Module 4

Pronunciation & prosody

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

Coffee costs £2.

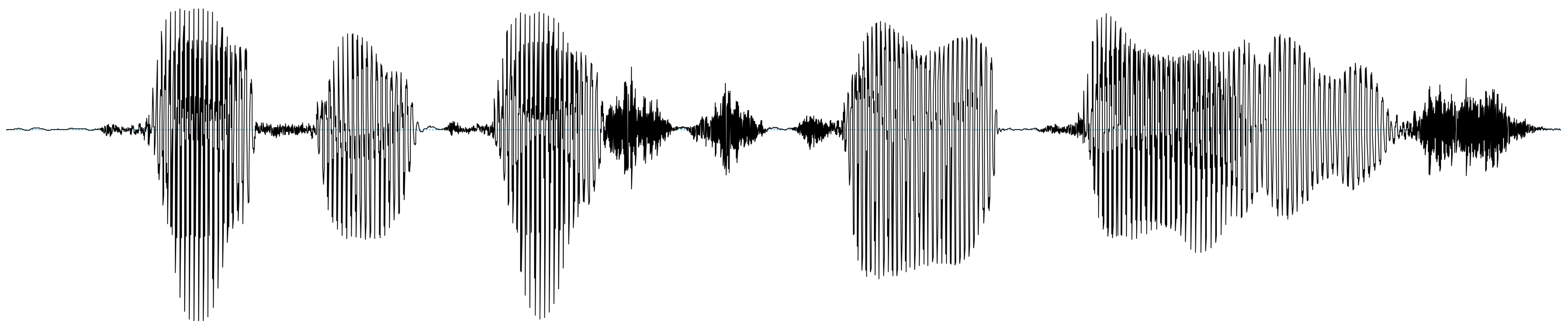
- Normalise text

coffee costs two pounds .

- Predict pronunciation & prosody

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

- Generate waveform



What you should already know

- <u>From the videos & readings</u>
  - Letter to sound (LTS)
  - A worked example of LTS using a classification tree
  - Prosody prediction

- morphology
- POS
- dictionary lookup of word + POS
- syllables & lexical stress
- LTS (rules or model)
- post-lexical rules

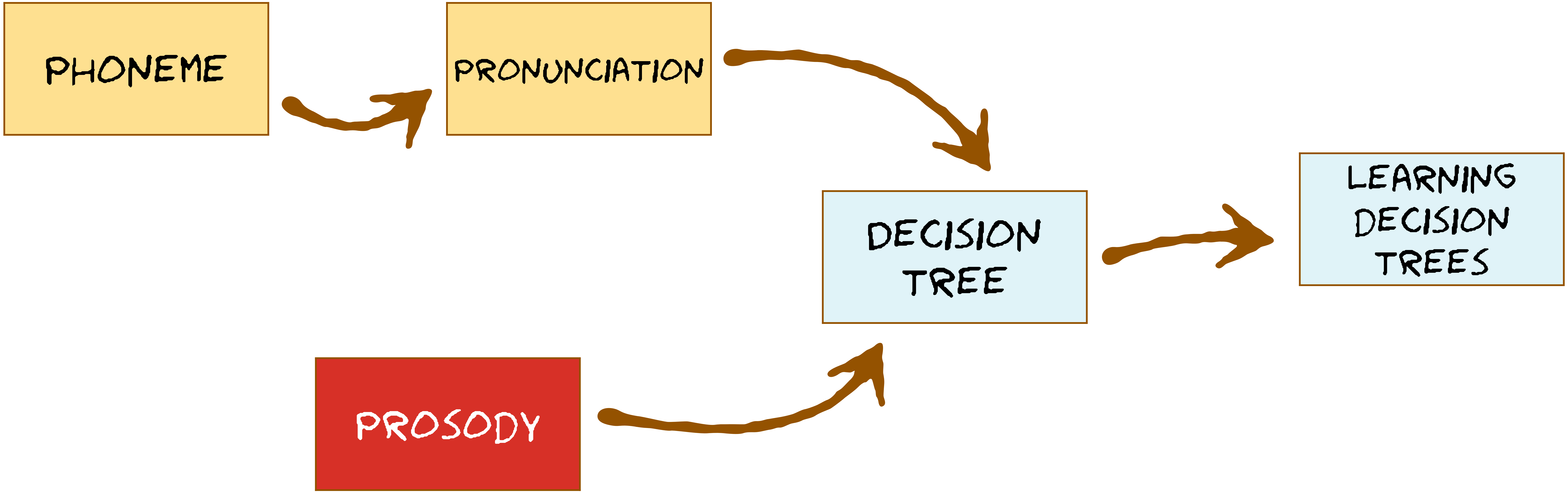
- gathering and preparing training data
- choosing the predictors
- growing the tree (learning from data)

