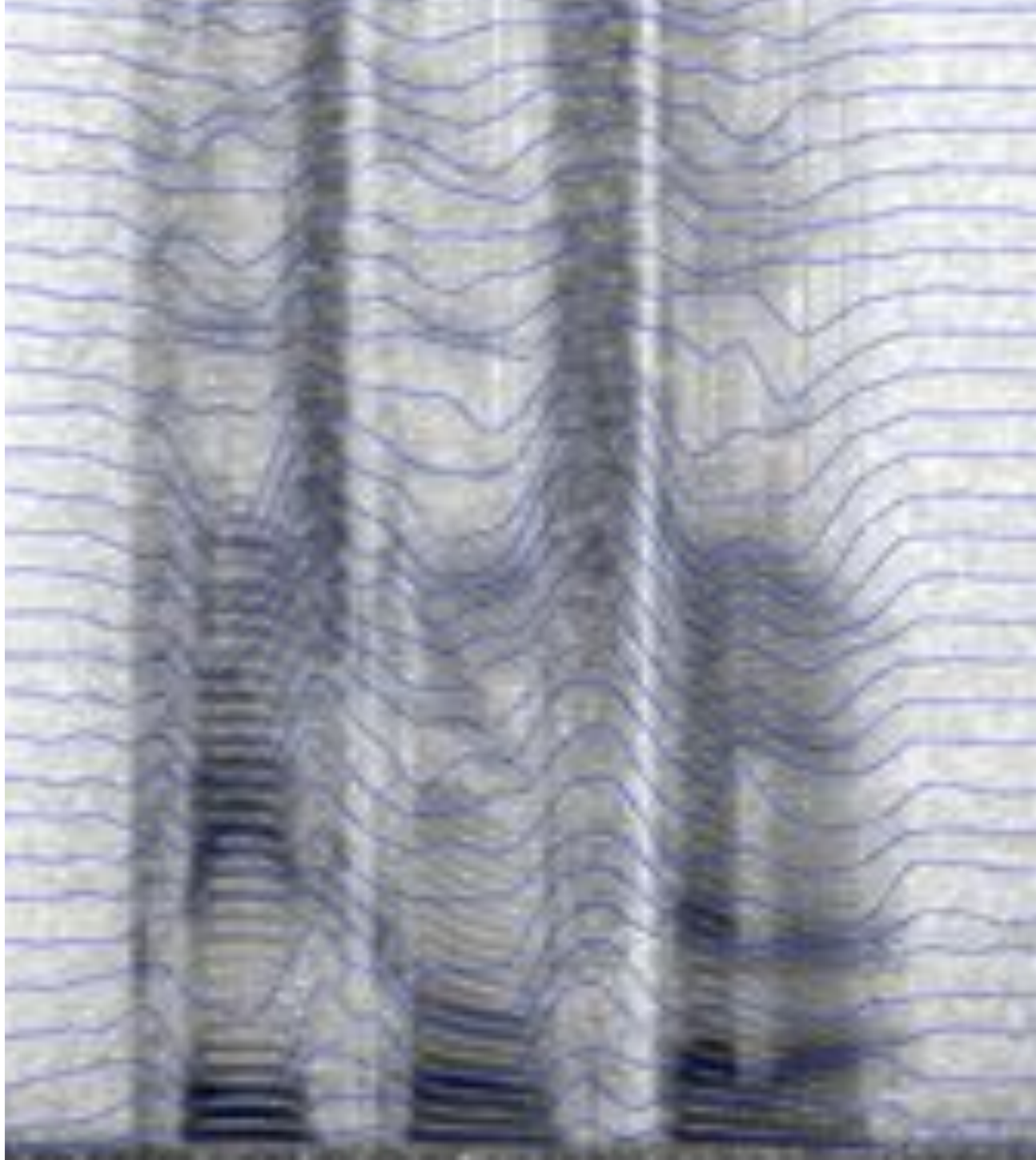
Speech Synthesis

Simon King
University of Edinburgh



Unit selection

- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming

What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming

What you should already know

- phonemes
  - place, manner, voicing, etc

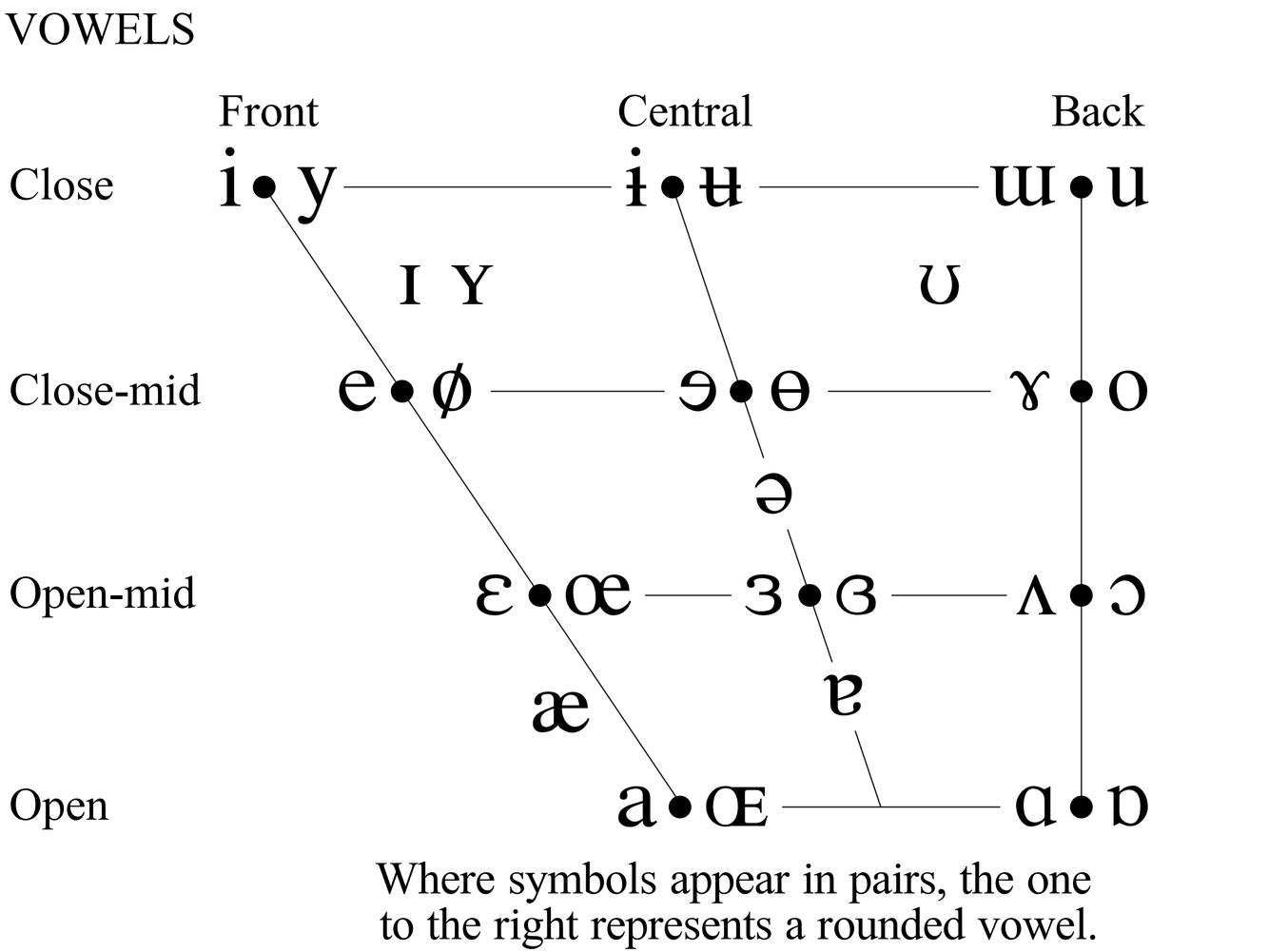
CONSONANTS (PULMONIC) © 2015 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

