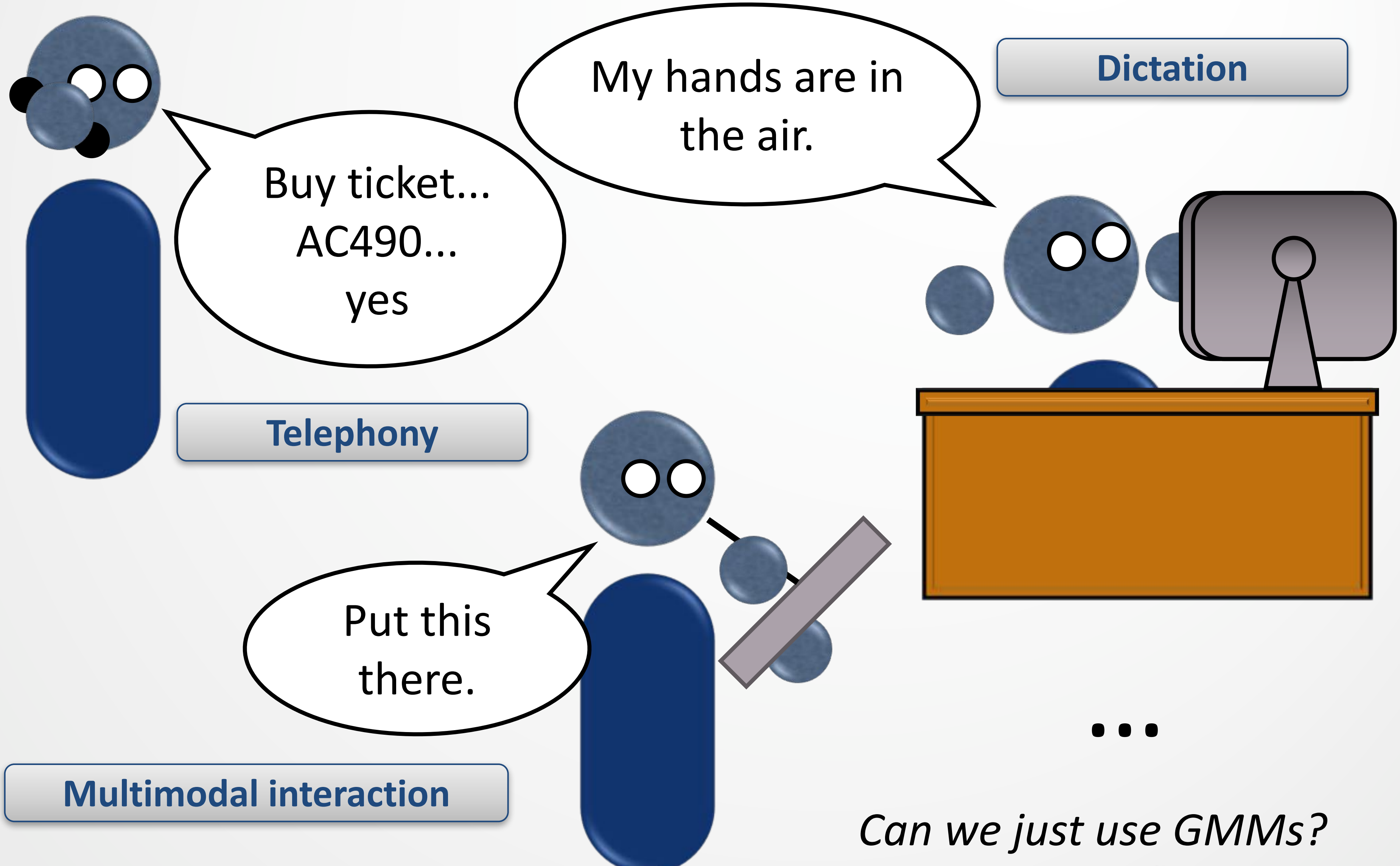
automatic speech recognition 2

CSC401/2511 – Natural Language Computing – Spring 2017
Lecture 8-2 Frank Rudzicz
University of Toronto

This lecture

- Automatic speech recognition (ASR)
  - Applying HMMs to ASR,
  - Practical aspects of ASR, and
  - Levenshtein distance.

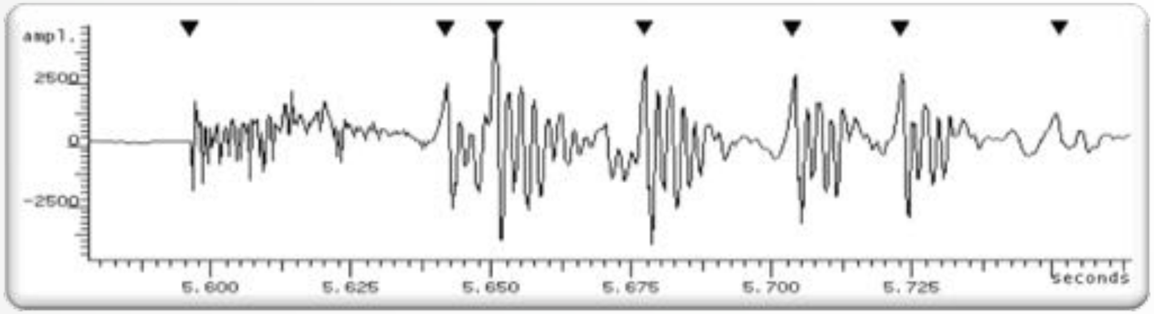
Consider what we want speech to do



Can we just use GMMs?

UNIVERSITY OF TORONTO

Speech is dynamic



- Speech **changes** over time.
  - GMMs are good for high-level clustering, but they encode **no notion** of *order*, *sequence*, or *time*.
- Speech is an expression of **language**.
  - We want to incorporate knowledge of how phonemes and words are ordered with **language models**.

Speech is sequences of phonemes (*)

We want to convert a series of MFCC vectors into a sequence of phonemes.

(*) not really

UNIVERSITY OF TORONTO

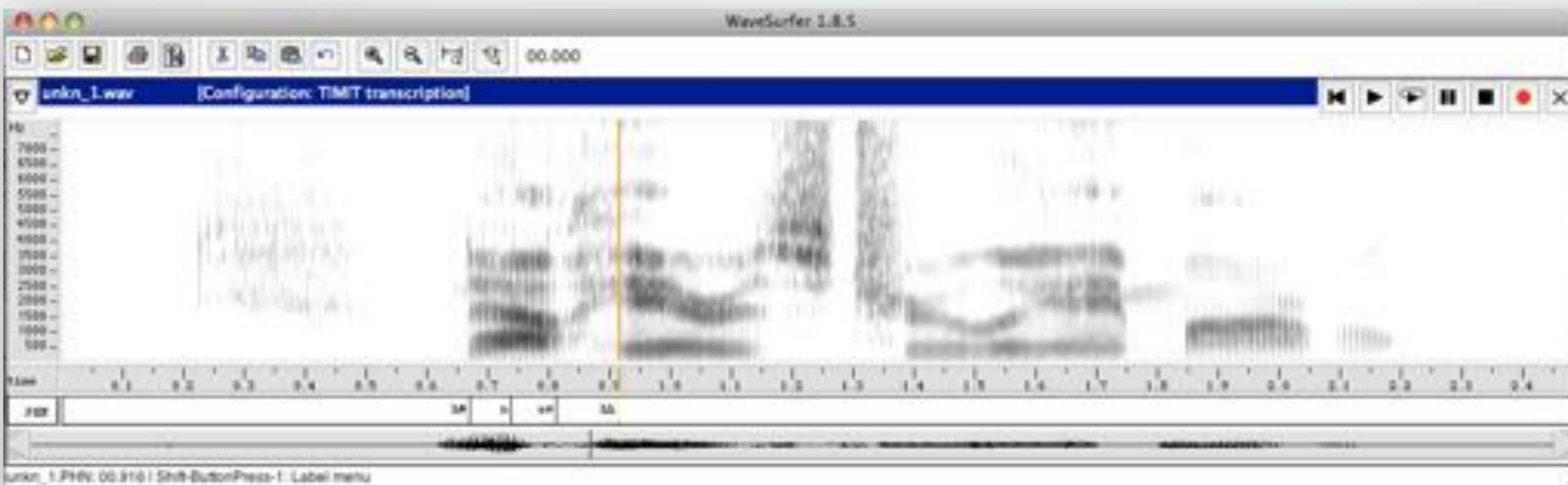
Phoneme dictionaries

- There are many **phonemic dictionaries** that map words to pronunciations (i.e., lists of phoneme sequences).
- The **CMU dictionary** (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) is popular.
  - 127K words transcribed with the ARPAbet.
  - Includes some rudimentary **prosody markers**.