- placement of events (classification)
- deciding their types (classification)
- realisation (regression)

Today's topics - Module 4: pronunciation & prosody

	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 4: pronunciation & prosody



Speech synthesis - pronunciation & prosody

- Machine Learning
- Classification And Regression Trees (CARTs)
  - classification: understanding entropy as a measure of predictability
  - regression: measuring the predictability of a continuous variable
  - stopping criteria

Step 1 - define the overall task we are going to solve

from the orthographic form: **HOGWASH**

predict the pronunciation: **HH AA G W AA SH**

PHONEME

PRONUNCIATION

Step 2 - break the task down into simple, solvable, sub-tasks

from *one letter*
of the orthographic form: **HOGWASH**

predict *zero, one or two phones*
of the pronunciation: **HH AA G W AA SH**

Step 3 - obtain the raw training data

- for Letter-to-Sound (LTS), this is **simply a pre-existing dictionary**

here are some words from the CMU dictionary that use the letter "**A**"

```
HOGWASH HH AA G W AA SH
CARWASH K AA R W AA SH
WARRANT W AO R AH N T
WARRANTY W AO R AH N T IY
HARDWARE HH AA R D W EH R
SOFTWARE S AO F T W EH R
WARES W EH R Z
```

Step 4 - define the *predictee*
- which phone are we going to predict from this letter?

H O G W **A** S H

HH AA G W AA SH

Step 5 - choose the *predictors*

H O G W **A** S H

HH AA G W **AA** SH

Step 6 - get the training data ready for machine learning

predictors							predictee
ppp	**pp**	**p**		**n**	**nn**	**nnn**	
o	g	w	**a**	s	h	-	aa
a	r	w	**a**	s	h	-	aa
-	-	w	**a**	r	r	a	ao
-	-	w	**a**	r	r	a	ao
r	d	w	**a**	r	e	-	eh
f	t	w	**a**	r	e	-	eh
-	-	w	**a**	r	e	s	eh

Speech synthesis - pronunciation & prosody

- Machine Learning
- Classification And Regression Trees (CARTs)
  - classification: understanding entropy as a measure of predictability
  - regression: measuring the predictability of a continuous variable
  - stopping criteria

In-class exercise

Building a decision tree for phrase-break prediction

Step 1 - define the overall task we are going to solve

For a sentence, predict where the phrase breaks should go.

