natural language computing

CSC401/2511 – Natural Language Computing – Spring 2022

Lecture 13

University of Toronto

This lecture

- An extractive summary of the course.
- Open office hours will follow, TBD

Exam

- 12 April from 19h—22h.

All students
EX 200
Exam Centre 255 McCaul Street 

May be subject to change

- **No aids allowed** – your desk should have nothing but:
  - Your UofT ID,
  - The exam, and
  - A writing implement.

Structure

- Following the format of previous years:
  - 20 **multiple-choice** questions [40 marks]
    - 4 options each.
  - 10 s**hort-answer** questions [30 marks]
    - Some of these involve simply giving a definition. Others involve some calculation.
  - 3 **subject-specific** questions [30 marks]
    - These questions involve a small component of original thinking.

8. Melamed's method of sentence alignment works by ...
   (a) minimizing the costs of alignments according to the lengths of the aligned sentences.
   (b) minimizing the costs of alignments according to the lengths of the aligned words.
   (c) estimating cognates based on 4-graphs.
   (d) estimating cognates based on longest common subsequences.
9. Greedy decoding in statistical machine translation iteratively updates the best guess of the English translation E^*, given the French sentence F, according to ...
   (a) transformations of words and alignments.
   (b) transformations of words only.
   (c) the total cost of alignment.
   (d) the total number of matching cognates.
10. Which of these phonemes is **not** voiced?
   (a) /*b*/.
   (b) /*ih*/.
   (c) /*m*/.
   (d) /*k*/.
11. The Nyquist rate is ...
   (a) the rate at which the glottis vibrates.
   (b) twice the rate at which the glottis vibrates.
   (c) twice the maximum frequency preserved in a sampled signal.
   (d) twice the sampling rate of a sampled signal.
12. Which feature is known to correlate positively with a sentence's selection into an extractive text summary in the news domain?
   (a) Early position in the document being summarized.
   (b) High function-word to content-word ratio.
   (c) High number of stigma words.
   (d) None of the above.

Short answer

2. State Bayes's Rule.

3. Name and define the three types of text-to-speech synthesis architectures. Give one advantage each architecture has over the others.

We can work it out

SMT 2. (5 marks)

Given the two reference translations below, compute the BLEU score for each of the two candidate translations, assuming that you only consider unigrams and bigrams, and that there is no cap. *Hint:* Your results should be of the form x^y where x is a fraction or some other term, and y is a positive or negative fraction.

Reference 1 Use the Force Luke

Reference 2 Use some Force Luke

Candidate 1 Use some of the Force

Candidate 2 Use the Force

Hints for studying

- **Definitions**: *n.pl*. Terms that are useful to know.
  - Highlights are also useful to know.
- Not all definitions/highlights are in the exam.
- Not all things on the exam have been highlighted.
  - This review lecture is likewise not a substitute for the rest of the material in this course.

