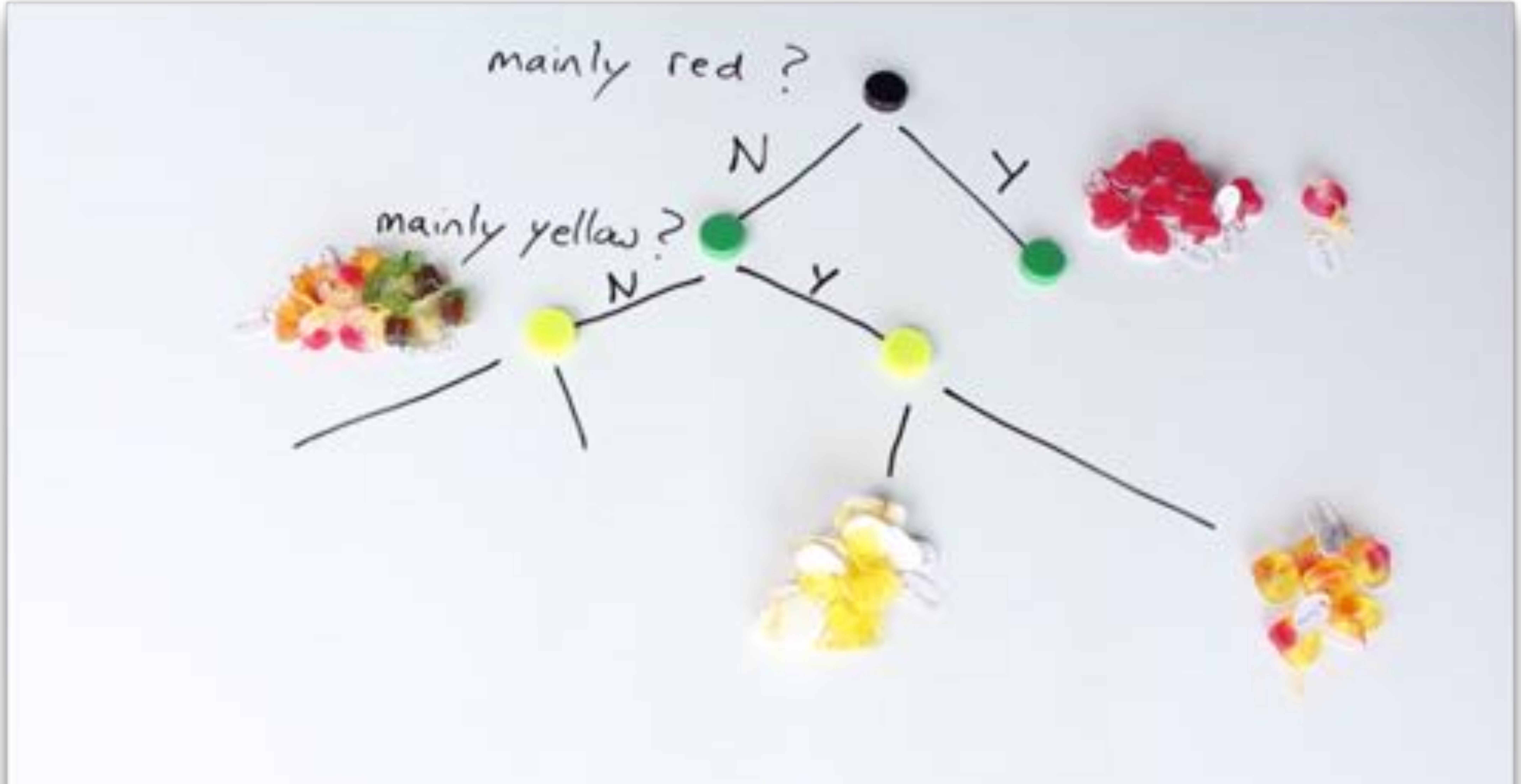
- Sequencing
  - **Hidden Markov Model**
  - Why? It's the simplest model we know, that can generate sequences!