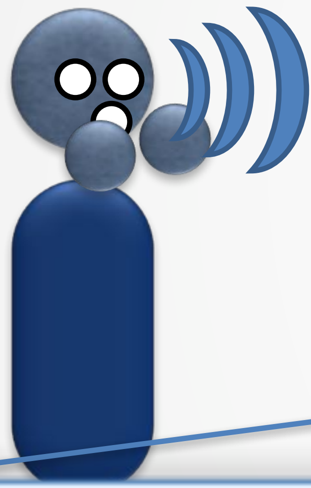
```
…
EVOLUTION          EH2 V AH0 L UW1 SH AH0 N
EVOLUTION(2)       IY2 V AH0 L UW1 SH AH0 N
EVOLUTION(3)       EH2 V OW0 L UW1 SH AH0 N
EVOLUTION(4)       IY2 V OW0 L UW1 SH AH0 N
EVOLUTIONARY       EH2 V AH0 L UW1 SH AH0 N EH2 R IY0
```

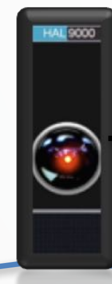
Annotation/transcription

- Speech data must be **segmented** and **annotated** in order to be useful to an ASR learning component.
  - Programs like Wavesurfer or Praat allow you to demarcate where a phoneme begins and ends in time.

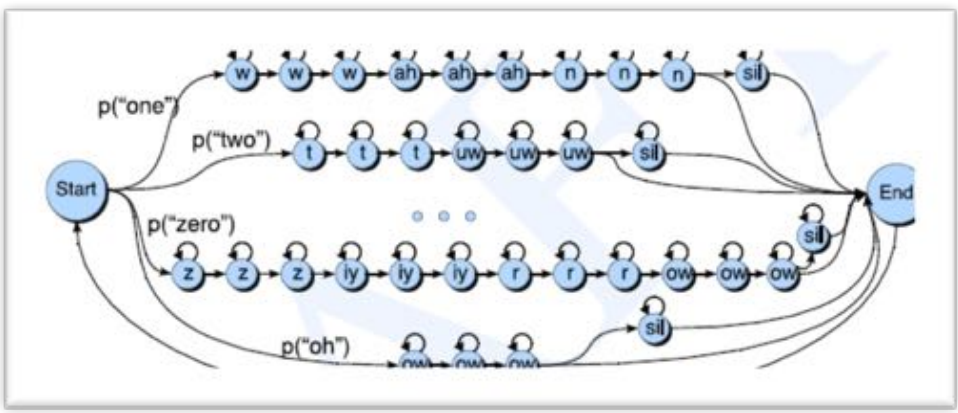


Putting it together?

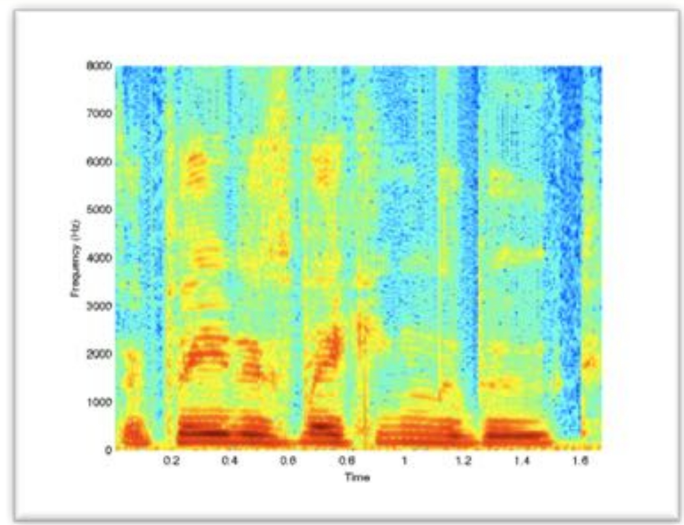




"open the pod bay doors"



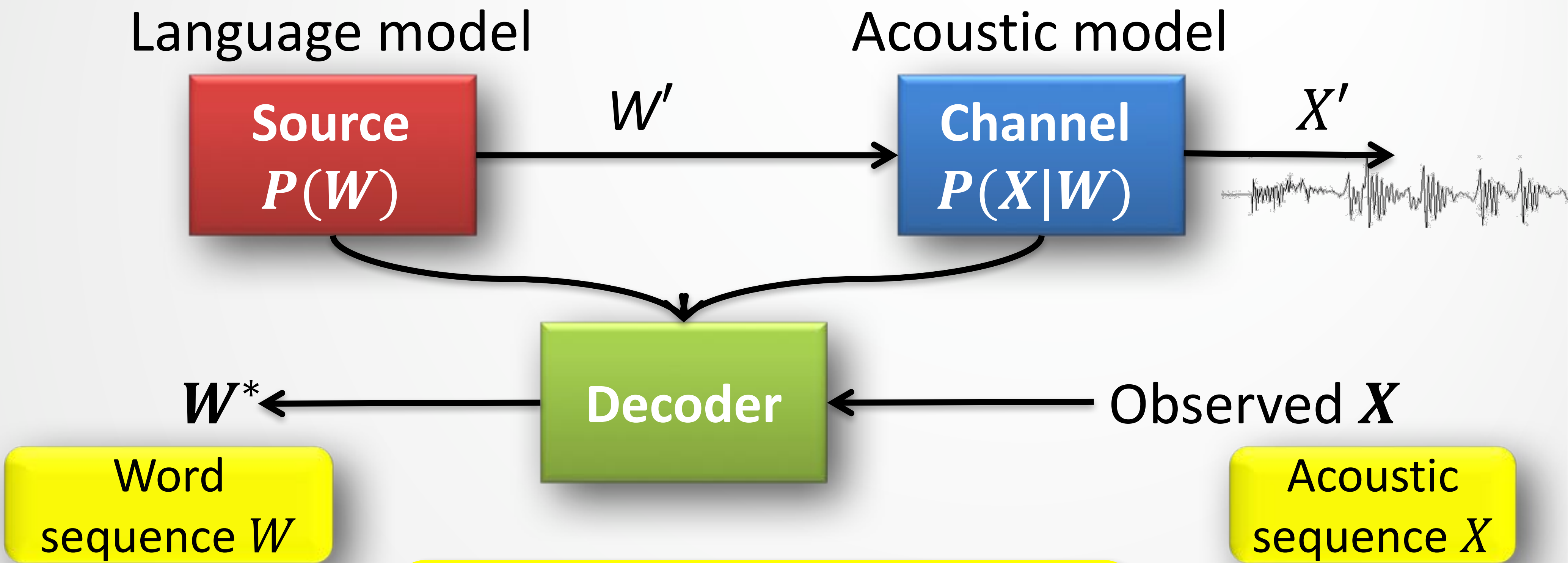
Language model



Acoustic model

UNIVERSITY OF TORONTO

The noisy channel model for ASR



$$W^* = \operatorname*{argmax}_W P(X|W)P(W)$$

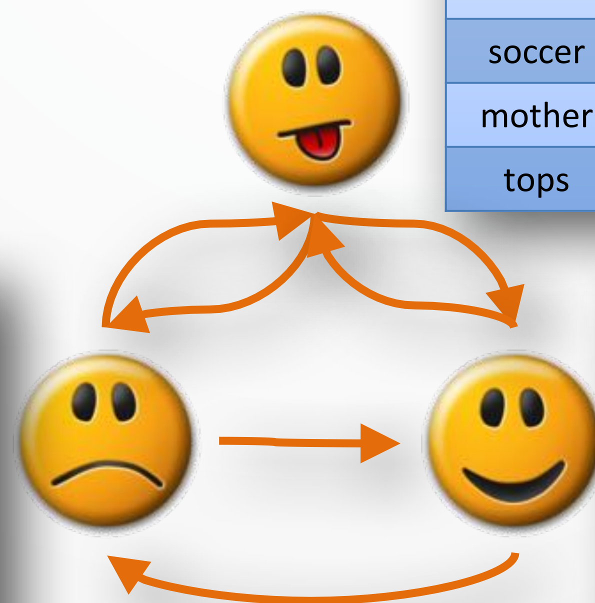
How to encode $P(X|W)$?