What you should already know

- phonemes
  - place, manner, voicing, etc

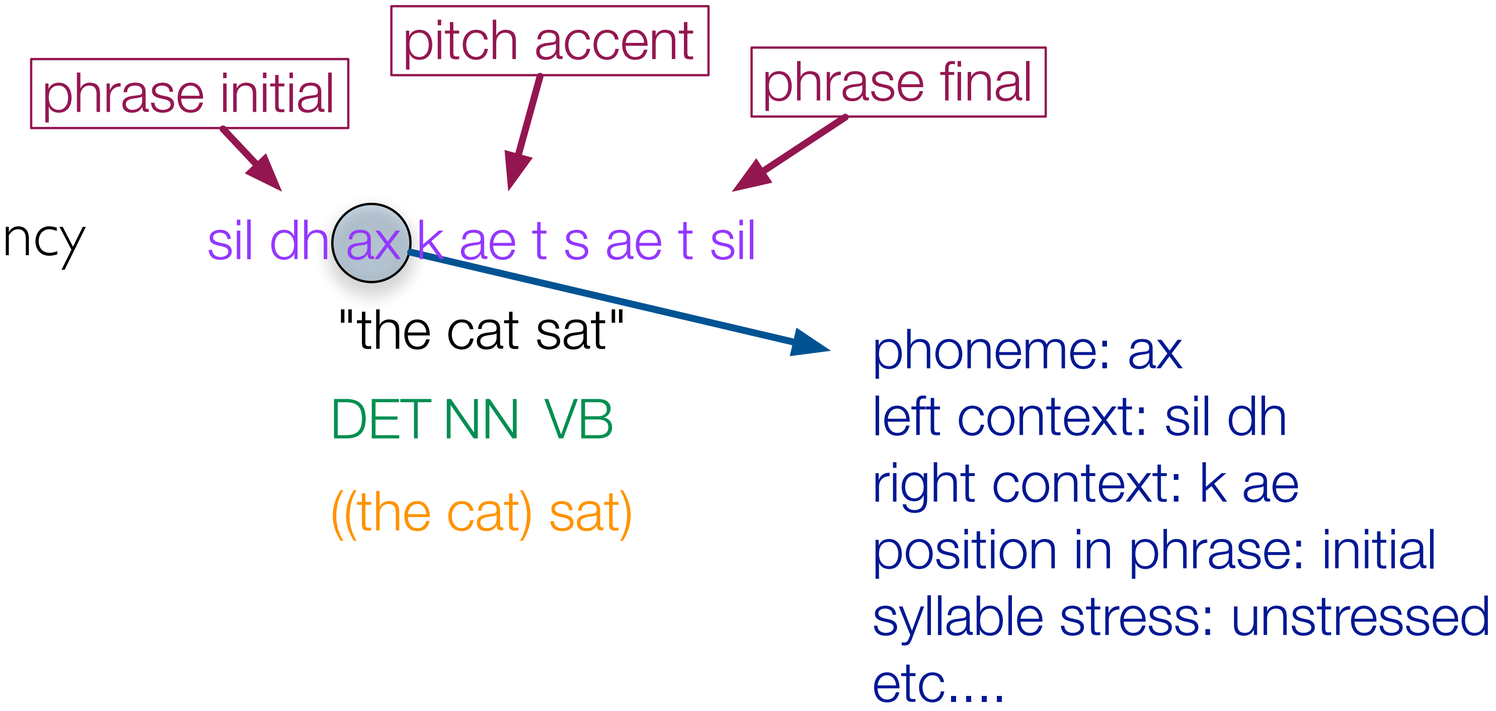


What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming

What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming



What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming

What you should already know

- phonemes
  - place, manner, voicing, etc
- source-filter model
  - F0, formants, vocal tract frequency response
- front-end text processing
  - linguistic specification
- diphone speech synthesis
  - database contains only one recording of each diphone type
- dynamic programming

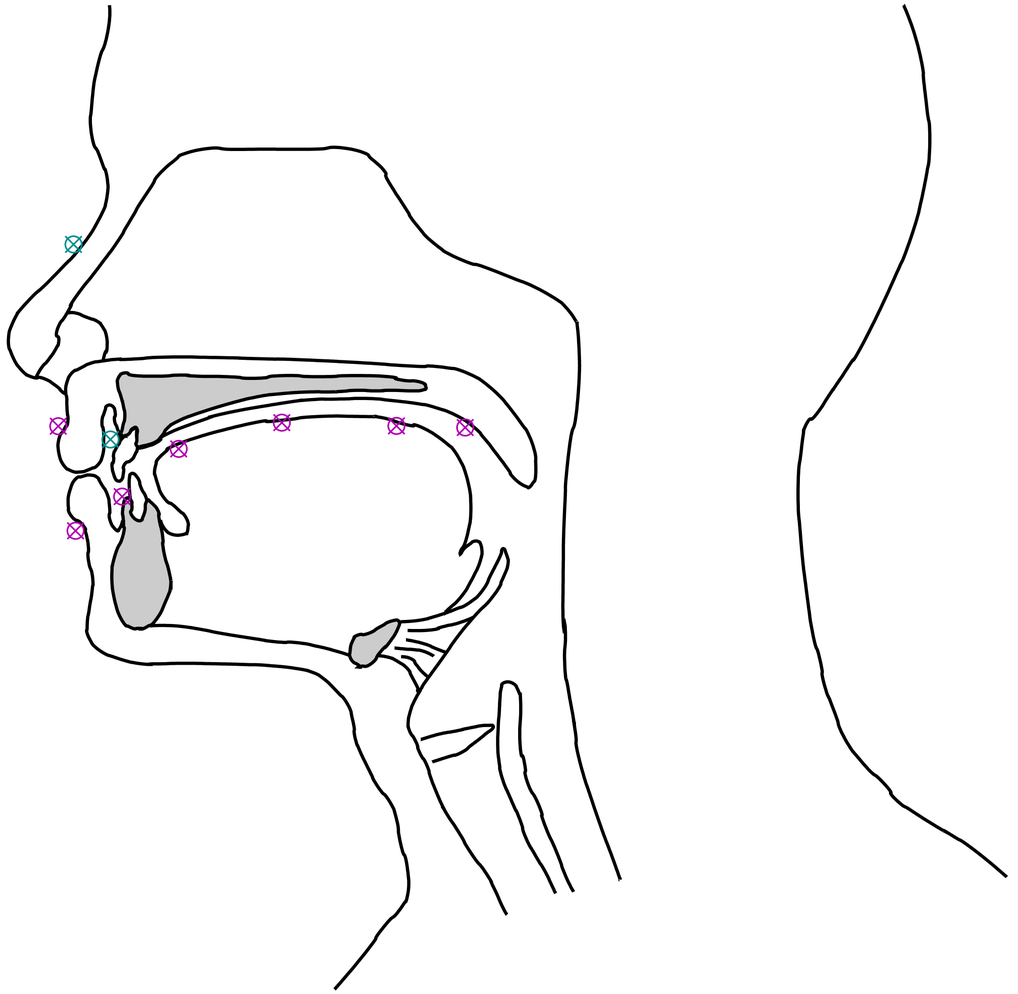
Unit selection

- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

Speech production

- Observed signal is result of several **interacting processes**

- The **context** in which a sound is produced **affects** that sound
  - articulatory trajectories
  - phonological effects
  - prosodic environment



What units should we divide speech into?

- The speech signal we observe (the waveform) is the product of interacting processes operating at **different time scales**
  - at any moment in time, the signal is affected not just by the current phoneme, but many other aspects of the context in which it occurs
  - the context is complex - it's *not just the preceding/following sounds*

- How can we reconcile this conflict?
- We want to simultaneously:
  - model speech as a string of units (so that we can *concatenate waveform fragments*)
  - take into account the effects of context, before/during/after the current moment in time

Context is the key

- **Context-dependent units** offer a solution
  - engineer the system in terms of a simple linear string of units
  - account for context with a different **version** of each unit for every different context

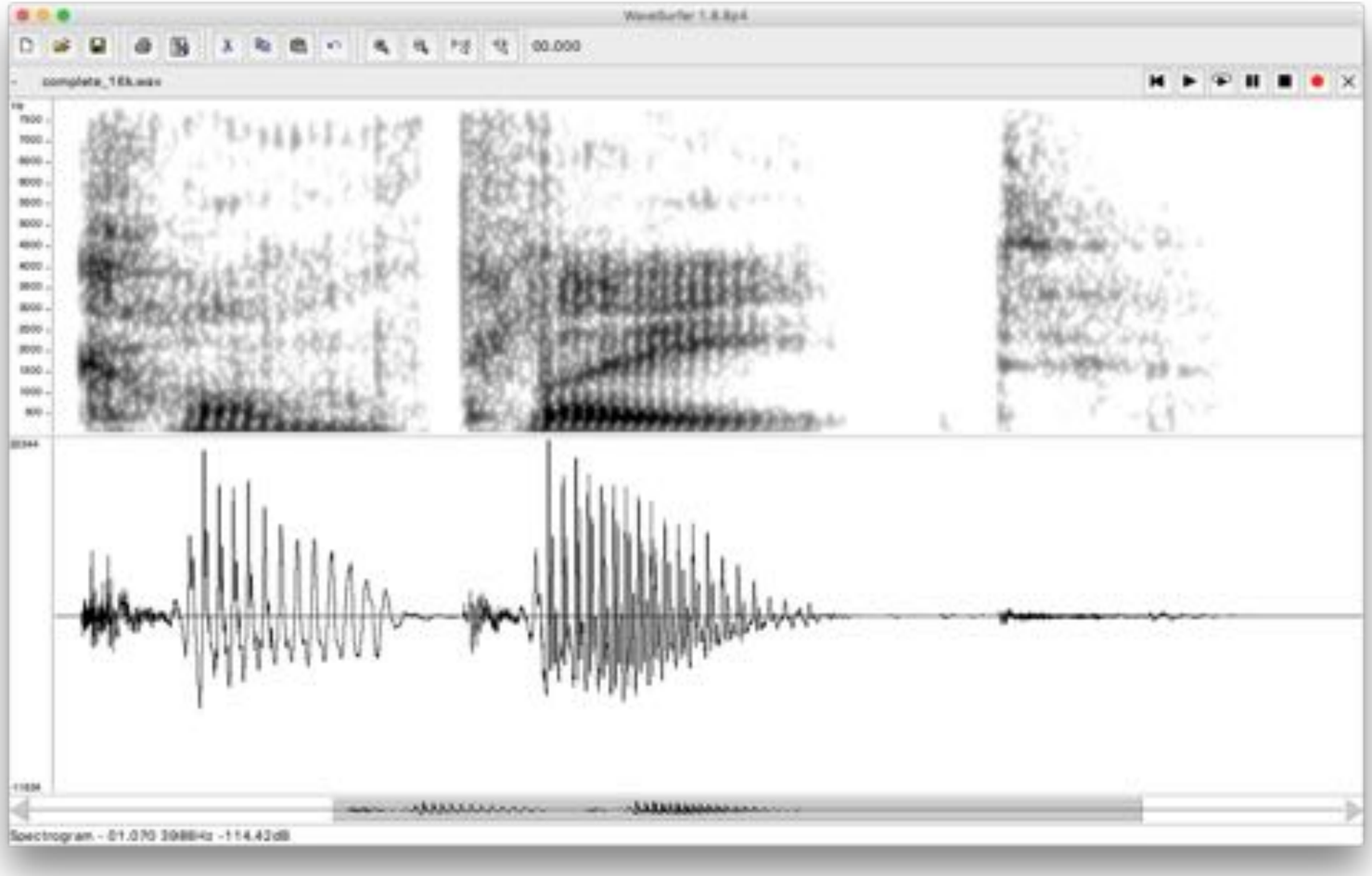
- But, how do we know what **all the different contexts** are?
- If we **enumerate** all possible contexts, they will be practically infinite
  - there are an infinite number of different sentences in a language
  - context potentially spans the whole sentence (or further)