Hints for studying

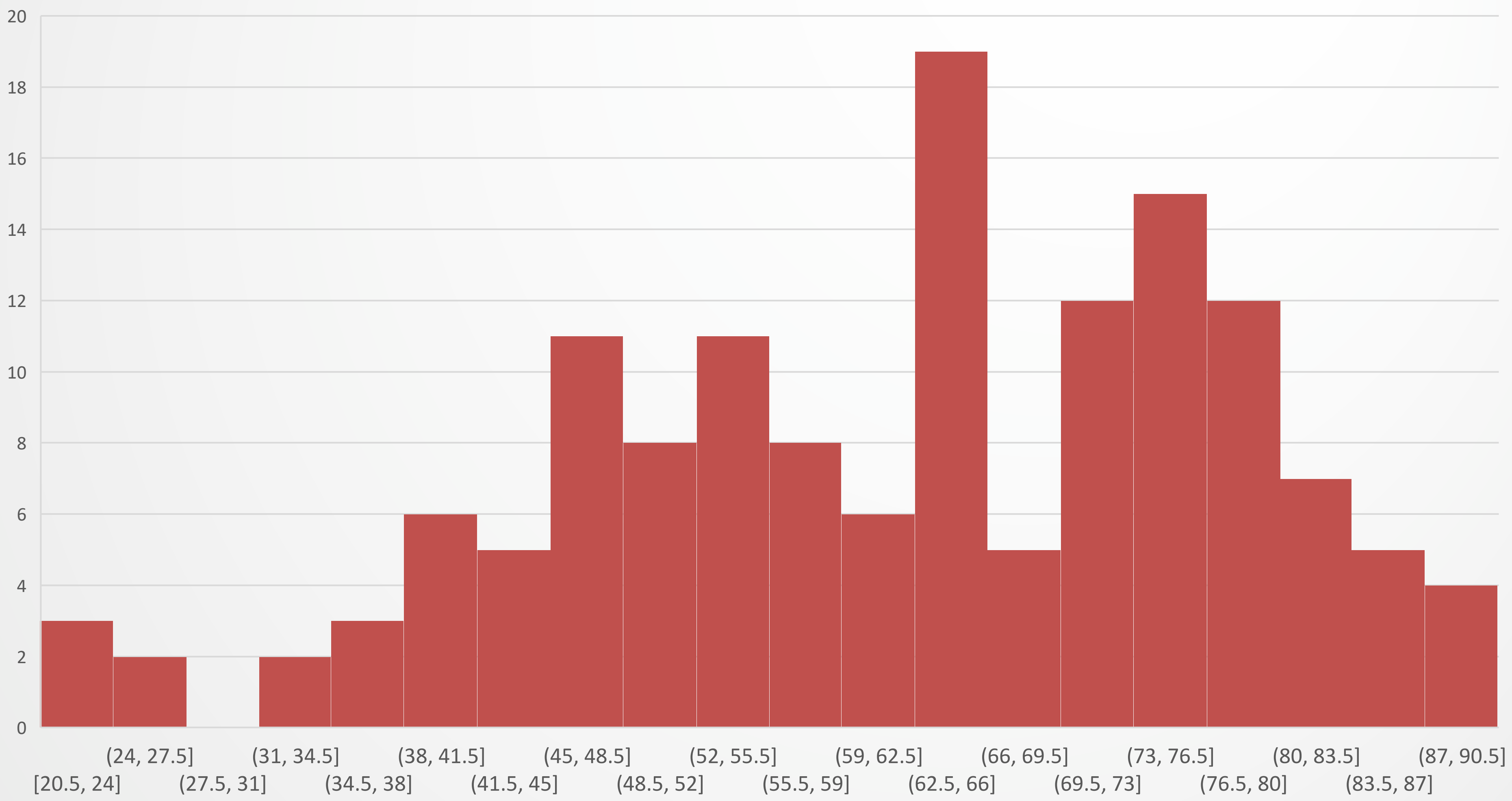
- Go through the **practice exam** from this year.
- Work out **worked-out examples** for yourself, ideally more than once.
- I find it helpful to **relax** before an exam.

Exam material

- The exam covers all material in the lectures and assignments **except**:
    - Material in the bonuses of assignments, and
    - Slides with ‘Aside’ in the title.

- The reading material (e.g., Manning & Schütze) provides background to concepts discussed in class.
    - If a concept appears in a linked paper but not in the lectures/assignments, you don’t need to know it, even if it’s **very interesting**.

2018 Final exam distribution



UNIVERSITY OF TORONTO

Categories of linguistic knowledge

- **Phonology**: the study of patterns of speech sounds.
  e.g., "read" → /r iy d/
- **Morphology**: how words can be changed by inflection or derivation.
  e.g., "read", "reads", "reader", "reading", …
- **Syntax**: the ordering and structure between words and phrases.
  e.g., *NounPhrase* → *det. adj. n.*
- **Semantics**: the study of how meaning is created by words and phrases.
  e.g., "book" → 
- **Pragmatics**: the study of meaning in broad contexts.

Corpora

- **Corpus**: *n.* A body of language data of a particular sort (*pl.* **corpora**).
- Most **valuable** corpora occur **naturally**
  - e.g., newspaper articles, telephone conversations, multilingual transcripts of the United Nations
- We use corpora to gather statistics; more is better (typically between 10^7 and 10^{12} tokens).