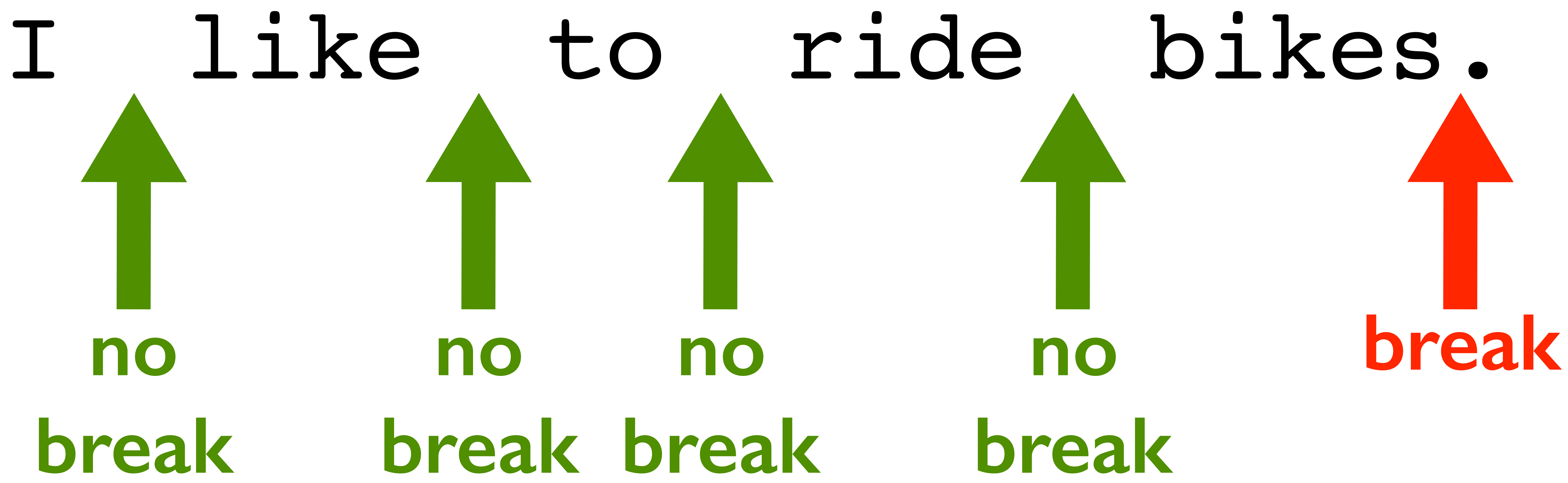
```
I  like  to  ride  bikes.
```

break

PROSODY

Step 2 - break the task down into simple, solvable, sub-tasks

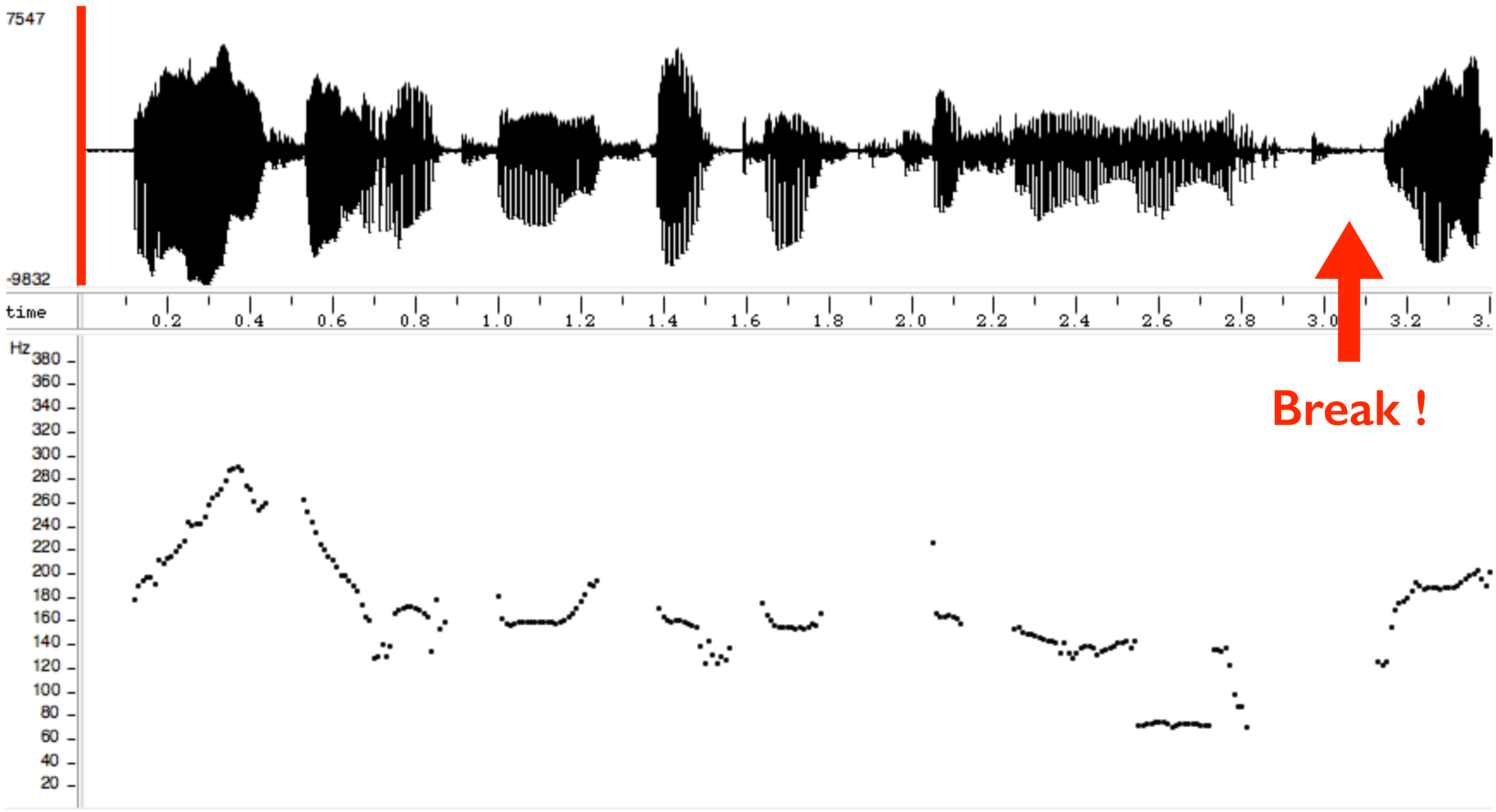
For **each word**, predict whether there is phrase break after it.