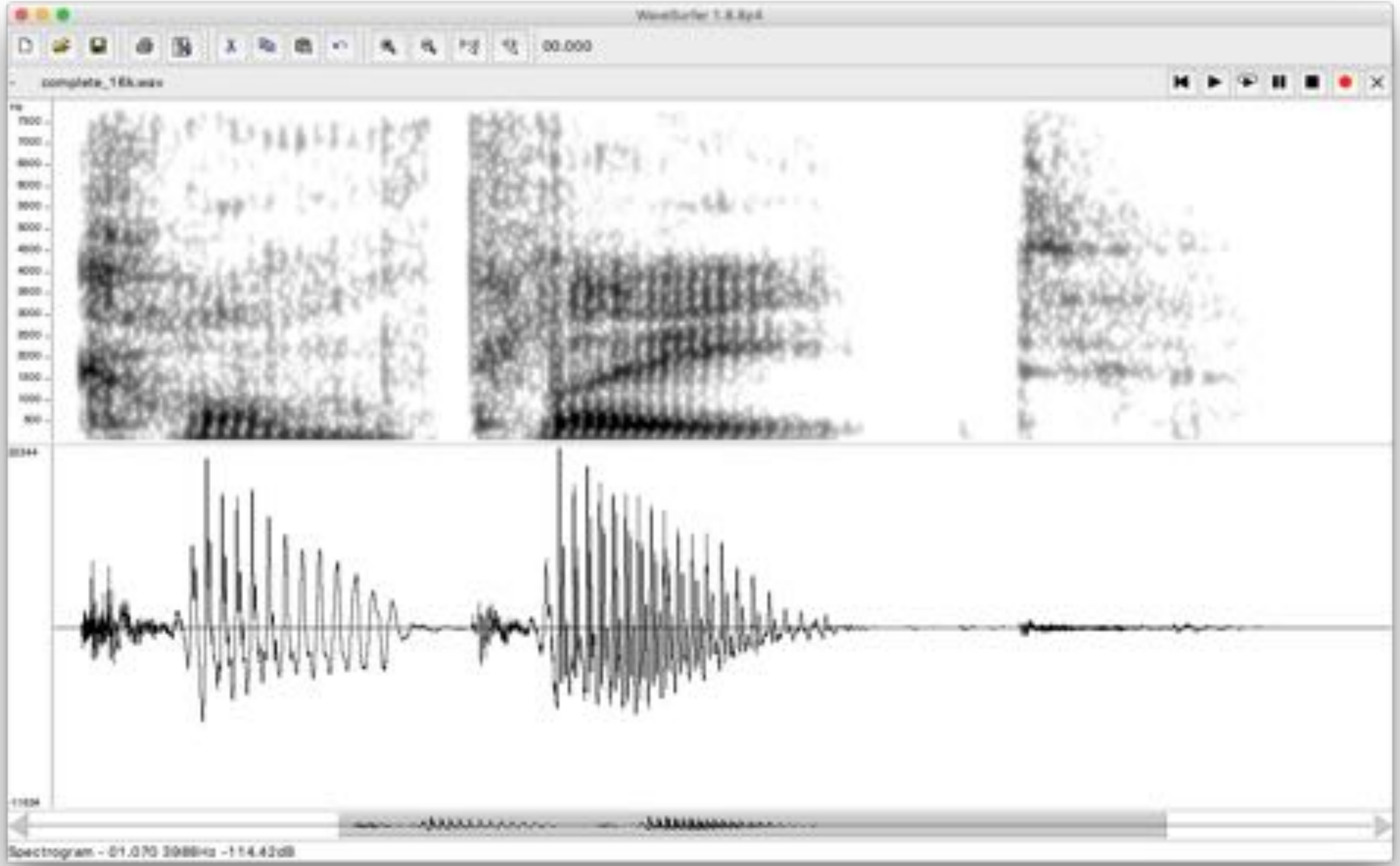
- Fortunately, what is important is the **effect** that the context has on the current speech sound - so next we can think about reducing the number of **effectively different** contexts

First solution: diphones

- Assume that the only context that affects the current sound is the identity of the
  - preceding phone, and
  - following phone

- Can be handled easily
  - **diphone units**
  - number of unit types $O(N^2)$




complete_16k.wav
00.000
Spectrogram - 01.070 3988Hz -114.42dB

Problems with diphone synthesis

- Signal processing is required to manipulate:
  - **F0** & **duration**: fairly easy, within a limited range
  - **Spectrum**: not so easy, can do simple smoothing at the joins but otherwise it's not obvious what aspects to modify

- But, this extensive signal processing
  - introduces artefacts and **degrades the signal**
  - cannot faithfully replicate every detail of natural variation in speech
    - we don't know **what** to replicate
    - we don't have powerful enough techniques to **manipulate** every aspect of speech

Reduce the need for manipulation by *increasing* the number of unit types?

- With **multiple versions** of each diphone, could choose one needing **least manipulation**

- How about stressed and unstressed versions of every diphone
  - database size *doubles* to 2000-4000 units
- Phrase-final / non-final versions
  - database size *doubles again* to 4000-8000 units

- This is not going to work….
  - the number of unit types grows exponentially with the number of contextual factors

Some contexts are (nearly) equivalent

- This is what makes unit selection synthesis feasible
  - *and statistical parametric synthesis too, as we will see later*

- Cannot record and store versions of every speech sound in every possible context
  - there are far too many
  - some will sound almost identical, so recording all of them is not necessary

- But we **can** have each speech sound in a **sufficient variety** of different contexts

Capture the variation observed in natural speech

- In diphone synthesis, we recorded one copy of each unit type in a carrier phrase
  - to ensure that the diphones were in a "neutral" context

- But now **we want the effects of context** (for lots of different contexts)

- The key concept of unit selection speech synthesis:
  - record a database of speech containing **natural variation** caused by context
  - at synthesis time, search for the **most appropriate sequence** of units

- *Several unit sizes (half phone, diphone, …) are possible - the principles are the same*

Orientation

- Before
  - diphone units
  - record one copy of each type
  - synthesis involves extensive signal **manipulation**