UNIVERSITY OF TORONTO

Reminder – discrete HMMs

- Previously we saw **discrete HMMs**: at each state we observed a discrete symbol from a finite set of discrete symbols.

word	P(word)
ship	0.1
pass	0.05
camp	0.05
frock	0.6
soccer	0.05
mother	0.1
tops	0.05

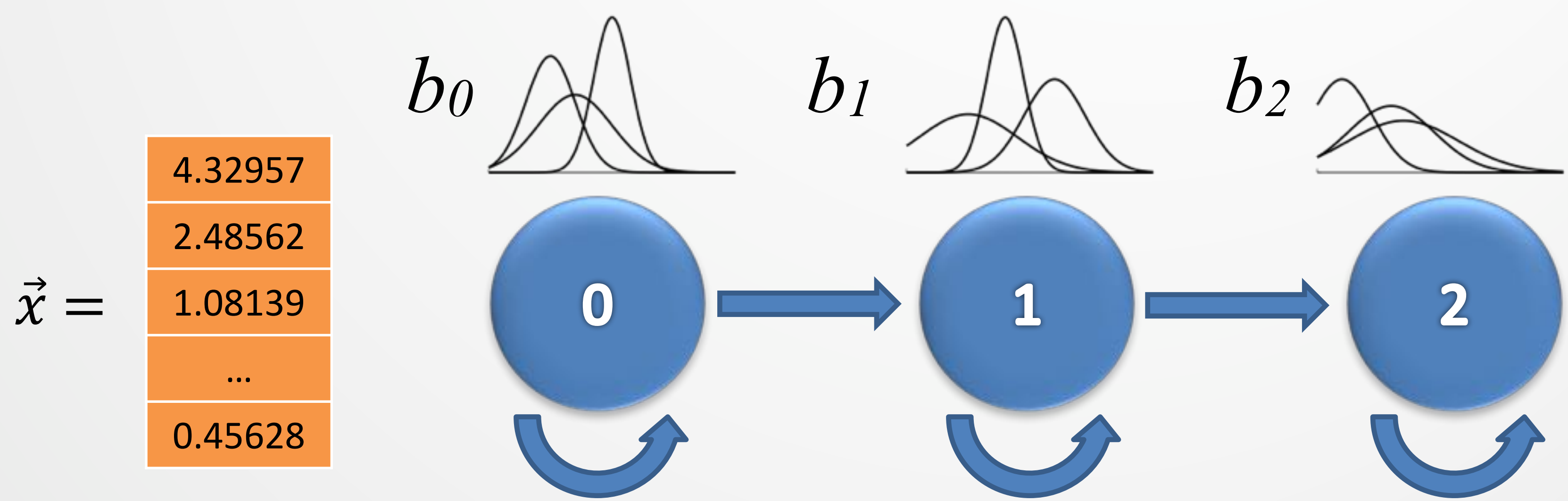


word	P(word)
ship	0.25
pass	0.25
camp	0.05
frock	0.3
soccer	0.05
mother	0.09
tops	0.01

word	P(word)
ship	0.3
pass	0
camp	0
frock	0.2
soccer	0.05
mother	0.05
tops	0.4

Continuous HMMs (CHMM)

- A **continuous HMM** has observations that are distributed over continuous variables.
  - Observation probabilities, b_i, are also continuous.
  - E.g., here $b_0(\vec{x})$ tells us the probability of seeing the (multivariate) continuous observation $\vec{x}$ while in state 0.

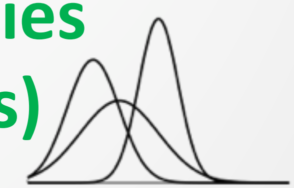


UNIVERSITY OF TORONTO

Defining CHMMs

- Continuous HMMs are very similar to discrete HMMs.
  - $S = \{s_1, \dots, s_N\}$: set of states (e.g., subphones)
  - $X = \mathbb{R}^{42}$: **continuous observation space**

$$\theta \left\{ \begin{array}{ll} \bullet\ \Pi = \{\pi_1, \dots, \pi_N\} & : \text{initial state probabilities} \\ \bullet\ A = \{a_{ij}\}, i, j \in S & : \text{state transition probabilities} \\ \bullet\ B = b_i(\vec{x}), i \in S, \vec{x} \in X & : \textbf{state output probabilities} \\ & \ \ \textbf{(i.e., Gaussian mixtures)} \end{array} \right.$$



yielding

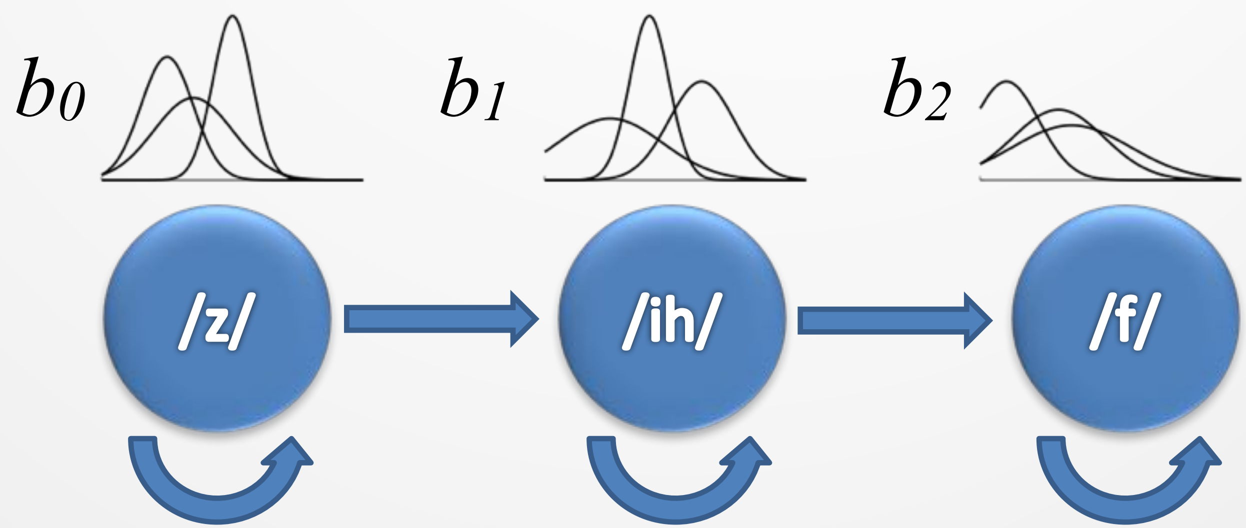
- $Q = \{q_0, \dots, q_T\}, q_i \in S$: state sequence
- $\mathcal{O} = \{\sigma_0, \dots, \sigma_T\}, \sigma_i \in X$: observation sequence

UNIVERSITY OF TORONTO

Word-level HMMs?