Step 3 - obtain the raw training data

this is going to be ***expensive*** *because it will involve manually labelling spoken utterances*

Words	I	like	to	ride	bikes	.
Breaks					**BREAK**	
Words	Food	and	drink	are	nice	.
Breaks	**BREAK**				**BREAK**	
Words	Apples	but	not	pears	.	
Breaks	**BREAK**			**BREAK**		
Words	He	is	but	she’s	not	!
Breaks		**BREAK**			**BREAK**	
Words	One	,	two	,	three	.
Breaks	**BREAK**		**BREAK**		**BREAK**	
Words	Shaken	yet	not	stirred	.	
Breaks	**BREAK**			**BREAK**		


7547
-9832
time
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
Hz
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
Break !

Step 4 - define the *predictee*

and the possible values it can take: BREAK *—or—* No break

Words	I	like	to	ride	bikes	.
Breaks	No break	No break	No break	No break	**BREAK**	No break
Words	Food	and	drink	are	nice	.
Breaks	**BREAK**	No break	No break	No break	**BREAK**	No break
Words	Apples	but	not	pears	.	
Breaks	**BREAK**	No break	No break	**BREAK**	No break	
Words	He	is	but	she's	not	!
Breaks	No break	**BREAK**	No break	No break	**BREAK**	No break
Words	One	,	two	,	three	.
Breaks	**BREAK**	No break	BREAK	No break	**BREAK**	No break
Words	Shaken	yet	not	stirred	.	
Breaks	**BREAK**	No break	No break	**BREAK**	No break	

Step 5 - choose the *predictors*

they can only be things that you will also know for the test data

Words	I	like	to	ride	bikes	.
POS	N	V	TO	V	N	PUNC
Breaks	No break	No break	No break	No break	**BREAK**	No break