Very simple predictions

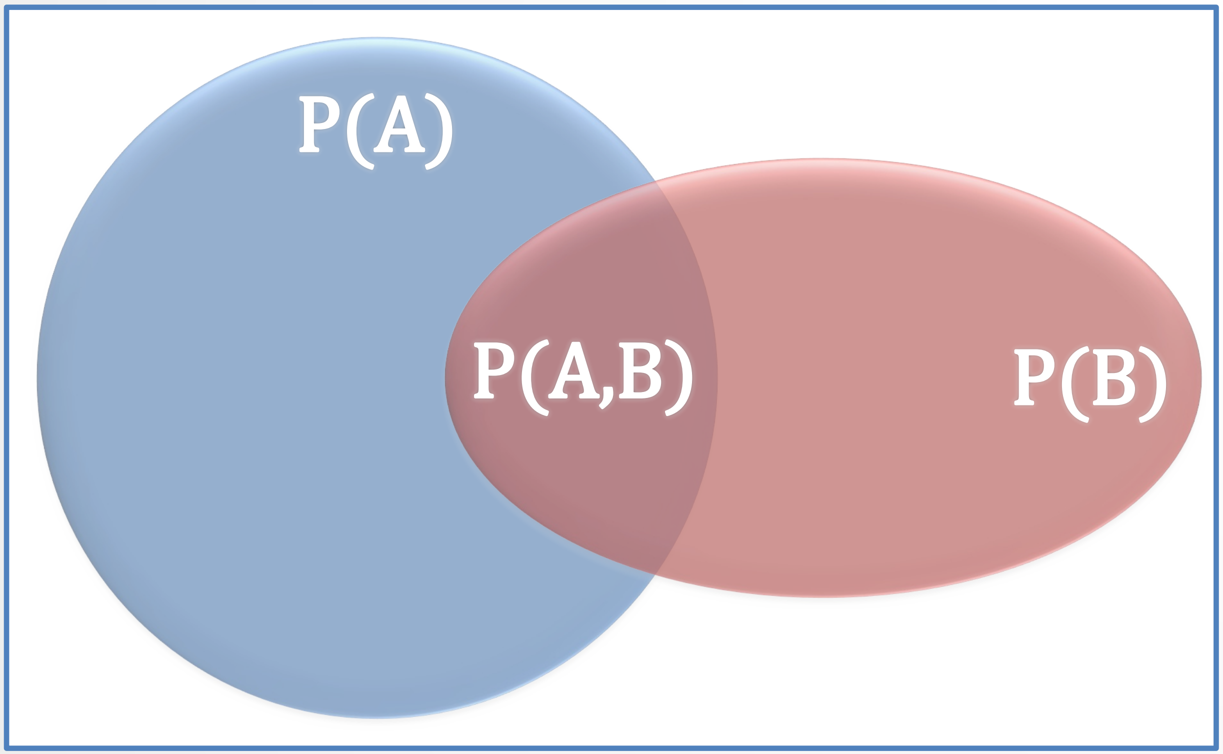
- A model at the heart of SMT, ASR, and IR…
- We want to know the probability of the **next** word given the **previous** words in a sequence.

- We can **approximate** conditional probabilities by counting occurrences in large corpora of data.
  - E.g., $P(food \mid I\ want\ Chinese) =$

$$\frac{P(I\ want\ Chinese\ food)}{P(I\ want\ Chinese)}$$

$$\approx \frac{Count(I\ want\ Chinese\ food)}{Count(I\ want\ Chinese)}$$

Bayes' theorem



$$P(A,B) = P(A)P(B|A)$$
$$P(A,B) = P(B)P(A|B)$$

Bayes theorem: $P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$

UNIVERSITY OF TORONTO

Maximum likelihood estimate

- **Maximum likelihood estimate (MLE)** of **parameters** θ in a **model** M, given **training data** T is

> the estimate that maximizes the likelihood of the *training data* using the *model.*

  - e.g., $\boldsymbol{T}$ is the Brown corpus,
    $\boldsymbol{M}$ is the bigram and unigram tables
    $\boldsymbol{\theta}_{(to|want)}$ is $P(to|want)$.