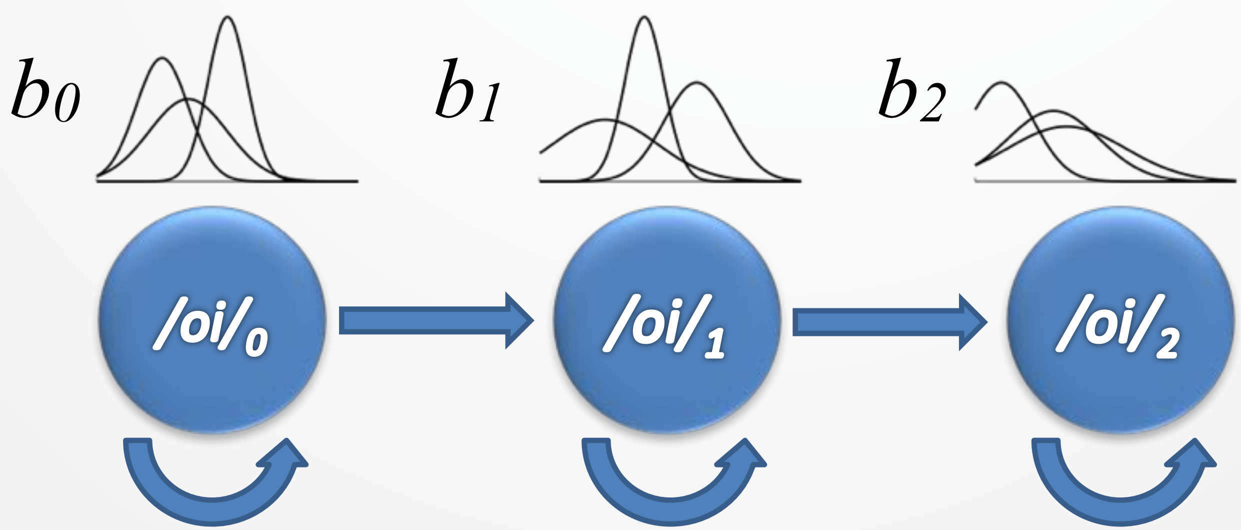
- Imagine that we want to learn an HMM for each word in our lexicon (e.g., 60K words → 60K HMMs).
- No, thank you! Zipf's law tells us that *many* words occur *very* infrequently.
  - 1 (or a few) training examples of a word is ***not*** enough to train a model as highly parameterized as a CHMM.

- In a word-level HMM, each state might be a phoneme.



Phoneme HMMs

- Phonemes *change* over time – we model these dynamics by building one HMM for *each* phoneme.
  - Tristate phoneme models are popular.
    - The centre state is often the 'steady' part.



tristate phoneme model (e.g., */oi/*)

Phoneme HMMs

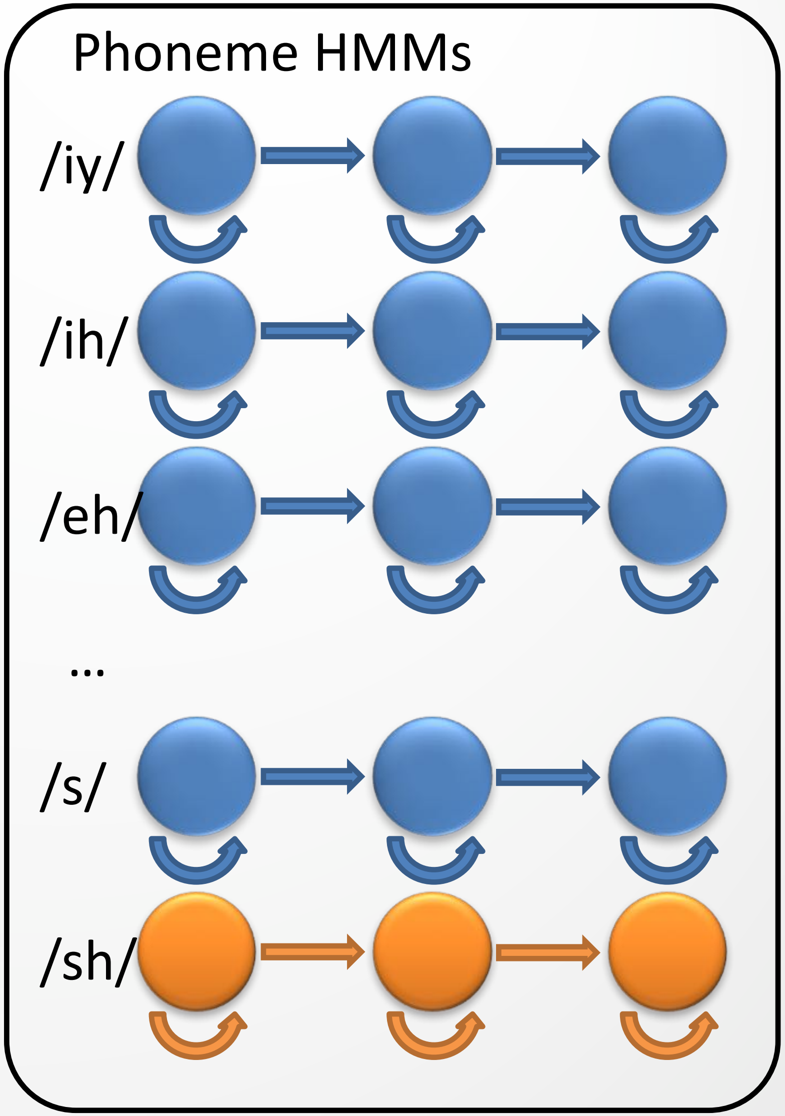
- We train each phoneme HMM using ***all*** sequences of that phoneme.
  - Even from different words.

```
...
64  85   ae
85  96   sh
96  102  epi
102 106  m
...
```

annotation

		Time, *t*				
		…	85	…	96	…
MFCC	1	…		…		…
	2	…		…		…
	3	…		…		…
	…	…	…	…	…	…
	42	…		…		…

observations



Combining models

- We can learn an *N*-gram **language model** from word-level transcriptions of speech data.
  - These models are discrete and are trained using MLE.

- Our phoneme HMMs together constitute our **acoustic model**.
  - Each phoneme HMM tells us how a phoneme 'sounds'.

- We can **combine** these models by **concatenating** phoneme HMMs together according to a known lexicon.
  - We use a word-to-phoneme dictionary.

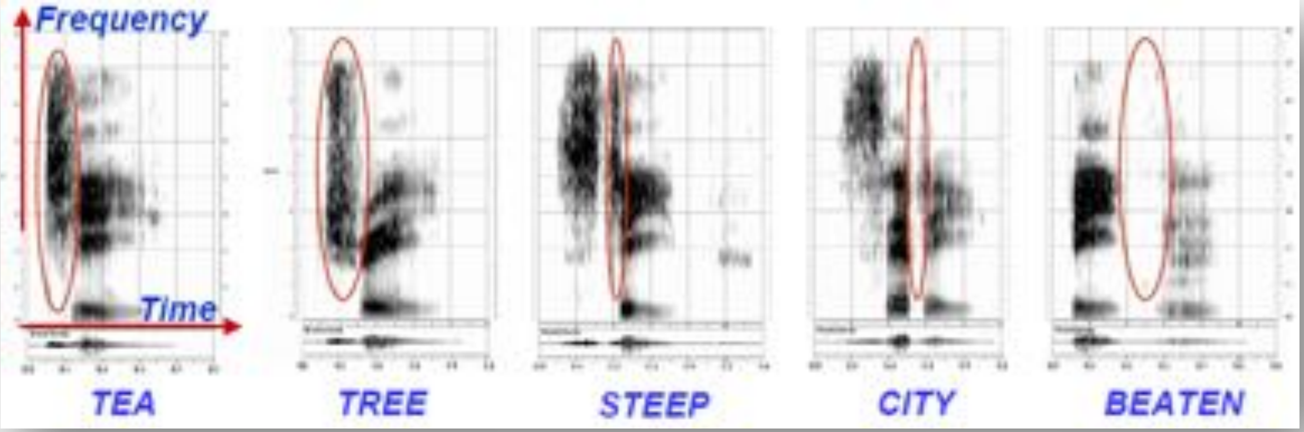
Combining models

- If we know how phonemes combine to make words, we can simply **concatenate** together our phoneme models by inserting and **adjusting** transition weights.
  - e.g., *Zipf* is pronounced */z ih f/*, so…

(It's a tiny bit more complicated than this –
normally phoneme HMMs have special 'handle' states
at either end that connect to other HMMs)

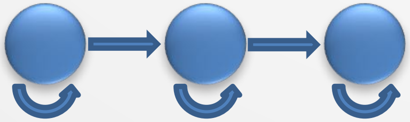
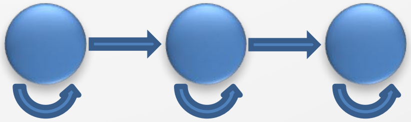
Co-articulation and triphones

- **Co-articulation**: *n.* When a phoneme is influenced by adjacent phonemes.