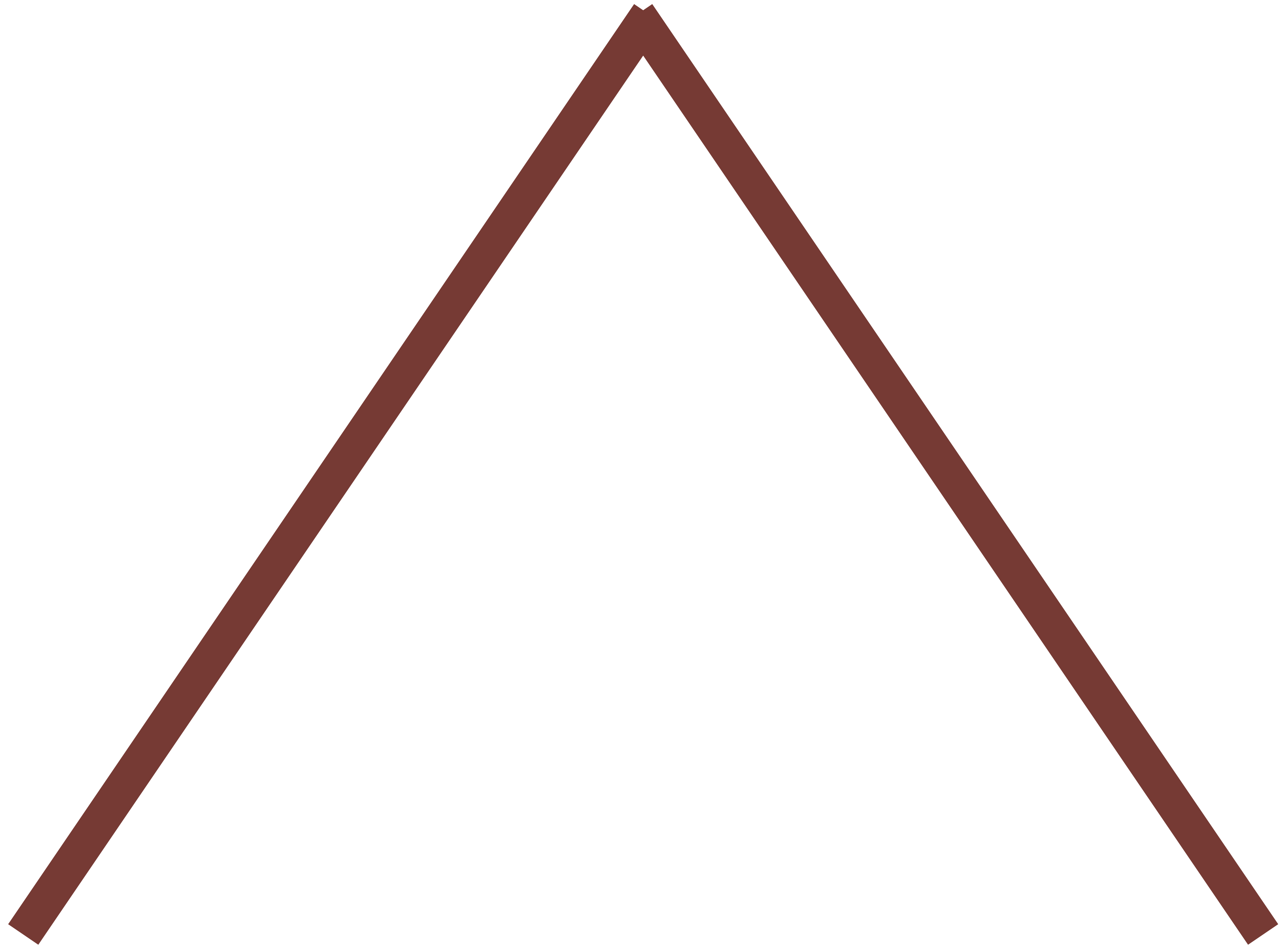
Step 5 - choose the *predictors*

Words	I	like	to	ride	bikes	.
POS	N	V	TO	V	N	PUNC
Breaks	No break	No break	No break	No break	**BREAK**	No break
Words	Food	and	drink	are	nice	.
POS	N	CC	N	V	JJ	PUNC
Breaks	**BREAK**	No break	No break	No break	**BREAK**	No break
Words	Apples	but	not	pears	.	
POS	N	CC	RB	N	PUNC	
Breaks	**BREAK**	No break	No break	**BREAK**	No break	
Words	He	is	but	she's	not	!
POS	N	V	CC	N	V	PUNC
Breaks	No break	**BREAK**	No break	No break	**BREAK**	No break
Words	One	,	two	,	three	.
POS	CD	PUNC	CD	PUNC	CD	PUNC
Breaks	**BREAK**	No break	BREAK	No break	**BREAK**	No break
Words	Shaken	yet	not	stirred	.	
POS	JJ	CC	RB	JJ	PUNC	
Breaks	**BREAK**	No break	No break	**BREAK**	No break	

Step 6 - get the training data ready for machine learning

Words	I	like	to	ride	bikes	.
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	NO-BREAK	NO-BREAK	NO-BREAK	NO-BREAK	BREAK	NO-BREAK
Words	Food	and	drink	are	nice	.
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	BREAK	NO-BREAK	NO-BREAK	NO-BREAK	BREAK	NO-BREAK
Words	Apples	but	not	pears	.	
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	BREAK	NO-BREAK	NO-BREAK	BREAK	NO-BREAK	
Words	He	is	but	she’s	not	!
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	NO-BREAK	BREAK	NO-BREAK	NO-BREAK	BREAK	NO-BREAK
Words	One	,	two	,	three	.
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	BREAK	NO-BREAK	BREAK	NO-BREAK	BREAK	NO-BREAK
Words	Shaken	yet	not	stirred	.	
Predictor 1 : L POS						
Predictor 2 : C POS						
Predictor 3 : R POS						
Predictee	BREAK	NO-BREAK	NO-BREAK	BREAK	NO-BREAK	