- Regression
  - **Regression tree** (i.e., a CART with continuously-valued predictee)
  - Why? Again, the simplest model we know, that can learn an arbitrary function
    - *the mapping from linguistic specification to speech spectrum is surely non-linear*



Regression

Reminder: CART



HMM for sequencing + **regression tree** for prediction

pitch accent

phrase initial

phrase final

sil dh ax k ae t s ae t sil

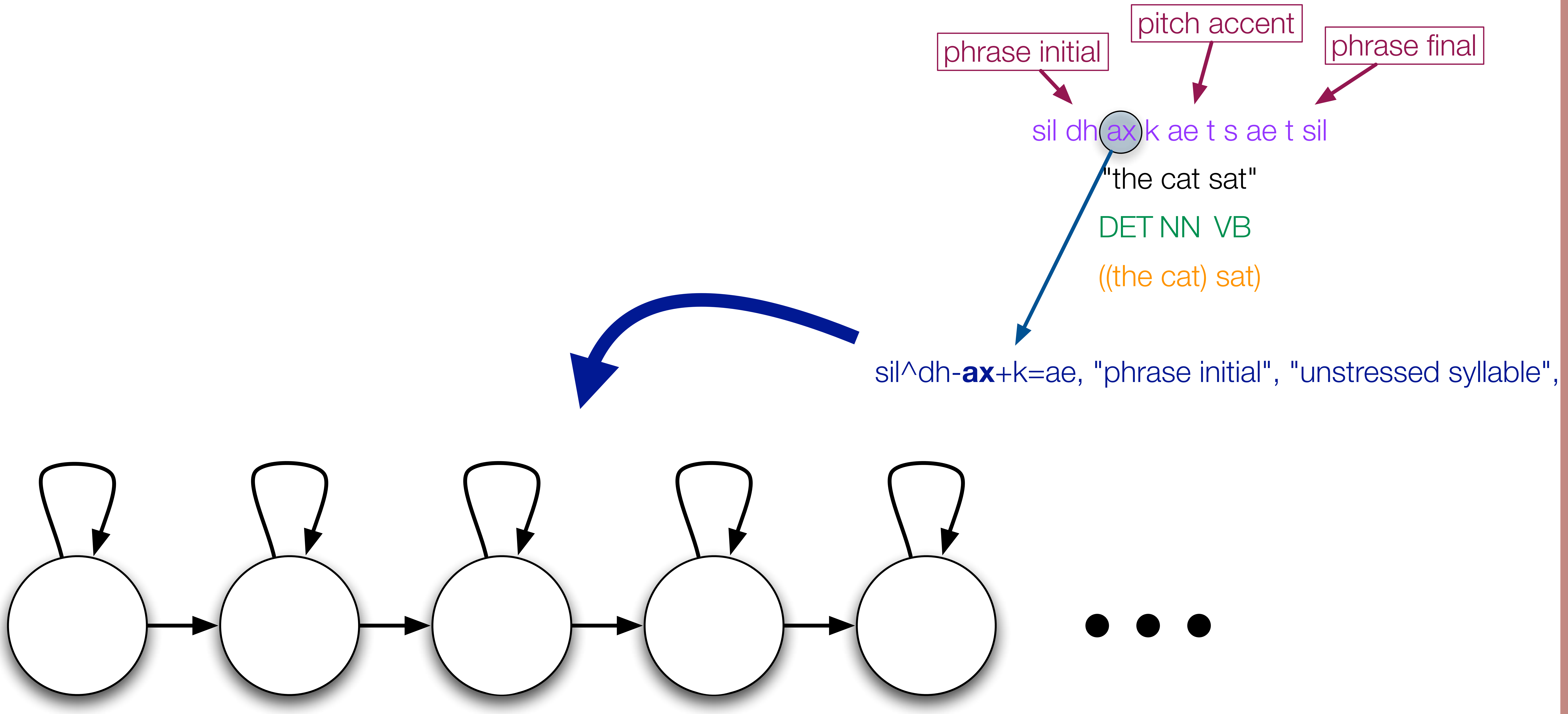
"the cat sat"

DET NN VB

((the cat) sat)