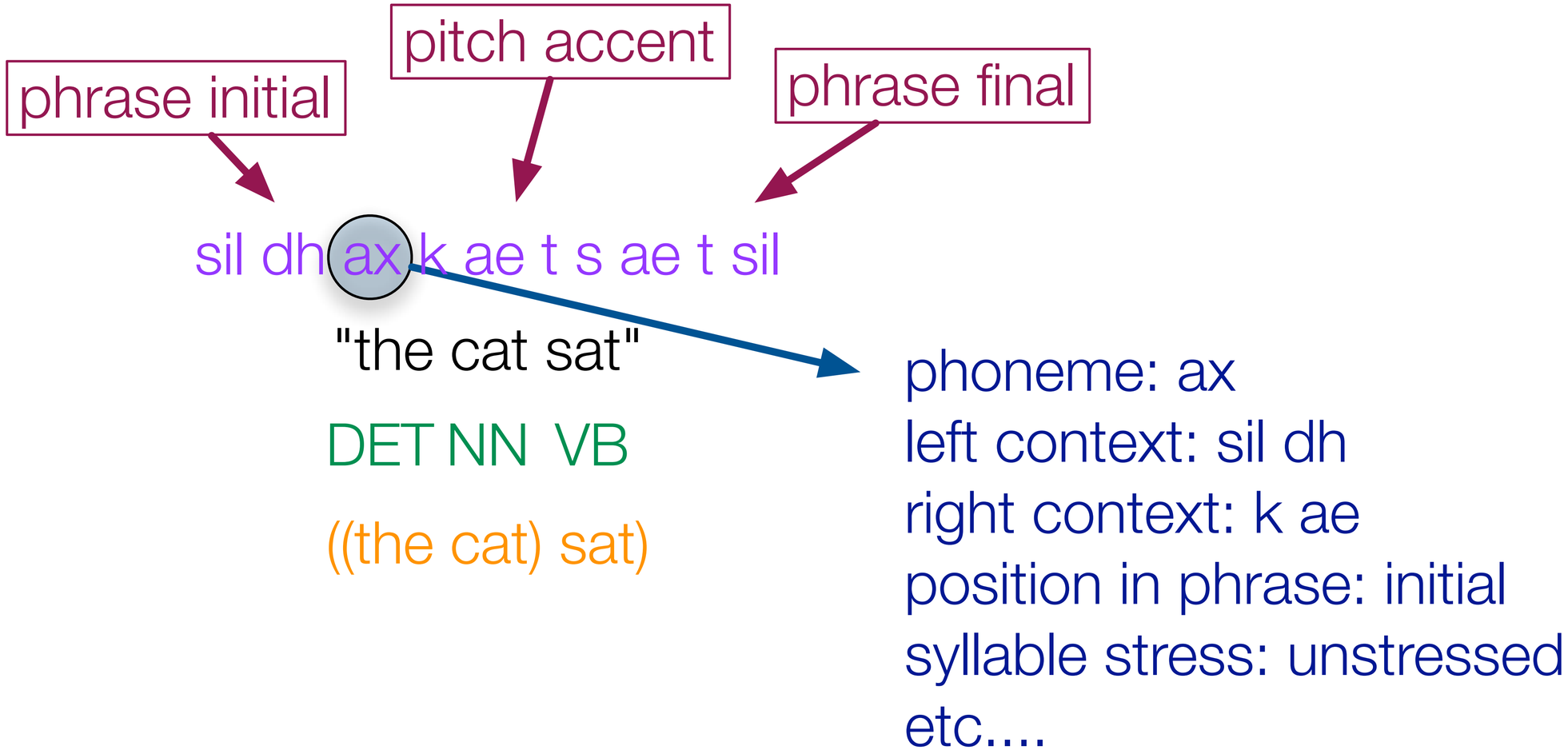
- Now
  - record naturally-varying units, occurring in complete utterances
  - synthesis involves careful **selection** of appropriate units

Unit selection

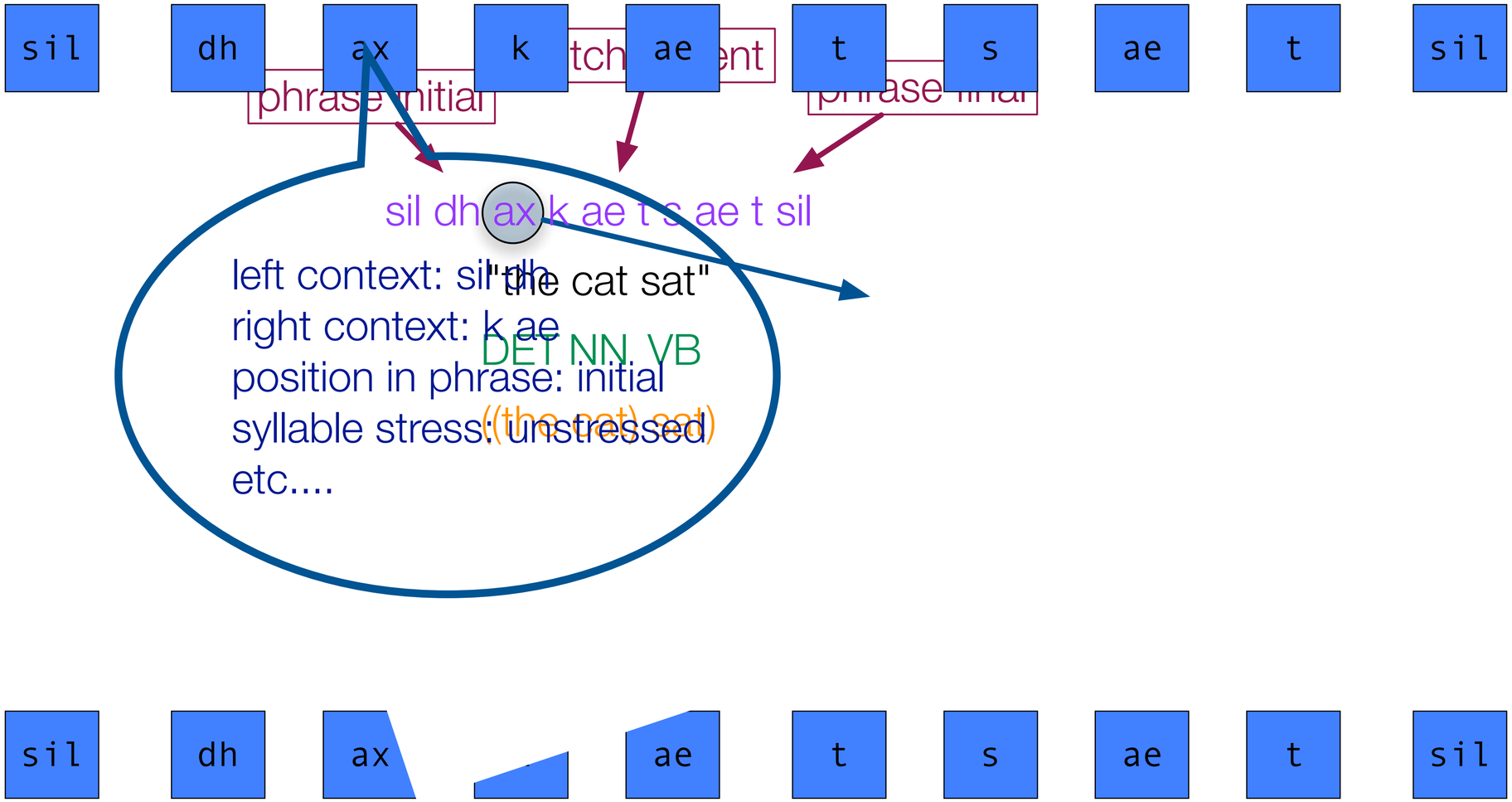
- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

From multi-level / tiered / structured linguistic information….

….to a **linear string** of context-dependent units



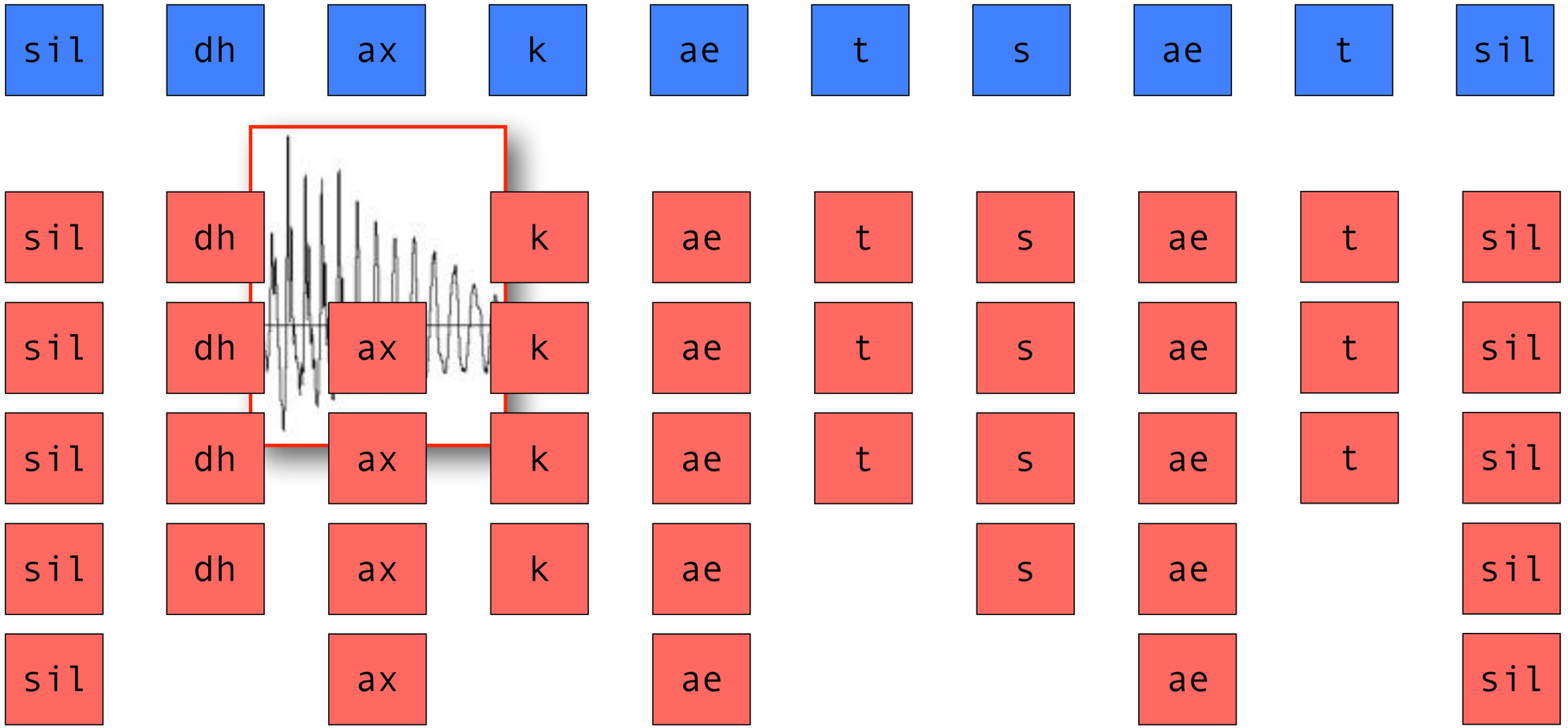
The target unit sequence



Unit selection

- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

Retrieve candidate units from the pre-recorded database



Orientation

- What have we got?
  - a sequence of **target** units
    - each annotated with linguistic features
  - for each target unit
    - several **candidates** (incl. waveforms)
    - each annotated with linguistic features