- A **triphone HMM** captures co-articulation.
  - Triphone model /a-b+c/ is phoneme **b** when preceded by **a** and followed by **c**.

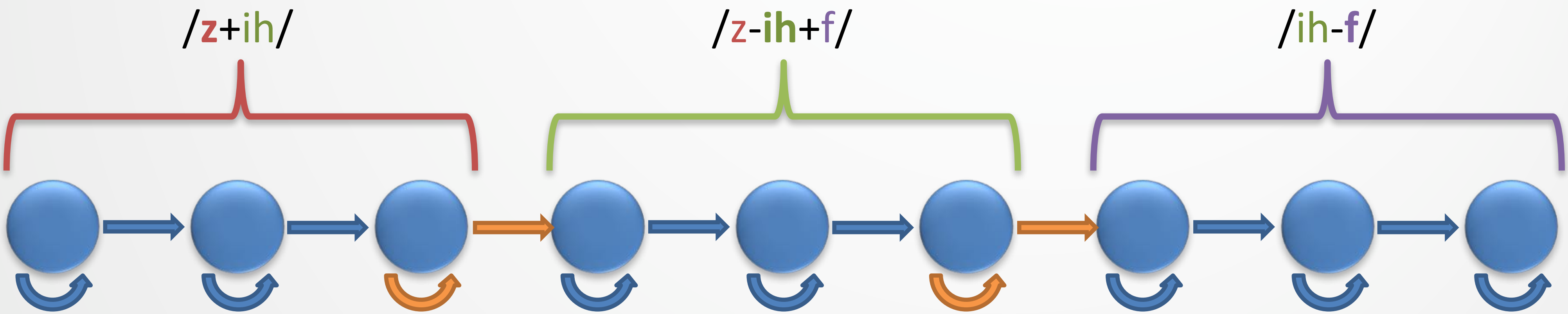
Two (of many) triphone HMMs for /t/

/s-t+iy/  /iy-t+eh/ 

UNIVERSITY OF TORONTO

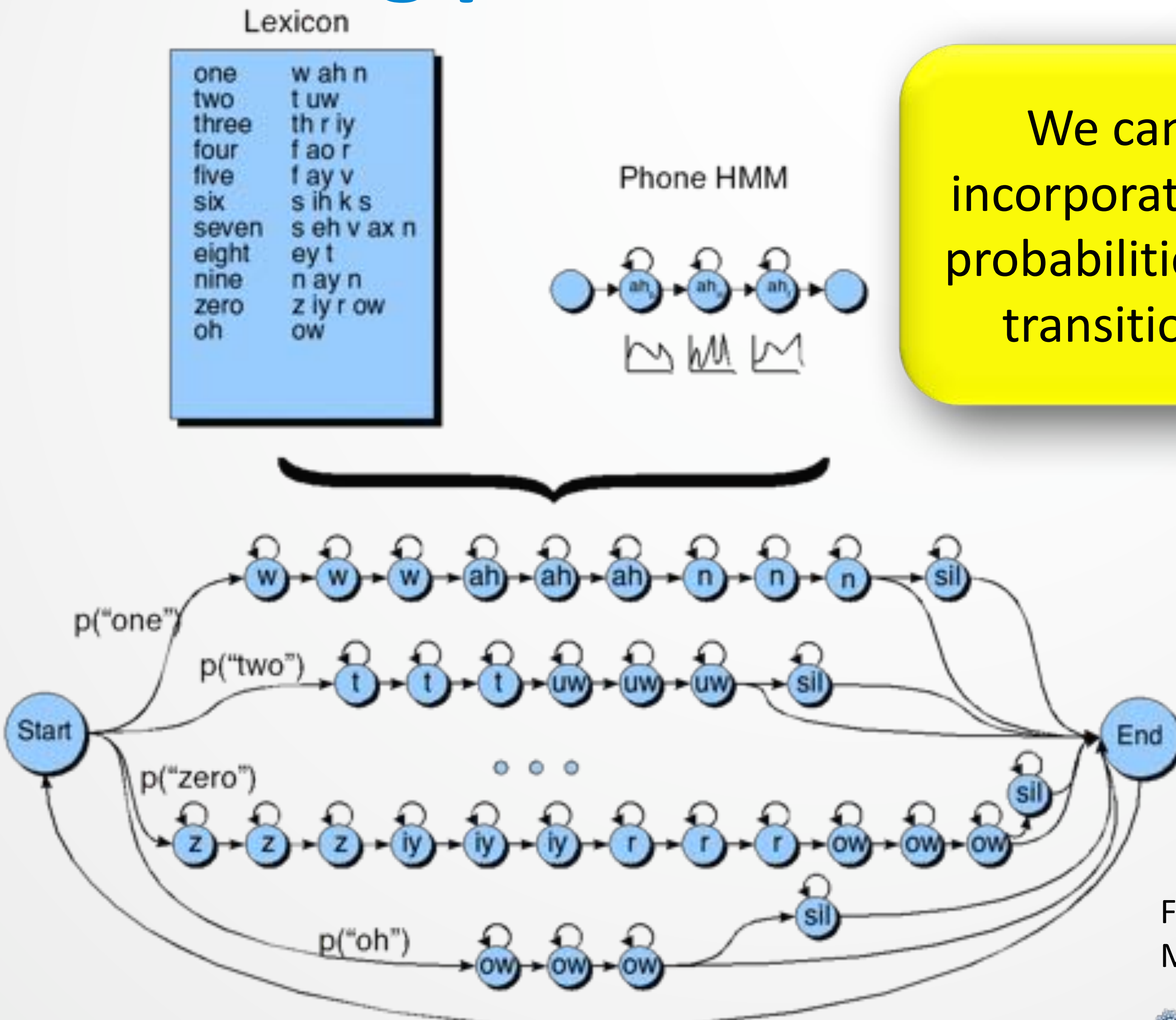
Combining triphone HMMs

- Triphone models can only connect to other triphone models that 'match'.