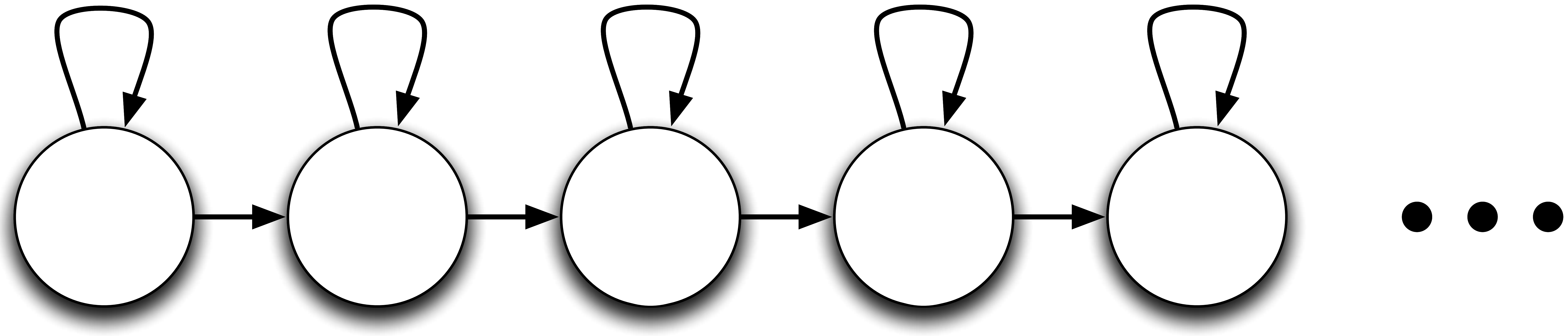
sil^dh-**ax**+k=ae, "phrase initial", "unstressed syllable", ...

HMM for sequencing + **regression tree** for prediction



Regression

HMM for sequencing + **regression tree** for prediction



sil^dh-**ax**+k=ae, "phrase initial", "unstressed syllable", ...

Two complementary explanations

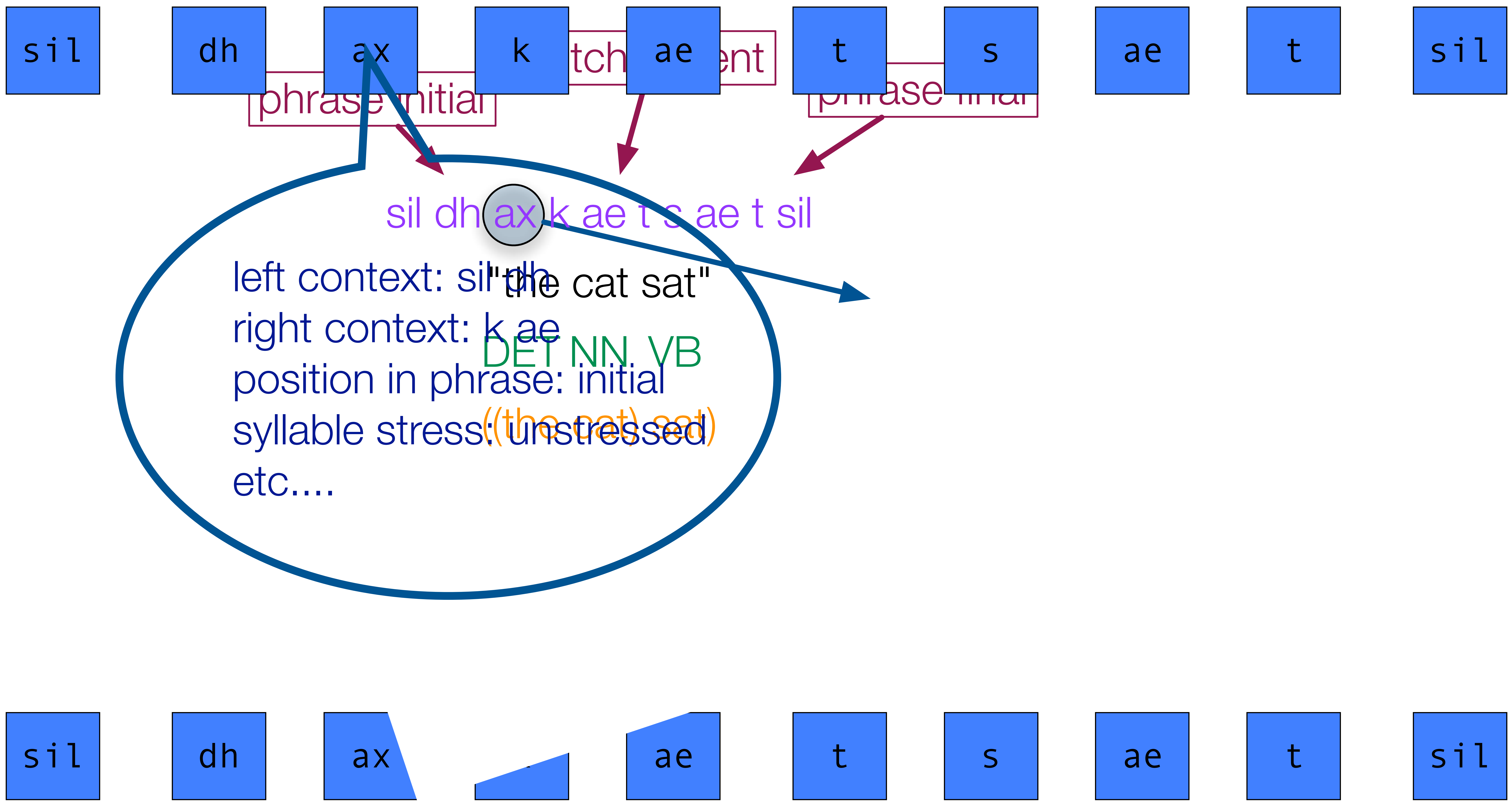
- Describing synthesis as a regression task
  - **prediction** of continuous speech parameters from linguistic features