Sparsity of unigrams vs. bigrams

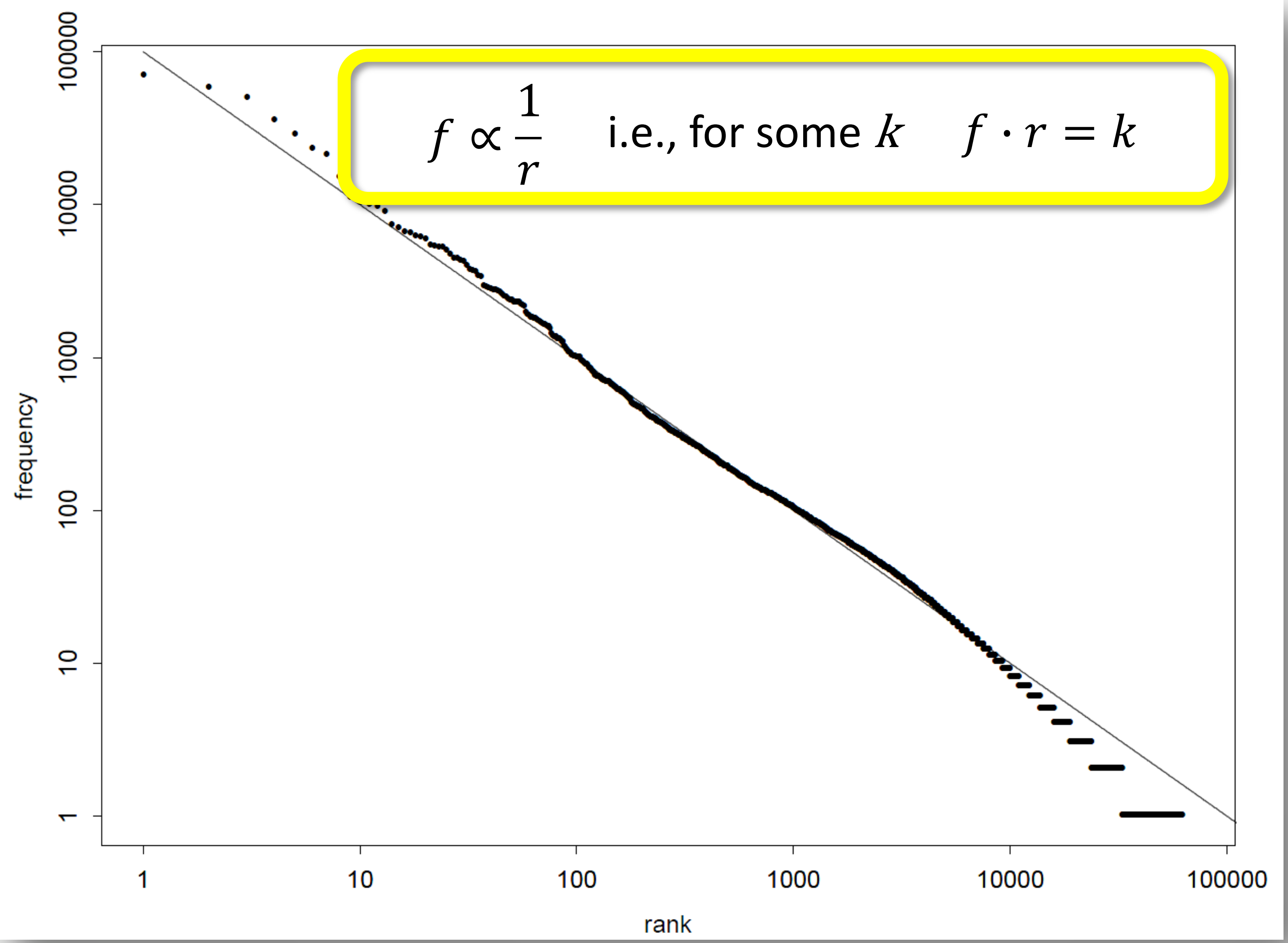
- E.g., we've seen lots of every unigram, but are missing many bigrams:

	I	want	to	eat	Chinese	food	lunch	spend
Unigram counts:	2533	927	2417	746	158	1093	341	278

$Count(w_{t-1},w_t)$		w_t							
		I	want	to	eat	Chinese	food	lunch	spend
w_{t-1}	I	5	827	0	9	0	0	0	2
	want	2	0	608	1	6	6	5	1
	to	2	0	4	686	2	0	6	211
	eat	0	0	2	0	16	2	42	0
	Chinese	1	0	0	0	0	82	1	0
	food	15	0	15	0	1	4	0	0
	lunch	2	0	0	0	0	1	0	0
	spend	1	0	1	0	0	0	0	0