UNIVERSITY OF TORONTO

Concatenating phoneme models

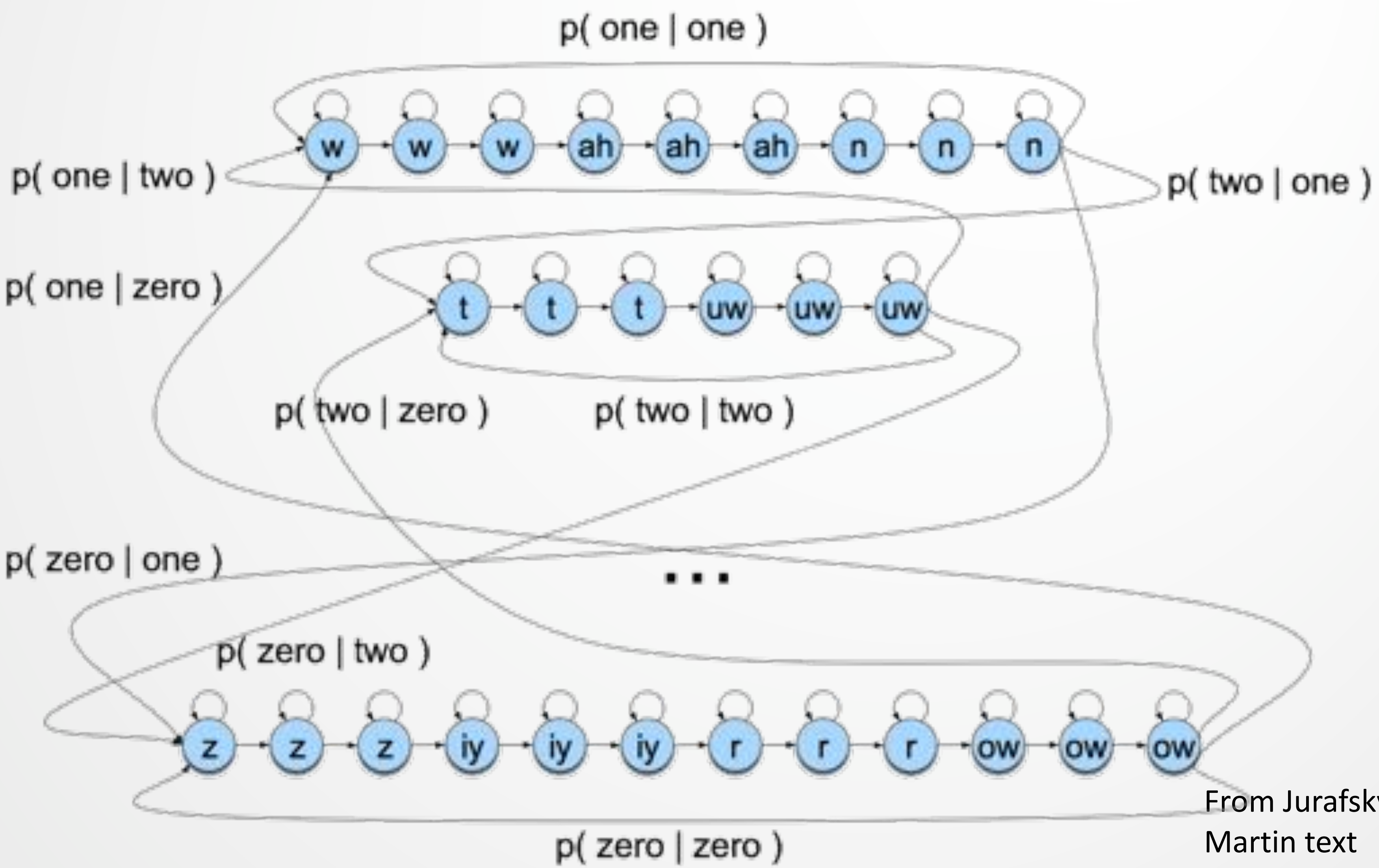


We can easily incorporate unigram probabilities through transitions, too.

From Jurafsky & Martin text

UNIVERSITY OF TORONTO

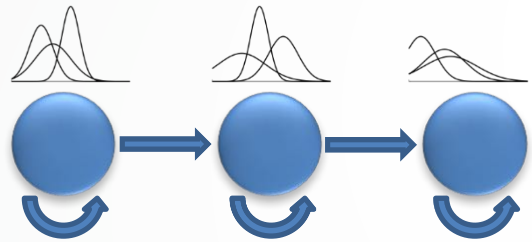
Bigram models



From Jurafsky & Martin text

UNIVERSITY OF TORONTO

Using CHMMs



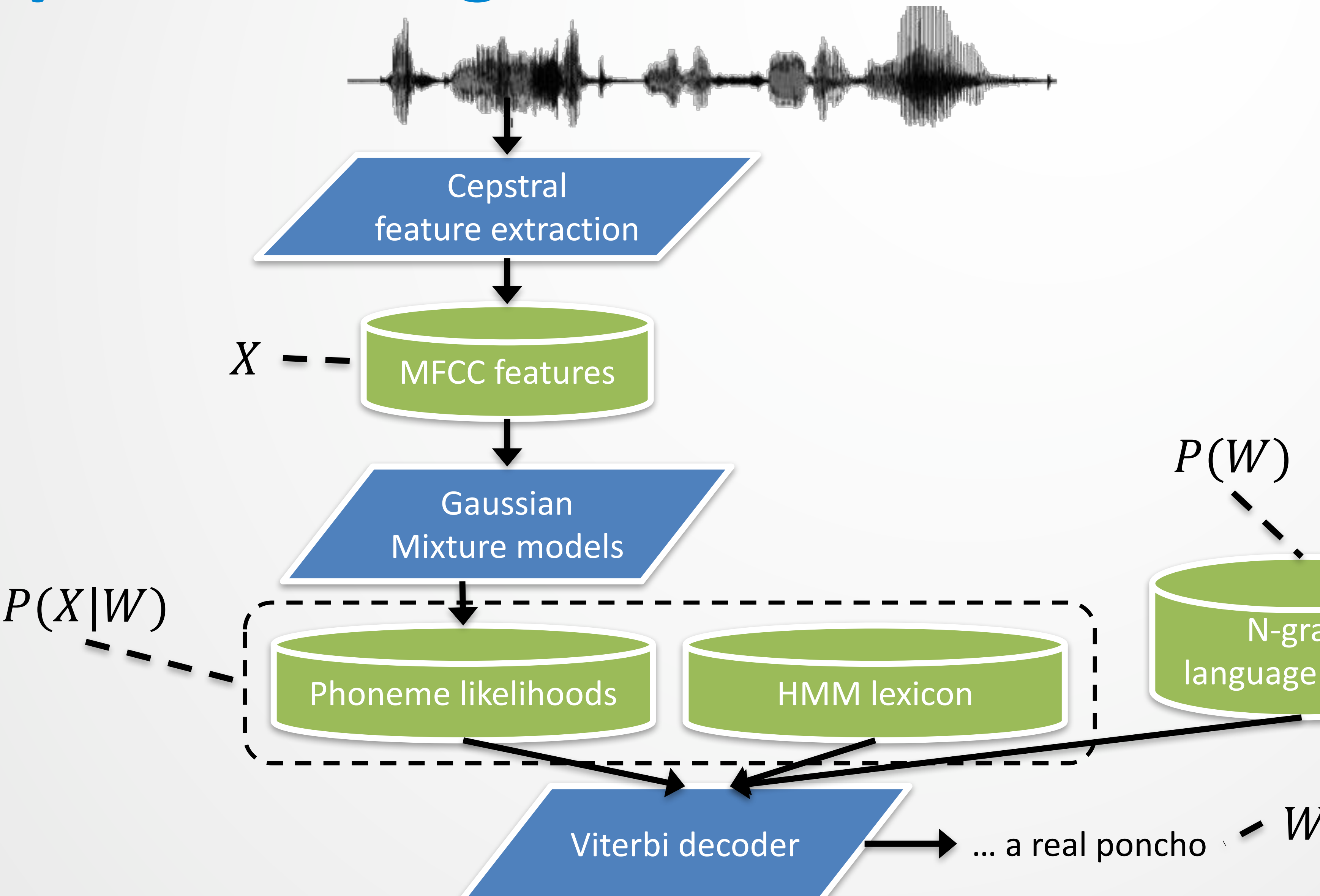
- As before, these HMMs are **generative** models that encode statistical knowledge of how output is **generated**.

- We **train** CHMMs with **Baum-Welch** (a type of Expectation-Maximization), as we did before with discrete HMMs.
  - Here, the observation parameters, $b_i(\vec{x})$, are adjusted using the GMM training 'recipe' from last lecture.

- We find the best state sequences using **Viterbi**, as before.
  - Here, the best state sequence gives us a **sequence of phonemes** and **words**.

UNIVERSITY OF TORONTO

Speech recognition architecture



UNIVERSITY OF TORONTO

Speech databases

- Large-vocabulary continuous ASR is meant to encode full conversational speech, with a vocabulary of >64K words.
  - This requires *lots* of data to train our models.

- The **Switchboard** corpus contains 2430 conversations spread out over about 240 hours of data (~14 GB).
- The **TIMIT** database contains 63,000 sentences from 630 speakers.
  - Relatively small (~750 MB), but very popular.
- Speech data from conferences (e.g., **TED**) or from broadcast news tends to be between 3 GB and 30 GB.

Aspects of ASR systems in the world

- **Speaking mode**: **Isolated** word (e.g., "*yes*") vs. **continuous** (e.g., *"Siri, ask Cortana for the weather"*)
- **Speaking style**: **Read** speech vs. **spontaneous** speech; the latter contains many **dysfluencies** (e.g., stuttering, *uh*, *like*, …)
- **Enrolment**: **Speaker-dependent** (all training data from one speaker) vs. **speaker-independent** (training data from many speakers).
- **Vocabulary**: **Small** (<20 words) or **large** (>50,000 words).
- **Transducer**: Cell phone? Noise-cancelling microphone? Teleconference microphone?