regression

- Practical implementation using context-dependent models
  - **create** *lots* of models: oops! for many, there is **no training data**
  - fix this by **sharing** parameters with existing models ("tying")

context-dependent modelling

Reminder: constructing the target unit sequence (for unit selection)



From linguistic specification to sequence of models

"Author of the ..."

```
sil~sil-sil+ao=th@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$.....
sil~sil-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
sil~ao-th+er=ah@2_1/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
ao~th-er+ah=v@1_1/A:1_1_2/B:0-0-1@2-1&2-6#1-4$.....
th~er-ah+v=dh@1_2/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
er~ah-v+dh=ax@2_1/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
ah~v-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
v~dh-ax+d=ey@2_1/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
```

Context-dependent modelling

- We cannot be sure to have examples of every unit type in every possible context in the training data
- In reality, the context is so rich (it spans the whole sentence), that almost every single token in the training data is the only token of its type

- Two key problems to solve
  - train models for types that we have **too few** examples of (e.g., 1)
  - create models for types that we have **no examples** of

- Joint solution: parameter sharing amongst groups of similar models

Training models for types that we have too few examples of

- We *could* train a model on just a single example (= single token)
- But it will be very poorly estimated
    - unlikely to perform well

- **Pooling training data** across groups of types will increase amount of data available
- How to decide **which groups** of models should share data?
    - i.e., which groups of models will end up with the same parameters

Some contexts exert similar effects

- Key insight
  - we can group *contexts* according to the effect that they have on the centre phoneme
  - for example
    - the [ae] in the contexts p-ae+t and b-ae+t may be very similar
  - how to group these contexts?
    - how to represent them so we can form useful groupings?