- What remains to be done?
  - find the best-sounding sequence of candidates

Orientation

- What have we got?
  - a sequence of **target** units
    - each annotated with linguistic features
  - for each target unit
    - several **candidates** (incl. waveforms)
    - each annotated with linguistic features

Importantly, the linguistic features are **local** to each target and each candidate unit

- What remains to be done?
  - find the best-sounding sequence of candidates

1. Quantify "best sounding"
2. Search for the best sequence

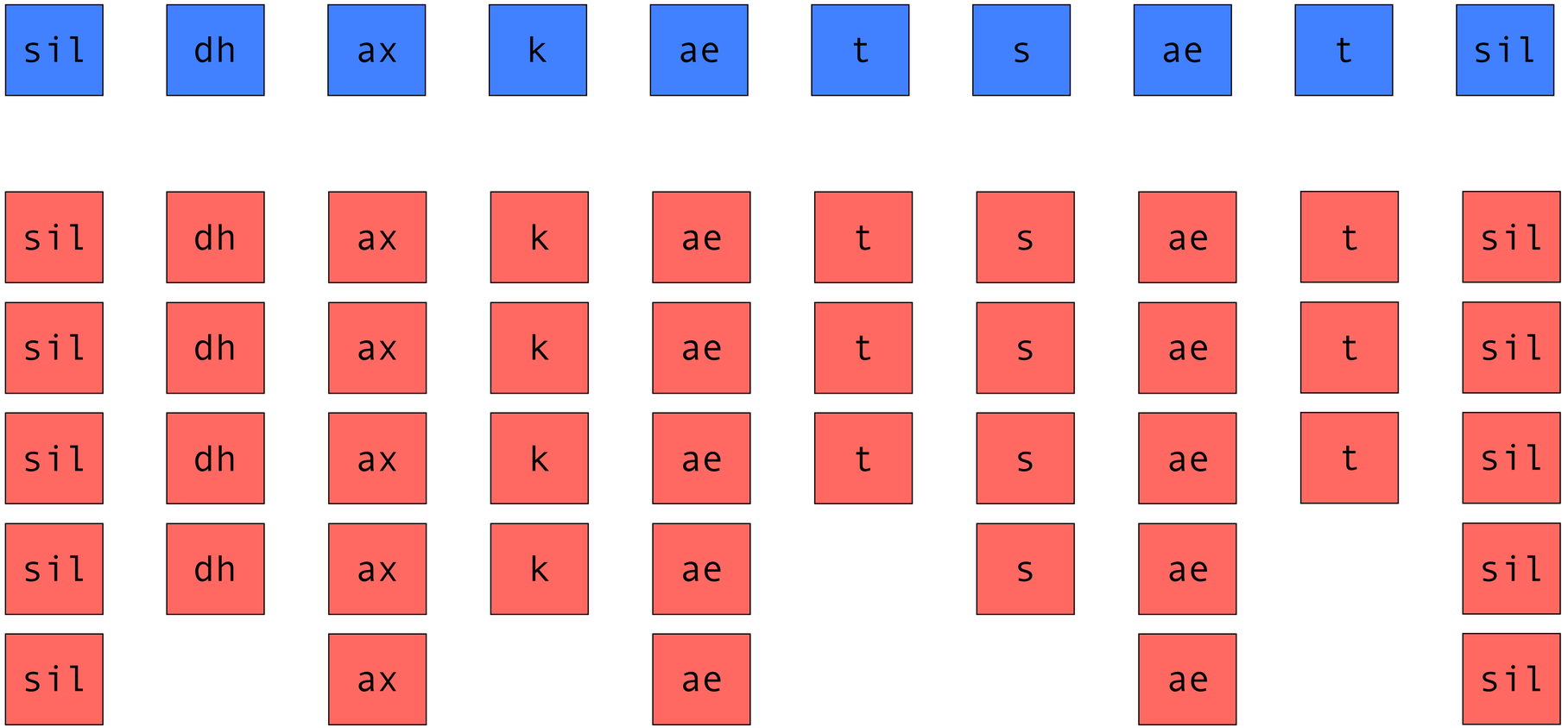
Unit selection

- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

} Quantify "best sounding"

} Search for the best sequence

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*

How to measure this similarity

- Taylor describes **two possible formulations** of the target cost function

  - independent feature formulation (**IFF**)
    - assume that units from similar linguistic contexts will sound similar
    - target cost function measures linguistic feature mismatch

  - acoustic-space formulation (**ASF**)
    - acoustic properties of the candidates are known
    - make a prediction of the acoustic properties of the target units
    - target cost function measures acoustic distance between candidates and targets

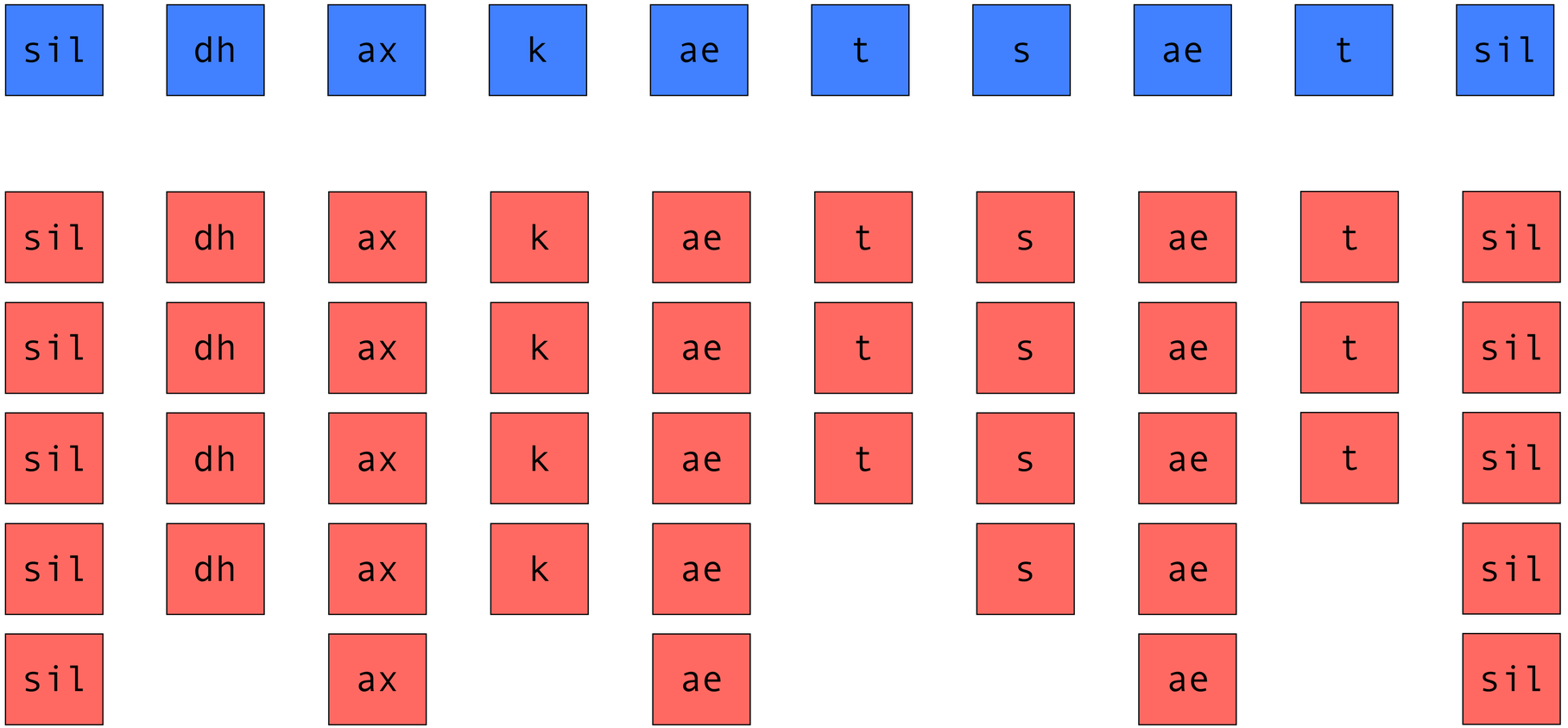
Unit selection

- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function
- search

} Quantify "best sounding"

} Search for the best sequence

Which candidate sequence will sound best?



Acoustic criteria

- After candidate units are taken from the database, they will be joined (**concatenated**)
- Cannot simply join two fragments of speech and hope that it will sound OK
  - generally, it will not !