DECISION TREE



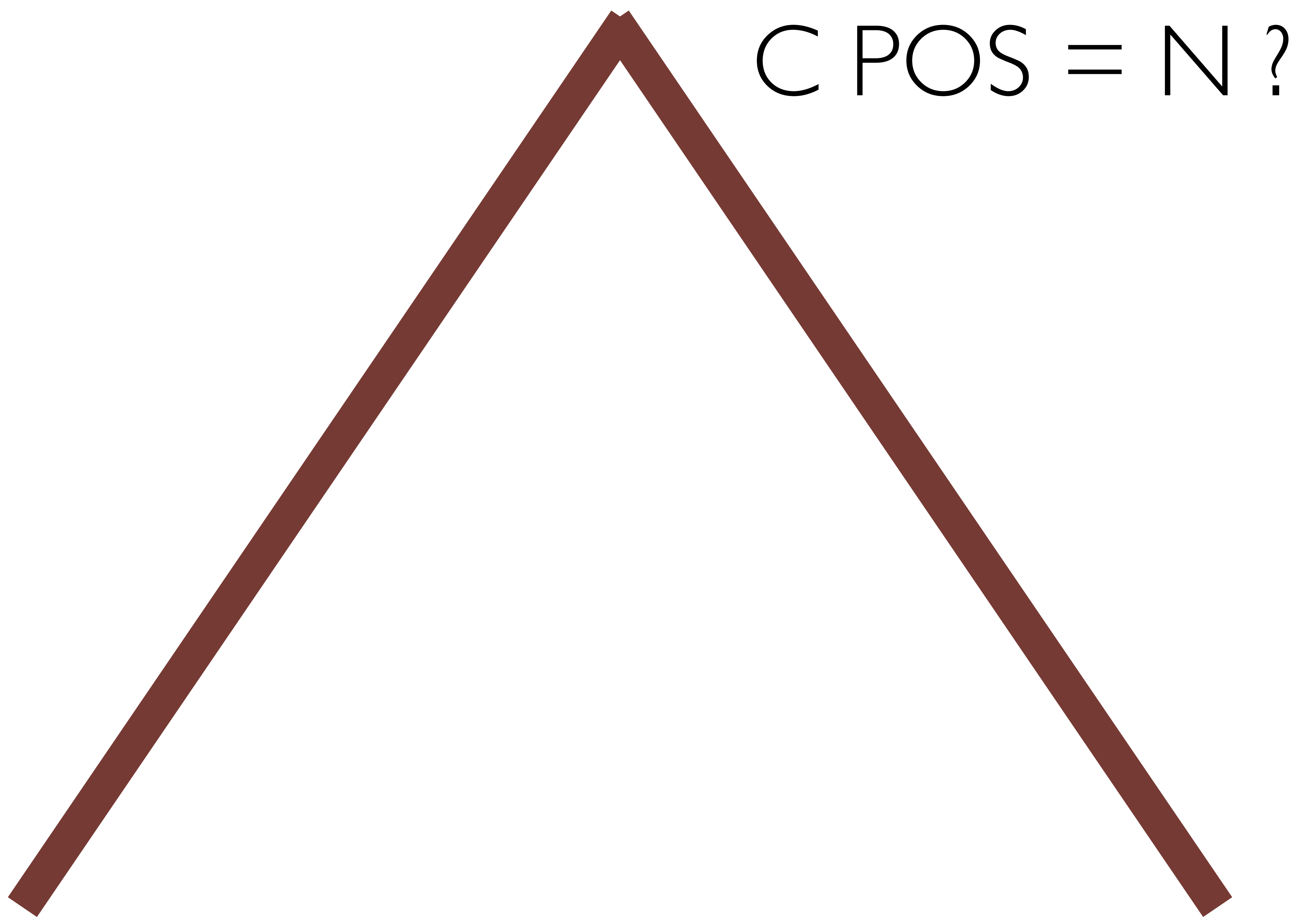
Make a list of all possible questions

L POS =	
L POS =	
L POS =	
L POS =	
L POS =	
L POS =	
L POS =	
L POS =	

C POS =	
C POS =	
C POS =	
C POS =	
C POS =	
C POS =	
C POS =	
C POS =	

R POS =	
R POS =	
R POS =	
R POS =	
R POS =	
R POS =	
R POS =	
R POS =	

Try question “C POS = N ?”



Measure goodness of split for the question “C POS = N ?”