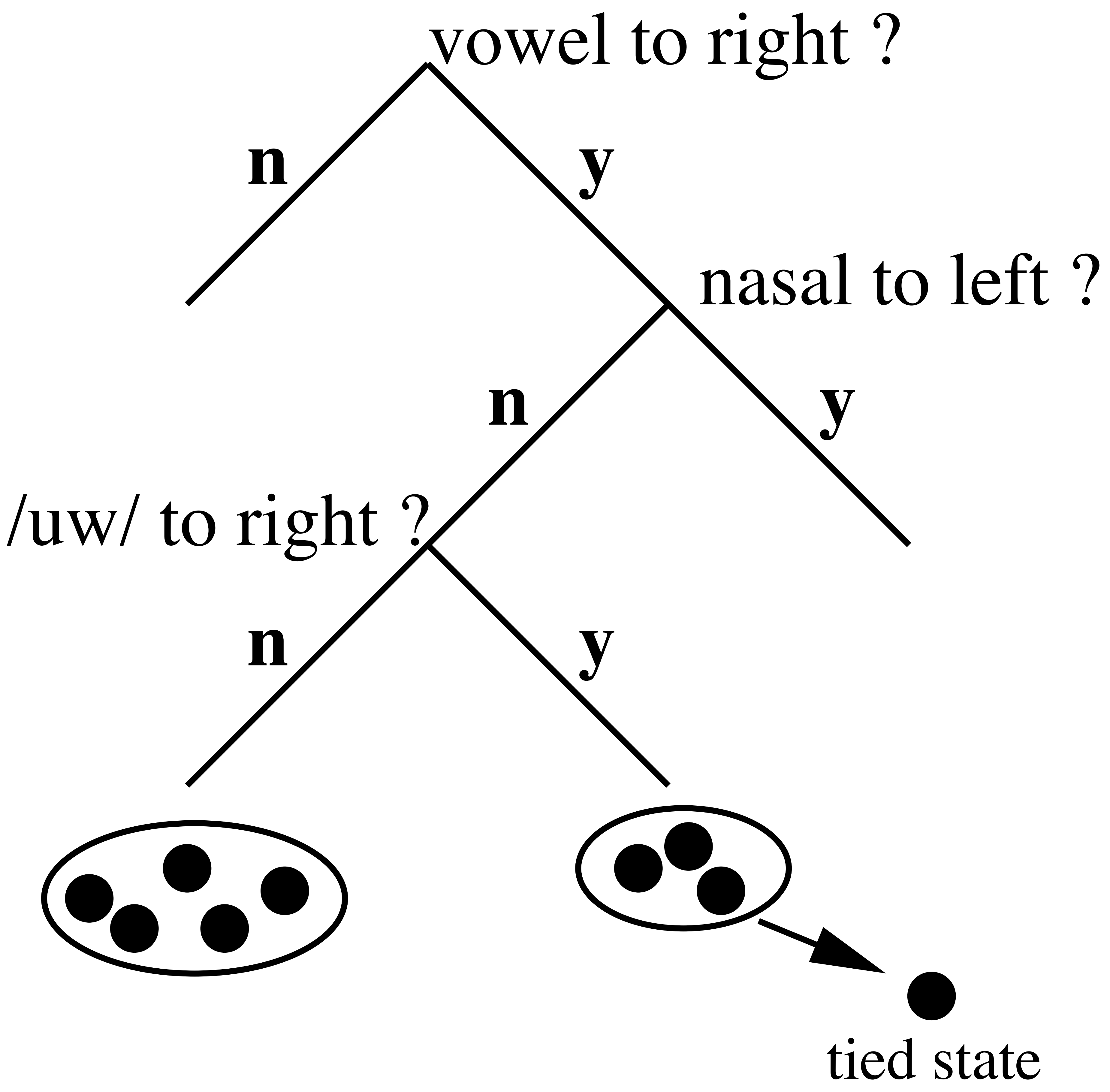
  - **use the phonetic features of the surrounding context**
    - place, manner, voicing, ...

Grouping contexts according to phonetic features

- Could try to write rules to express our knowledge of how co-articulation and other context effects work
  - *"all bilabial stops have a similar effect on the following vowel"*
  - *"all nasals have a similar effect on the preceding vowel"*
  - *... etc*
- Of course, it's better to learn this from the data, for 2 reasons
  - find those groupings that actually make **a difference to the acoustics**
  - **adjust the granularity** of the groups according to how much data we have