UNIVERSITY OF TORONTO

Zipf’s law on the Brown corpus



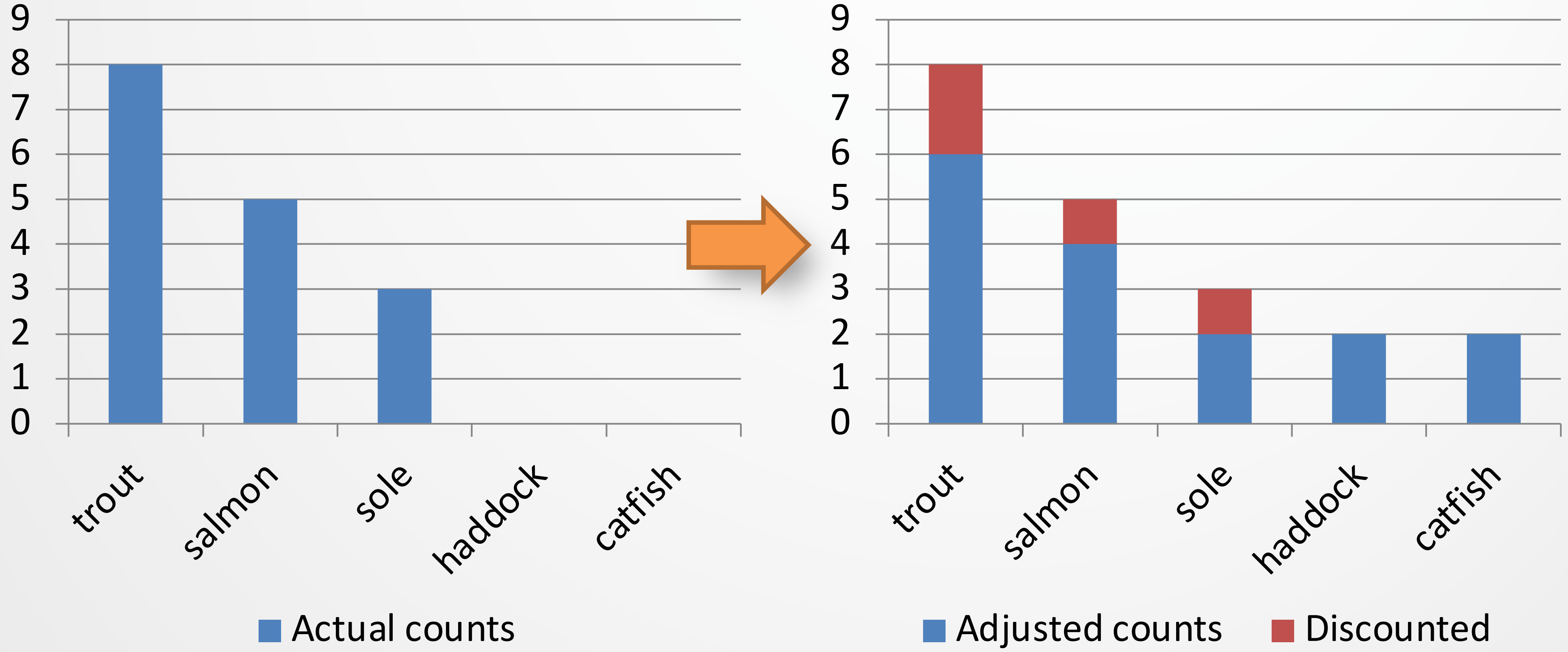
$$f \propto \frac{1}{r} \quad \text{i.e., for some } k \quad f \cdot r = k$$

From Manning & Schütze

UNIVERSITY OF TORONTO

Smoothing as redistribution

- Steal from the rich and give to the poor.
- E.g., $Count(I\ caught\ \ \cdot)$



Add-δ smoothing

- Laplace's method generalizes to the add-δ estimate :

$$\boxed{P_\delta(w_i) = \frac{C(w_i) + \delta}{N + \delta\|\mathcal{V}\|}}$$

- Consider alternative methods of smoothing.

UNIVERSITY OF TORONTO

Parts of speech (PoS)

- Linguists like to group words according to their **structural function** in building sentences.
  - This is similar to grouping Lego by their shapes.

- **Part-of-speech**: *n.* lexical category or morphological class.