Signal-to-noise ratio

- We are often concerned with the **signal-to-noise ratio** (SNR), which measures the **ratio** between the power of a **desired signal** within a recording (P_{signal}, e.g., the human speech) and **additive noise** (P_{noise}).
  - Noise typically includes:
    - **Background noise** (e.g., people talking, wind),
    - **Signal degradation**. This is *normally* 'white' noise produced by the medium of transmission.

$$SNR_{db} = 10 \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$$

You don't have to memorize this formula.

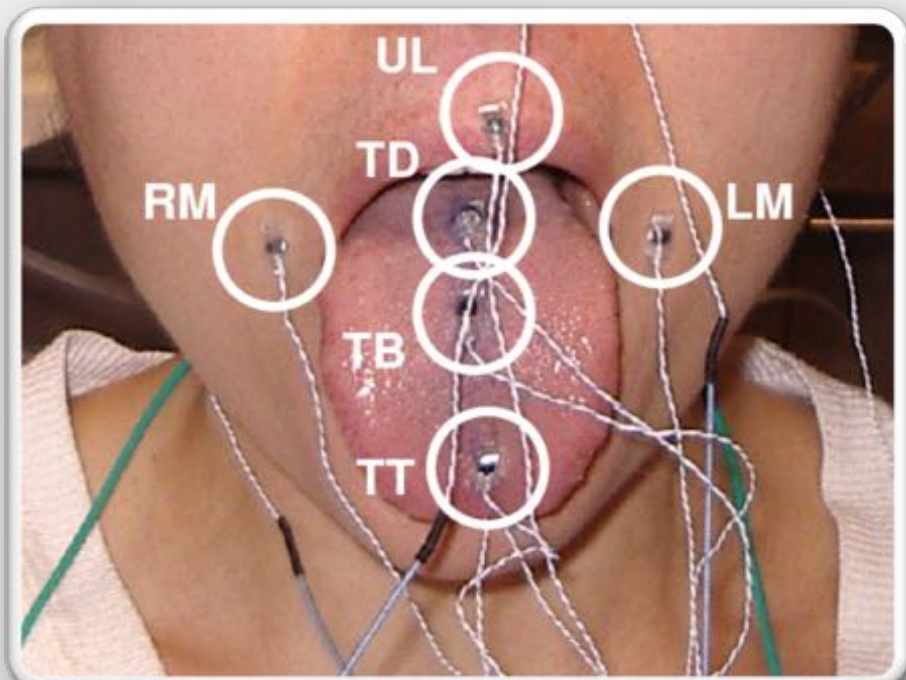
- High SNR_{db} is >30dB. Low SNR_{db} is < 10 dB.

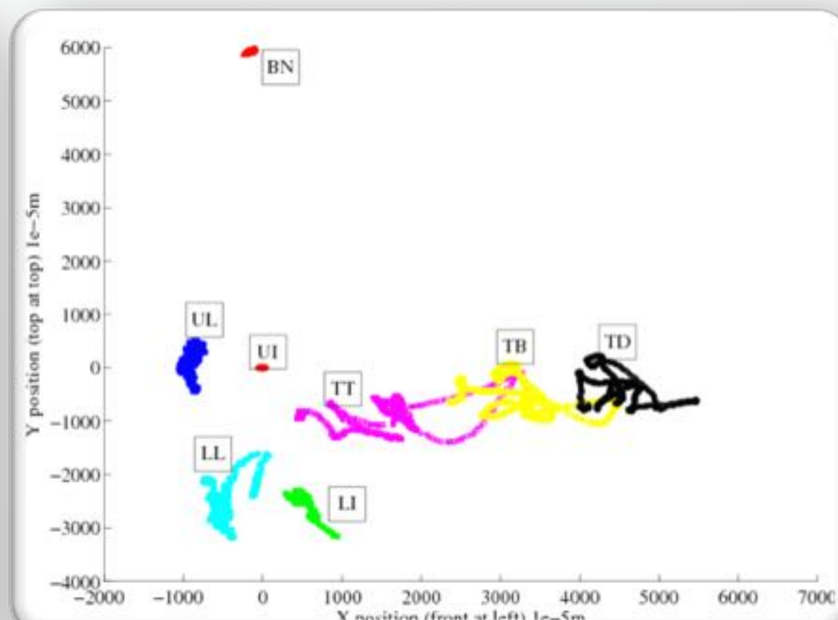
Audio-visual speech methods

- Observing the **vocal tract** directly, rather than through inference, can be very helpful in automatic speech recognition.
- The shape and aperture of the mouth gives some clues as to the phoneme being uttered.
  - Depending on the level of invasiveness, we can even measure the glottis and tongue directly.

Example of articulatory data

- TORGO was built to train augmented ASR systems.
  - **9** subjects **with cerebral palsy (1 with ALS)**, **9 matched controls**.
  - Each reads 500—1000 prompts over **3 hours** that cover **phonemes** and **articulatory contrasts** (e.g., *meat* vs. *beat*).
  - **Electromagnetic articulography** (and video) track points to <1 mm.