- But we still want to make use of our **phonetic knowledge**

Combining phonetic knowledge with data-driven learning

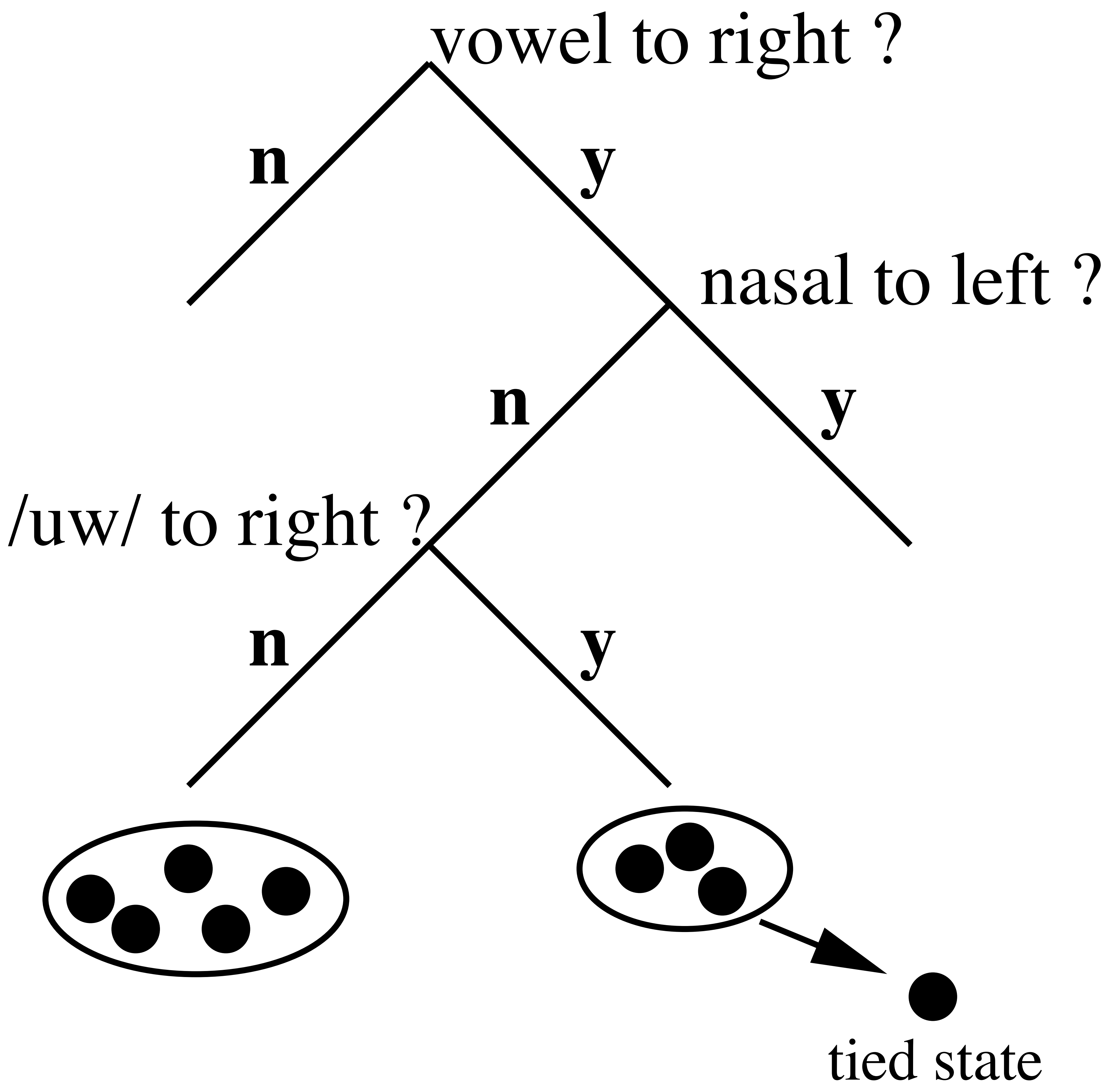


How to choose the best split

- Ideal measure
  - a) train a single model on data pooled across the unsplit set of contexts
  - b) train two models: one on each split of the data
  - compare the **likelihood increase** from a) to b)
- This is not feasible in practice - too computationally-expensive
  - cannot retrain models for every possible split, at every node in the tree
- Instead, use an **approximation** to the likelihood increase
  - this can be computed without actually retraining any models
  - only requires access to the state occupancy statistics and Gaussian parameters

What about models for unseen contexts?

- To find out which model to use for a particular context
  - just follow the tree from root to leaf, answering the questions
- Crucially, to do this we only need to know the **name** of the model, in order to answer those questions
- So it works for models which have training data, and also for models that don't




```
ah~v-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
```

Summary: linguistic processing, training, synthesis

- Linguistic processing
  - from text to linguistic features using the **front end** (same as in unit selection)
  - attach linguistic features to phonemes: **"flatten"** the linguistic structures

  - we then create one context-dependent HMM for **every unique combination** of linguistic features

Summary: linguistic processing, training, synthesis

- Training the HMMs
  - need **labelled** speech data, just as for ASR (supervised learning)
  - need models for all combinations of linguistic features, including those **unseen** in the training data
    - this is achieved by parameterising the models using a regression tree