- Why? …because of **mismatches** in **acoustic properties** around the join point, including
  - the spectral envelope, F0, energy

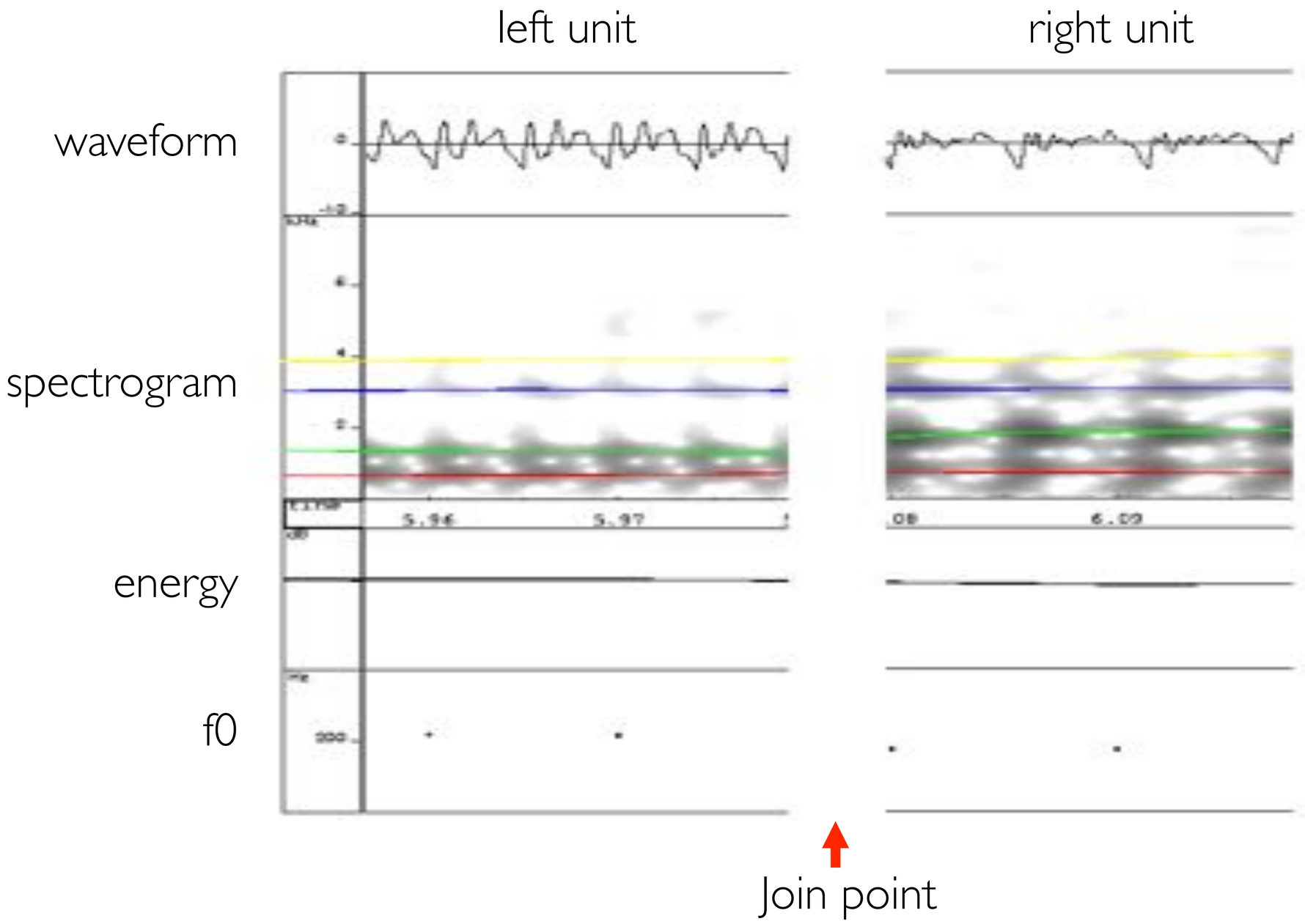
- The **acoustic mismatch** between consecutive candidates is measured by the *join cost function*

Join cost

- The join cost measures the **acoustic mismatch** between two candidate units
  - we are assuming this reflects **perceptual** mismatch
- 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)

- Common to also inject some **knowledge** into the join cost
  - some phones are easier to splice than others
  - so, can bias against joins in difficult places (e.g., vowels before an [r])


left unit
right unit
waveform
spectrogram
energy
f0
5.96
5.97
6.09
Join point

What about a join cost computed across multiple frames?

- Typical join cost function (e.g., Festival's *multisyn*) uses **one frame** from each side of the join
  - very **local**, may miss important information

- Ways to improve the join cost
  - consider **several frames** around the join, or the entire sequence of frames
  - consider **deltas**

- A natural extension of this would be a (probabilistic) model of the **sequence of frames**
  - **hybrid** synthesis, which typically involves predicting acoustic parameter trajectories, addresses this

Unit selection

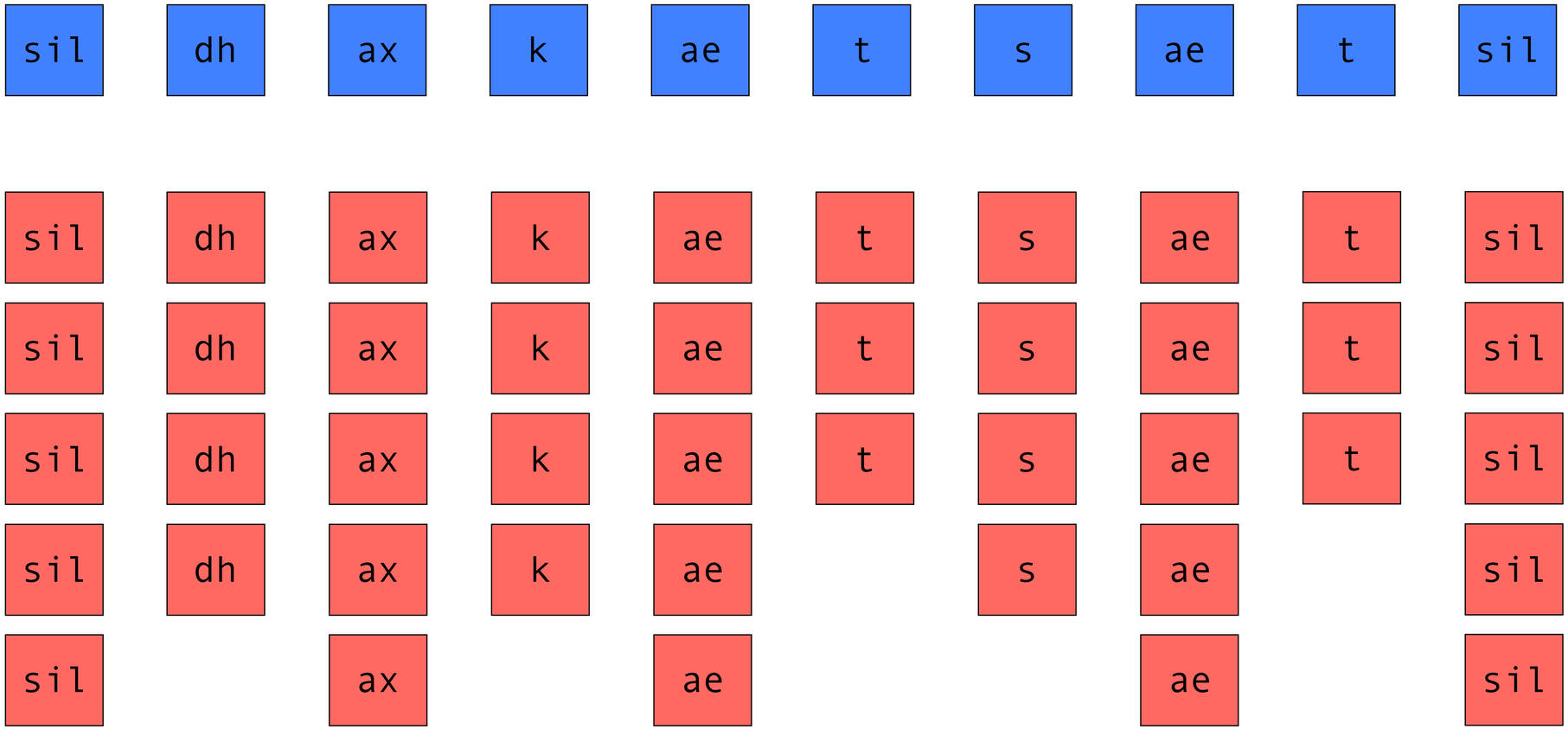
- key concepts
- target unit sequence
- candidate units from the database
- measuring similarity using the target cost function
- measuring concatenation quality using the join cost function

} Quantify "best sounding"

- search

} Search for the best sequence

Which candidate sequence will sound best? The one with **lowest cost** !