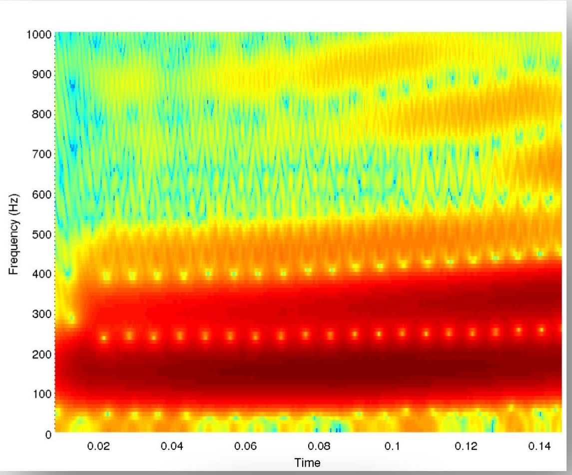


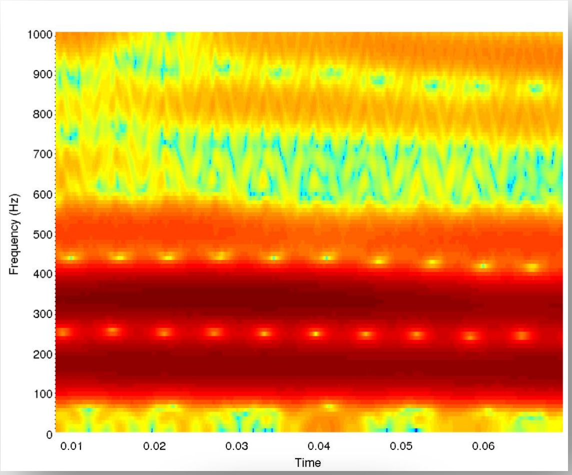


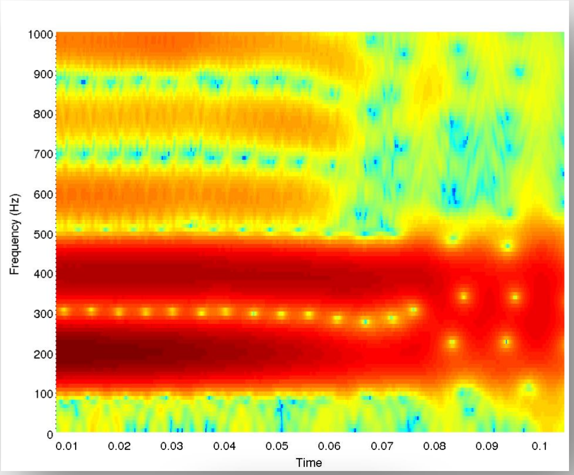
UNIVERSITY OF TORONTO

Example – Lip aperture and nasals

Acoustic spectrograms

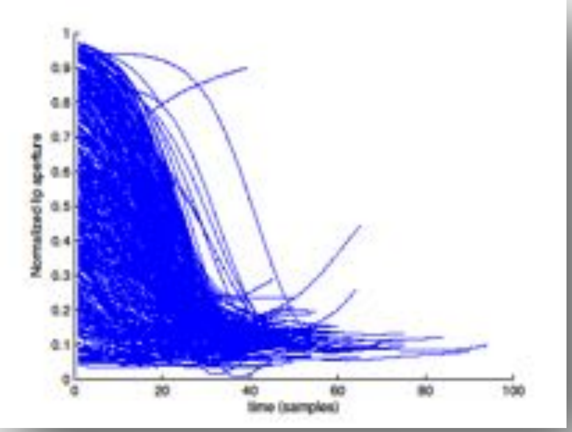


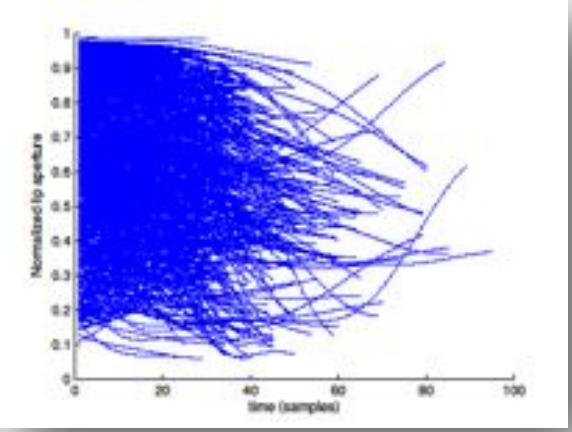


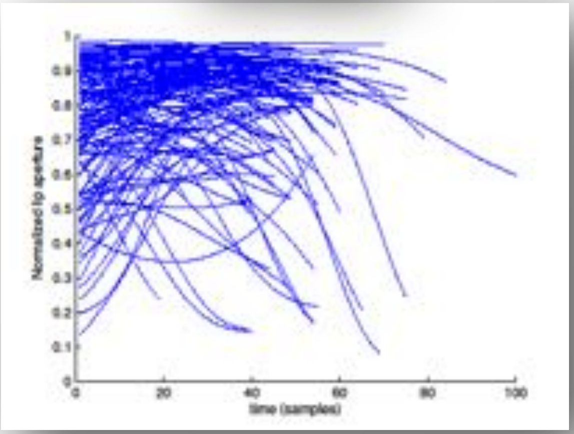


/m/ /n/ /ng/

Lip apertures over time







UNIVERSITY OF TORONTO

Evaluating ASR accuracy

- How can you tell how good an ASR system at recognizing speech?
  - E.g., if somebody said

    Reference: *how to recognize speech*

    but an ASR system heard

    Hypothesis: *how to wreck a nice beach*

    how do we quantify the error?

- One measure is **word accuracy**: *#CorrectWords/#ReferenceWords*
  - E.g., $2/4$, above
  - This runs into problems similar to those we saw with SMT.
    - E.g., the hypothesis '*how to recognize speech boing boing boing boing boing*' has 100% accuracy by this measure.
    - Normalizing by *#HypothesisWords* also has problems…

UNIVERSITY OF TORONTO

Word-error rates (WER)

- ASR enthusiasts are often concerned with **word-error rate (WER)**, which counts different **kinds** of errors that can be made by ASR at the word-level.
  - **Substitution error**: One word being mistook for another e.g., ‘***shift***’ given ‘*ship*’
  - **Deletion error**: An input word that is ‘skipped’ e.g. ‘*I Torgo*’ given ‘*I **am** Torgo*’
  - **Insertion error**: A ‘hallucinated’ word that was not in the input. e.g., ‘*This **Norwegian** parrot is no more*’ given ‘*This parrot is no more*’

Evaluating ASR accuracy

- But how to decide which errors are of each type?
- E.g., Reference: *how to recognize speech*
  Hypothesis: *how to wreck a nice beach*,

- It's not so simple: '*speech*' seems to be mistaken for '*beach*', except the /s/ phoneme is incorporated into the preceding hypothesis word, '*nice*' (/n ay s/).
  - Here, '*recognize*' seems to be mistaken for '*wreck a nice*'
    - Are each of '*wreck a nice*' **substitutions** of '*recognize*'?
    - Is '*wreck*' a **substitution** for '*recognize*'?
      - If so, the words '*a*' and '*nice*' must be **insertions**.
    - Is '*nice*' a **substitution** for '*recognize*'?
      - If so, the words '*wreck*' and '*a*' must be **insertions**.

Levenshtein distance

- In practice, ASR people are often more concerned with **overall** WER, and don't care about how those errors are partitioned.
  - E.g., 3 substitution errors are 'equivalent' to 1 substitution plus 2 insertions.

- The **Levenshtein** distance is a straightforward algorithm based on dynamic programming that allows us to compute overall WER.

Levenshtein distance

Allocate matrix $R[n+1, m+1]$ // where n is the number of reference words
// and m is the number of hypothesis words

Initialize $R[0,0] := 0$, **and** $R[i,j] := \infty$ **for all other** $i = 0$ **or** $j = 0$

for $i := 1..n$ // #ReferenceWords

  for $j := 1..m$ // #Hypothesis words

    $R[i,j] := \min($ $R[i-1,j]+1,$ // **deletion**

      $R[i-1,j-1],$ // if the i^{th} reference word and
      // the j^{th} hypothesis word **match**

      $R[i-1,j-1]+1,$ // if they differ, i.e., **substitution**

      $R[i,j-1]+1\ )$ // **insertion**

Return $100\times R[n,m]/n$

UNIVERSITY OF TORONTO

Levenshtein distance – initialization

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞						
	to	∞						
	recognize	∞						
	speech	∞						

The value at cell (i, j) is the **minimum** number of **errors** necessary to align i with j.

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0					
	to	∞						
	recognize	∞						
	speech	∞						

- $R[1,1] = \min(\infty + 1, (0), \infty + 1) = 0$ (match)
- We put a little arrow in place to indicate the choice.
  - 'Arrows' are normally stored in a **backtrace matrix**.

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞						
	recognize	∞						
	speech	∞						

- We continue along for the first reference word…
  - These are all **insertion** errors

UNIVERSITY OF TORONTO

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞	1	0	1	2	3	4
	recognize	∞						
	speech	∞						

- And onto the second reference word

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞	1	0	1	2	3	4
	recognize	∞	2	1	**1**	**2**	**3**	**4**
	speech	∞						

- Since *recognize* ≠ *wreck*, we have a **substitution** error.
- At some points, you have >1 possible path as **indicated**.
  - We can prioritize types of errors arbitrarily.

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞	1	0	1	2	3	4
	recognize	∞	2	1	1	2	3	4
	speech	∞	3	2	2	2	3	**4**

- And we finish the grid.
- There are $R[n, m] = 4$ word errors and a WER of $4/4 = 100\%$.
  - WER can be greater than 100% (relative to the reference).

UNIVERSITY OF TORONTO

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞	1	0	1	2	3	4
	recognize	∞	2	1	1	2	3	4
	speech	∞	3	2	2	2	3	**4**

- If we want, we can **backtrack** using our arrows to find the proportion of substitution, deletion, and insertion errors.

Levenshtein distance

			hypothesis					
		-	how	to	wreck	a	nice	beach
Reference	-	0	∞	∞	∞	∞	∞	∞
	how	∞	0	1	2	3	4	5
	to	∞	1	0	1	2	3	4
	recognize	∞	2	1	1	2	3	4
	speech	∞	3	2	2	2	3	**4**

- Here, we estimate 2 **substitution** errors and 2 **insertion** errors.
- Arrows can be encoded within a special **backtrace** matrix.

Recent performance

Corpus	Speech type	Lexicon size	ASR WER (%)	Human WER (%)
Digits	Spontaneous	10	0.3 %	0.009 %
Phone directory	Read	1000	3.6 %	0.1 %
Wall Street Journal	Read	64,000	6.6 %	1 %
Radio news	Mixed	64,000	13.5 %	-
Switchboard (telephone)	conversation	10,000	19.3 %	4 %