Summary: linguistic processing, training, synthesis

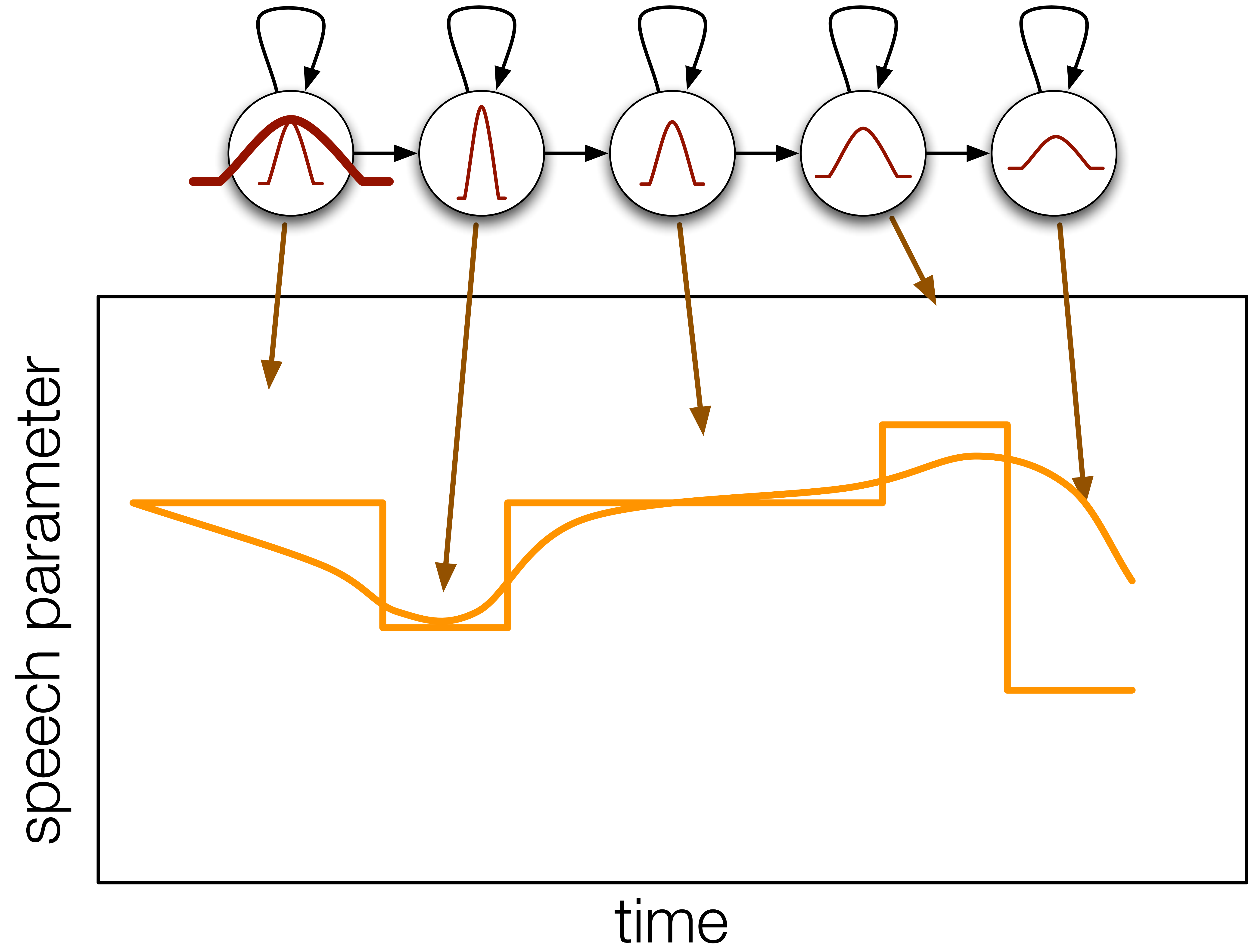
- Synthesising from the HMMs
  - use the front end to predict required **sequence** of context-dependent models
    - the regression tree provides the **parameters** for these models
  - use those models to **generate** speech parameters
  - use a **vocoder** to convert those to a waveform

Generating from the regression tree + Hidden Markov Model

- This should be straightforward, because the HMM is a generative model
- Follow the Maximum Likelihood principle
  - generate the **most likely** output
  - that will simply be the sequence of state **means**

- What about duration?
  - we need a model to predict this
  - let's just use another regression tree, predicting duration **per state**
    - <u>predictors:</u> linguistic context + state-position-within-phone
    - <u>predictee:</u> duration of the current state, in frames

Trajectory generation



Orientation

- Our **first attempt** at statistical parametric speech synthesis
  - we used models that we are familiar with and understand well

This is perfectly sensible: we have **good algorithms** for training the models, for example.

- Regression trees are weak models

The key weakness of the method. We must replace the regression tree with something more powerful.

- Although Gaussians are convenient
  - e.g., so we can borrow many useful techniques from ASR

e.g., model adaptation

What next?

- **Better regression model**
  - a Neural Network
  - input & output features essentially the same as regression tree + HMM

- Quality will still be limited by the **vocoder**

- Later, we will also address that problem
  - hybrid synthesis
  - direct waveform generation