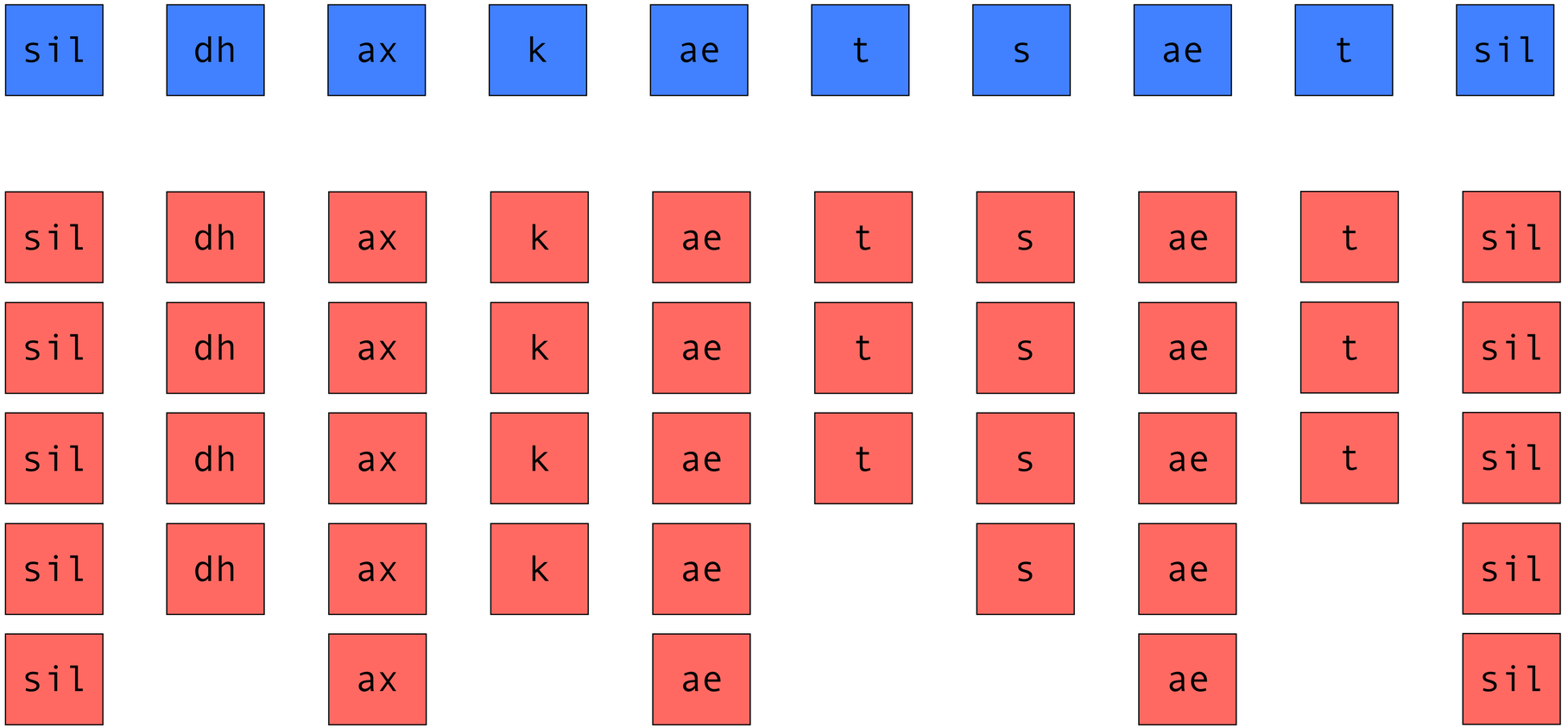
Why must a search be performed?

- The total cost of a particular candidate unit sequence under consideration is the sum of
  - the target cost for every **individual candidate unit** in the sequence
  - the join cost between every pair of **consecutive candidate units** in the sequence

- Because of the join cost, the choice of which candidate to use in one position **depends** on which units are chosen for the neighbouring positions
  - it is not possible to independently choose the best candidate for each target

- There **is** a single globally-optimal (lowest total cost) sequence
  - a **search** is required, to find this sequence

Efficient search using dynamic programming



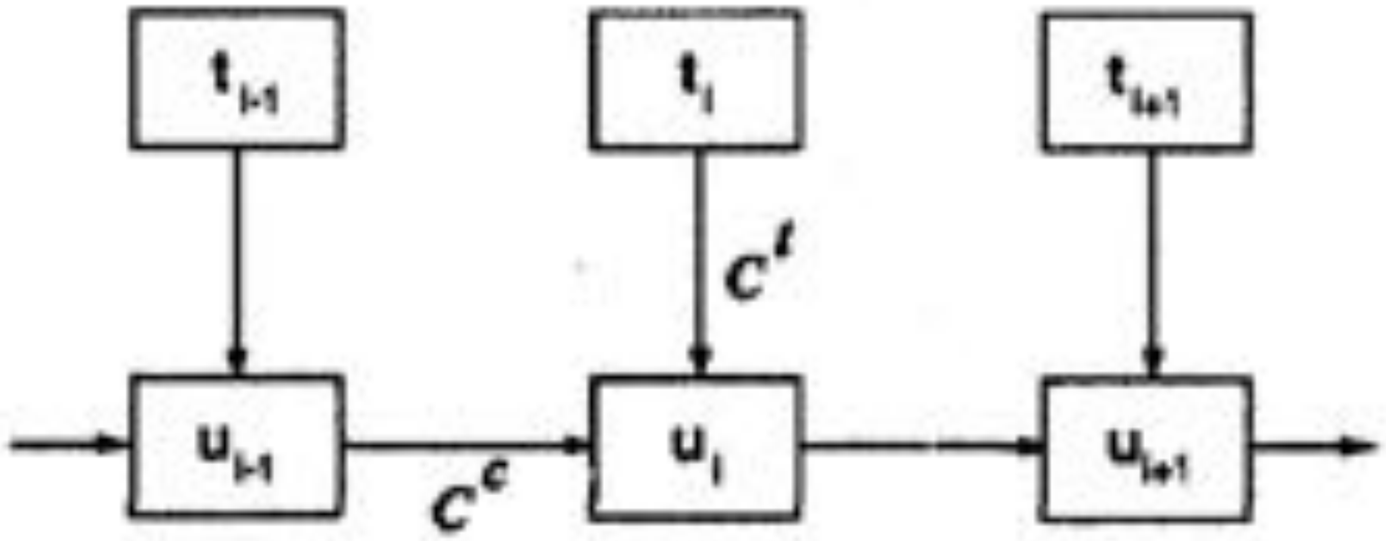


Figure 1. Unit Selection Costs

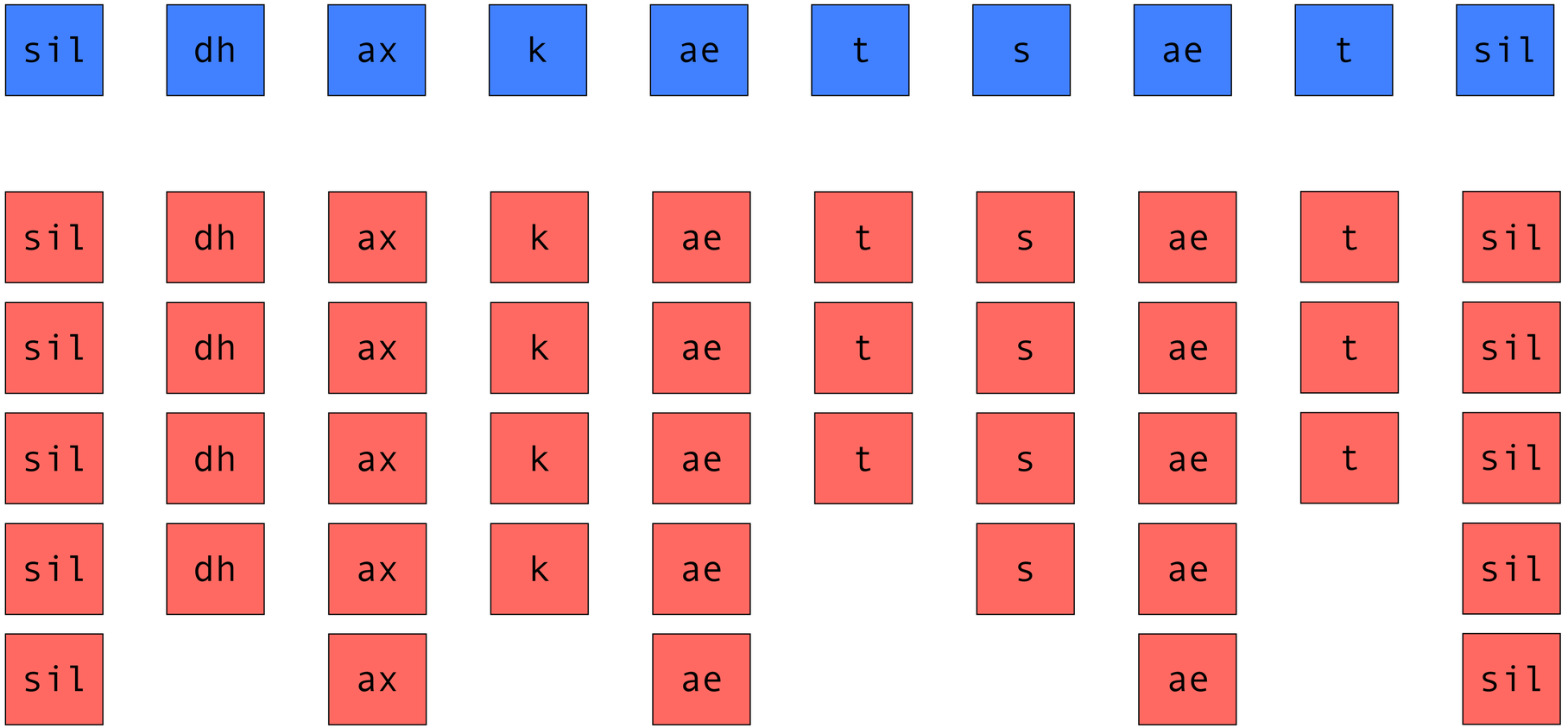
Figure from Hunt, A. J., and Black, A. W. Unit selection in a concatenative speech synthesis system using a large speech database. In Proc. ICASSP '96, Atlanta, Georgia, USA (1996), pp. 373–376.