Entropy at the Y node	4	BREAK	0.50		-0.50	
	4	NO BREAK	0.50		-0.50	
# data points	8				**1.00**	bits

Entropy at the N node	8	BREAK	0.31		-0.52	
	18	NO BREAK	0.69		-0.37	
# data points	26				**0.89**	bits

# data points in total	34					
Total entropy					**0.92**	bits

Try all possible questions, measuring goodness of split (as entropy, in bits)

L POS =	PUNC
L POS =	N
L POS =	V
L POS =	TO
L POS =	CC
L POS =	JJ
L POS =	RB
L POS =	CD

C POS =	PUNC	
C POS =	**N**	**0.92**
C POS =	V	
C POS =	TO	
C POS =	CC	
C POS =	JJ	
C POS =	RB	
C POS =	CD	

R POS =	**PUNC**	**0.47**
R POS =	N	
R POS =	V	
R POS =	TO	
R POS =	**CC**	**0.74**
R POS =	JJ	
R POS =	RB	
R POS =	CD	

Speech synthesis - pronunciation & prosody

- Machine Learning
- Classification And Regression Trees (CARTs)
  - classification: understanding entropy as a measure of predictability
  - <u>regression: measuring the predictability of a continuous variable</u>
  - stopping criteria

$\sigma^2 = 18.7$

13.4
3.14
2.73
11.3
1.23
4.52
9.42
2.11
10.1
1.87

3.14
2.73
1.23
4.52
2.11
1.87

$\sigma^2 = 1.11$

13.4
11.3
9.42
10.1

$\sigma^2 = 2.29$

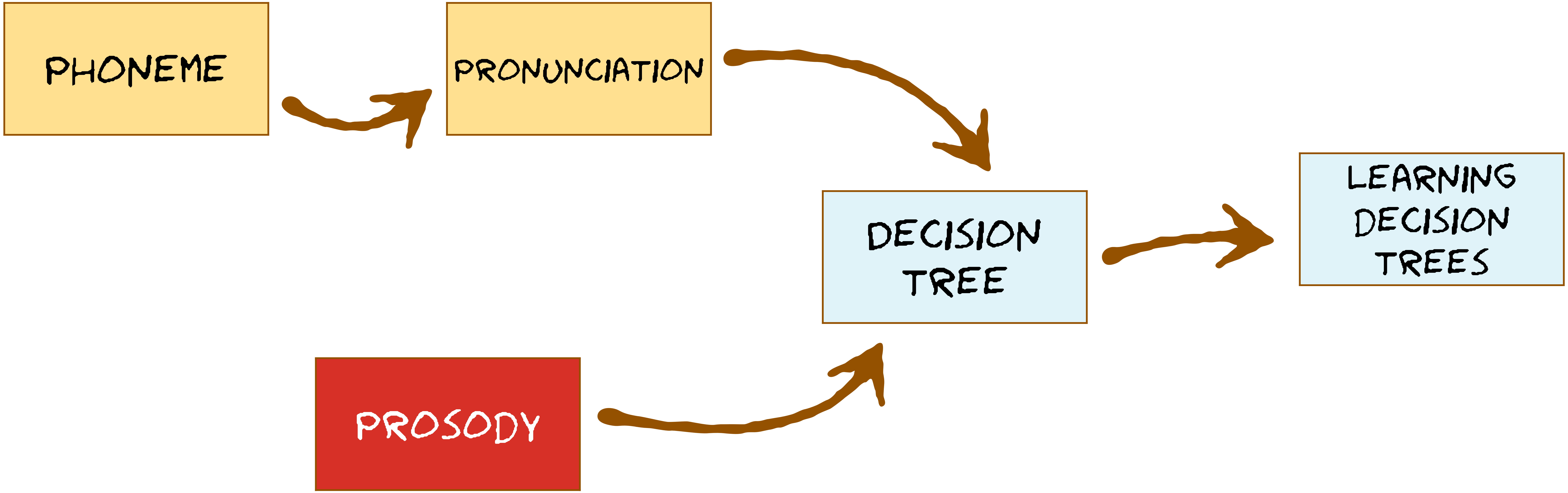
Speech synthesis - pronunciation & prosody

- Machine Learning
- Classification And Regression Trees (CARTs)
  - classification: understanding entropy as a measure of predictability
  - regression: measuring the predictability of a continuous variable
  - stopping criteria

Stopping criteria (we may use several)

- Classification or Regression
  - all data points have the **same value for the predictee** (job done!)
  - all data points have the **same values for all predictors**
    - equivalently: no available question can split them
  - **number of data points in parent node** is below a threshold
  - **number of data points in a child node** would fall below a threshold
- Classification only
  - cannot reduce **entropy** by more than some pre-specified amount
- Regression only
  - cannot reduce **variance** by more than some pre-specified amount