What base unit type is really used? Homogeneous or heterogeneous units?

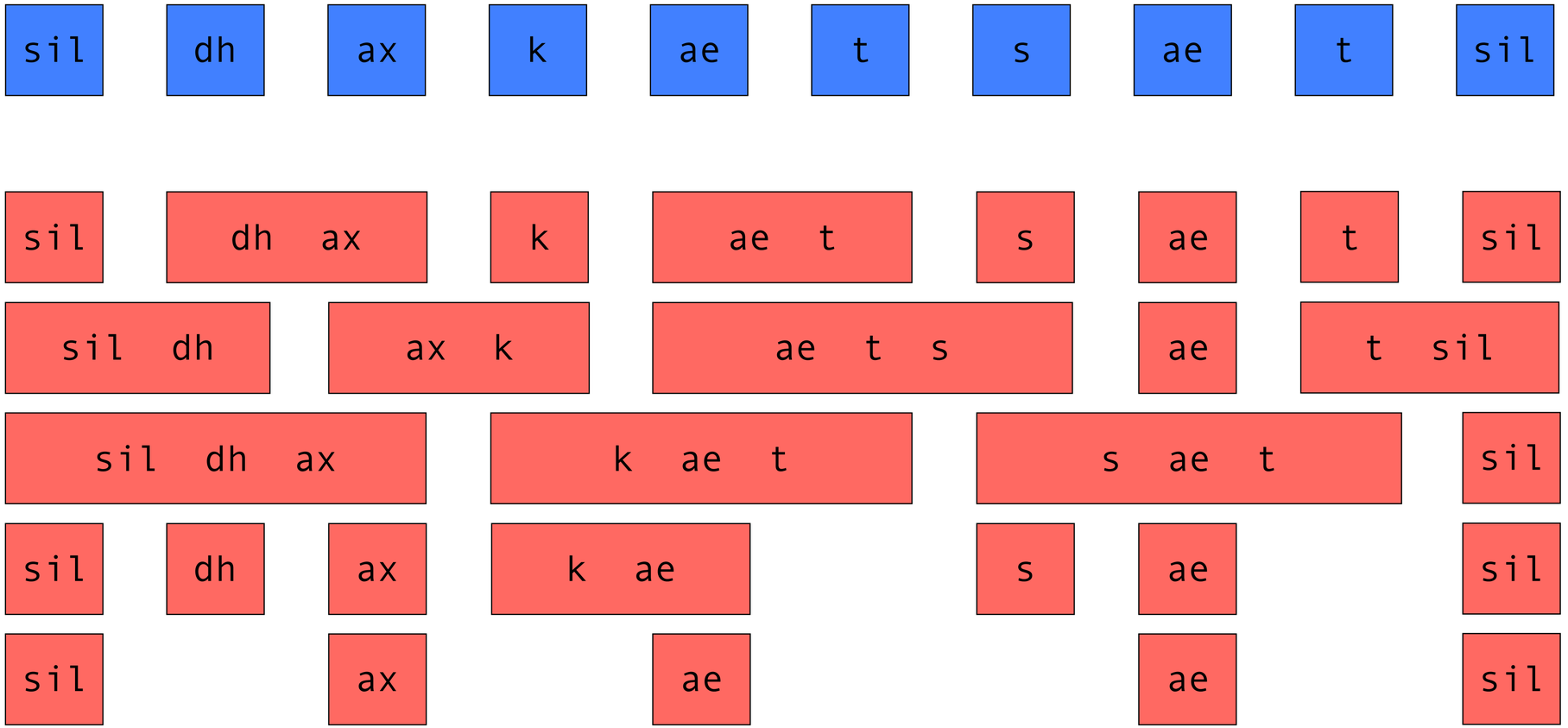
- Homogeneous system will be easier to implement in software
  - the start and end points of all candidate units align
  - the search lattice is simple

- Heterogeneous system will be a little more complex
  - start and end points will not all align
  - number of target costs and join costs to sum up will **vary** for different paths through the lattice
  - some normalisation may be needed to correct for this

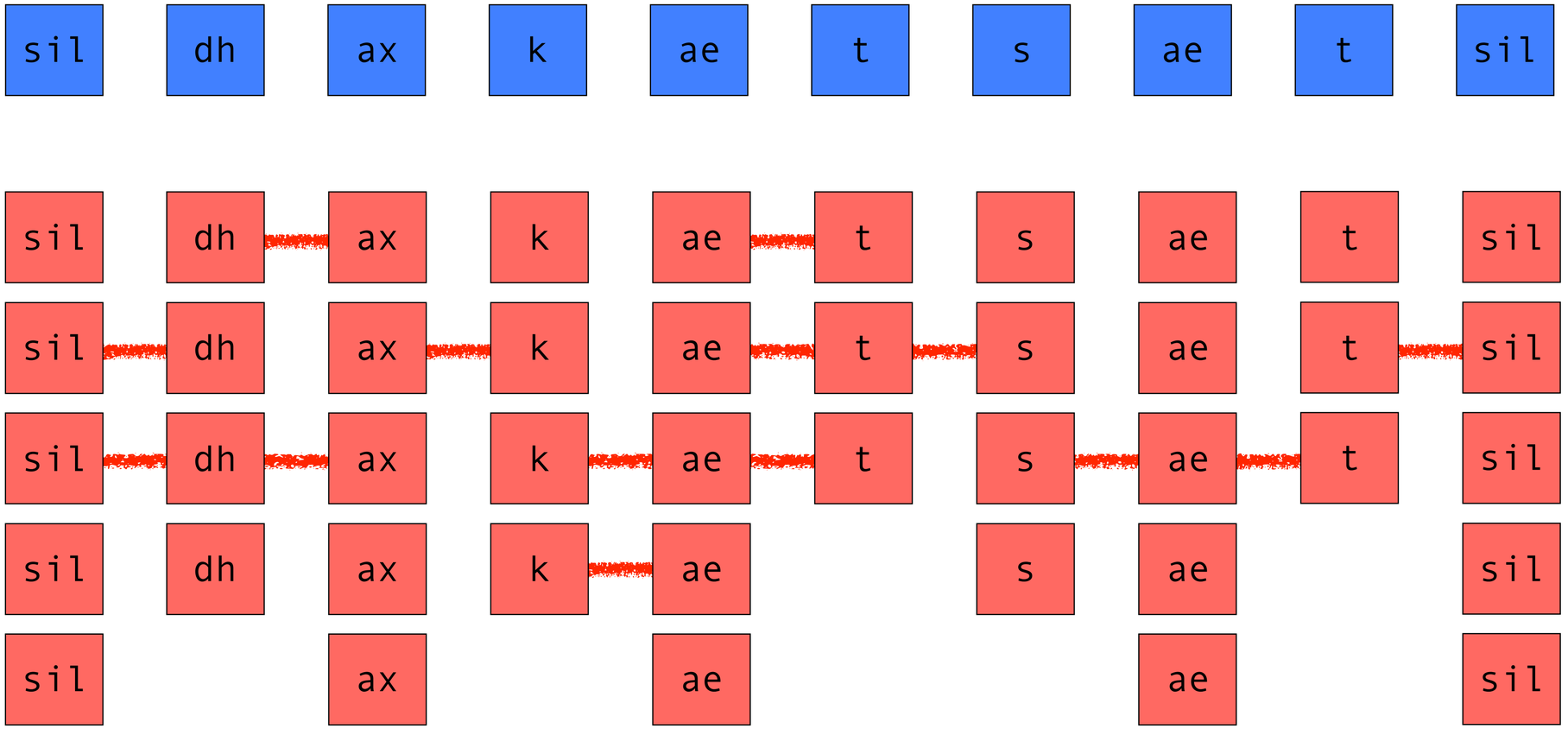
Homogeneous unit type (whole phones in this toy example)



Heterogeneous unit type (multi-phone units in this toy example)



Homogeneous unit type with the "zero join cost trick"
= heterogeneous units !



What next?

- A closer look at the target cost
  - Independent Feature Formulation
  - Acoustic Space Formulation

- Then, how to create the database of recorded, annotated speech

What next?

- A closer look at the target cost
  - Independent Feature Formulation
  - Acoustic Space Formulation

This will eventually lead us to **hybrid** methods which use statistical models to make **predictions** about the acoustic properties of the target unit sequence

- Then, how to create the database of recorded, annotated speech