Today's topics - what we covered

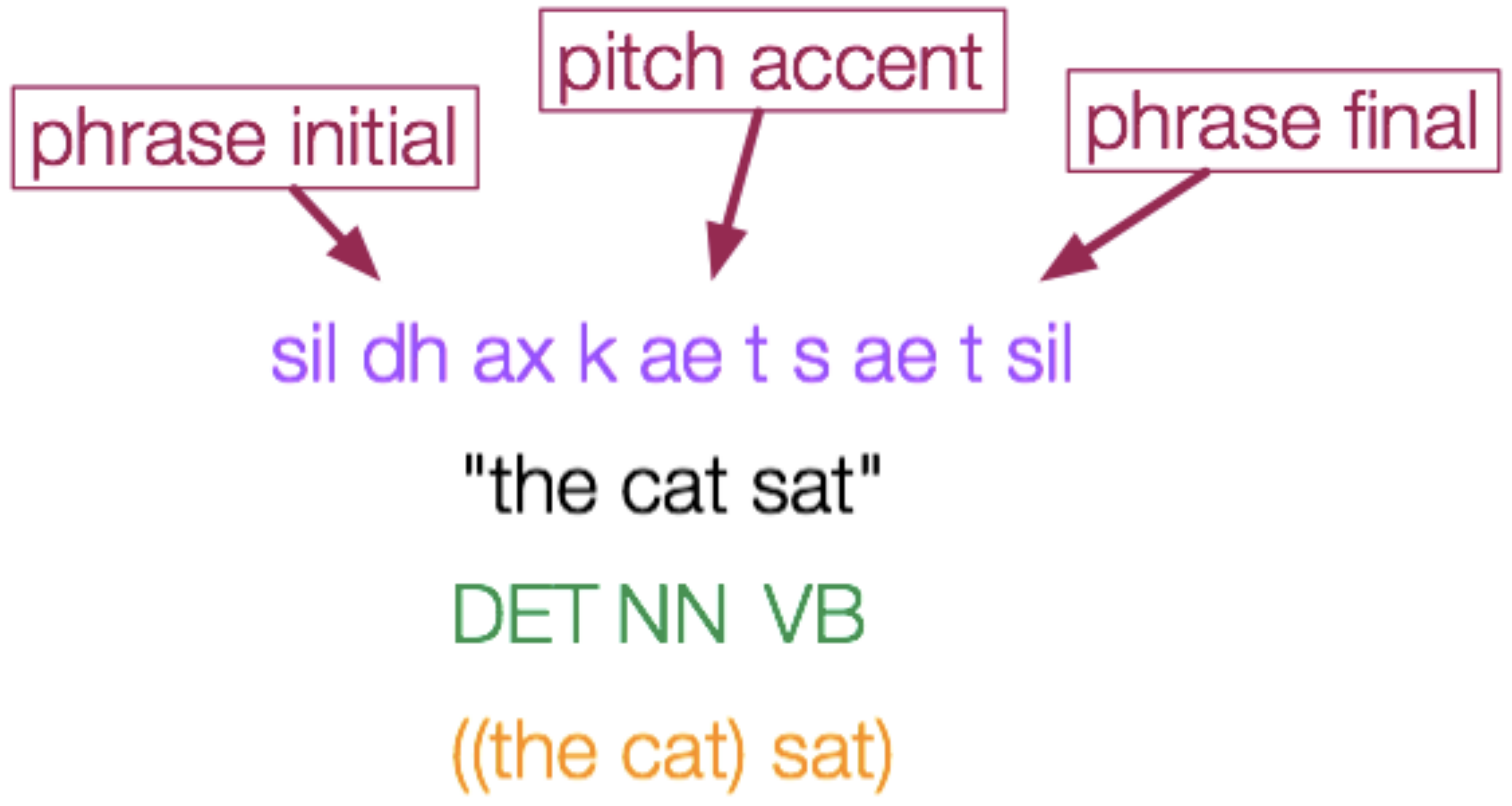


What next?

- We have
  - **normalised** the text
  - predicted **pronunciation**
  - predicted **prosody**

- That completes the **linguistic specification**

- Next, from that linguistic specification
  - it's time to generate a **waveform**



In Module 5