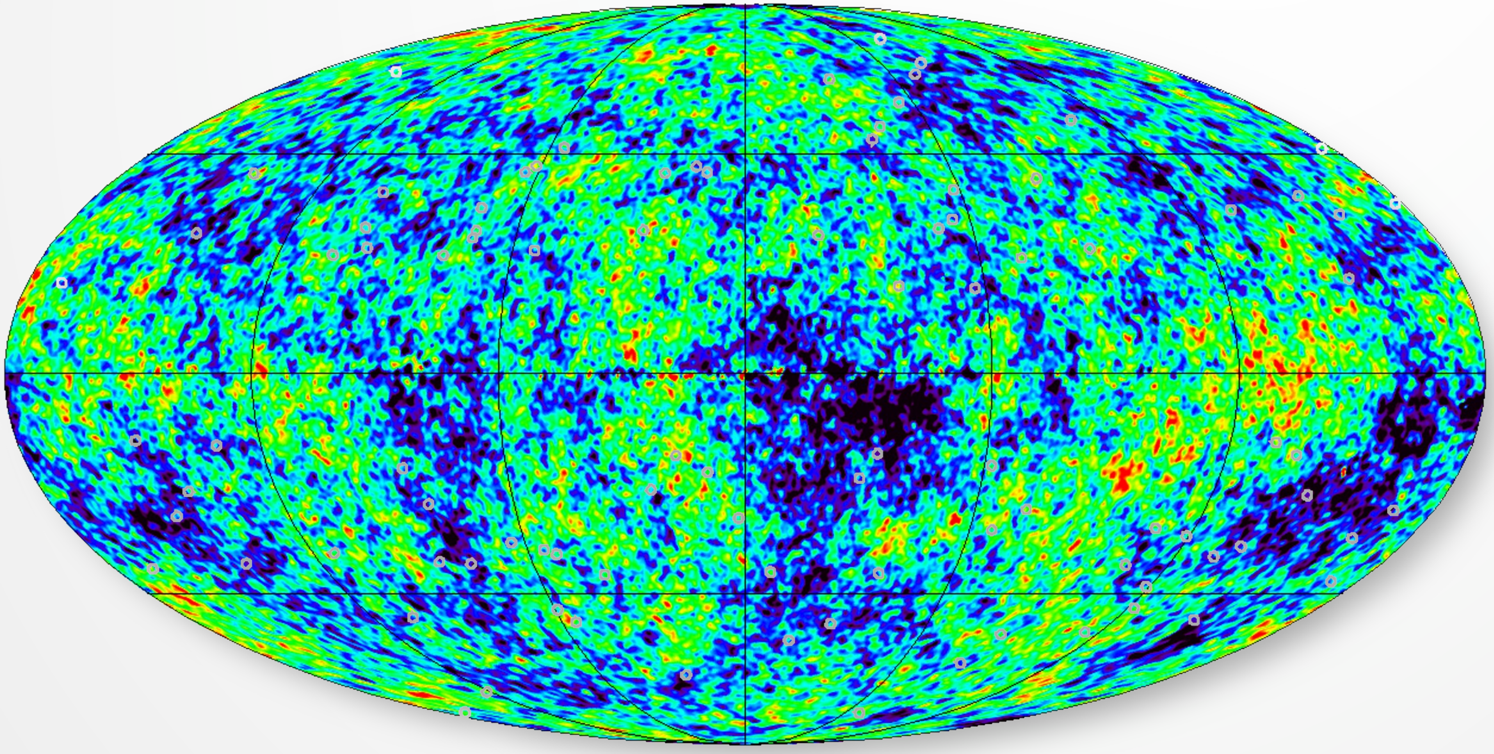
Nouns collectively constitute a part of speech (called *Noun*)

UNIVERSITY OF TORONTO

Parts of speech (PoS)

- Things that are useful to know about PoS:
  - **Content words** vs. **function words**
  - **Properties** of content words (e.g., number).
  - **Agreement**. Verbs and nouns should match in *number* in English (e.g., “*the dogs runs*” is ‘wrong’.)
  - What **PoS Tagging is**, and perhaps some vague idea of how to do it.

Information and entropy



Entropy

- **Entropy**: *n.* the **average** amount of information we get in observing the output of source S.

$$H(S) = \sum_i p_i I(w_i) = \sum_i p_i \log_2 \frac{1}{p_i}$$

ENTROPY

Note that this is ***very*** similar to how we define the expected value (i.e., 'average') of something:

$$E[X] = \sum_{x \in X} p(x)\, x$$

Joint entropy

- **Joint Entropy**: *n.* the **average** amount of information needed to specify multiple variables simultaneously.

$$H(X,Y) = \sum_x \sum_y p(x,y) \log_2 \frac{1}{p(x,y)}$$

Same general form as entropy, except you sum over each variable, and probabilities are joint

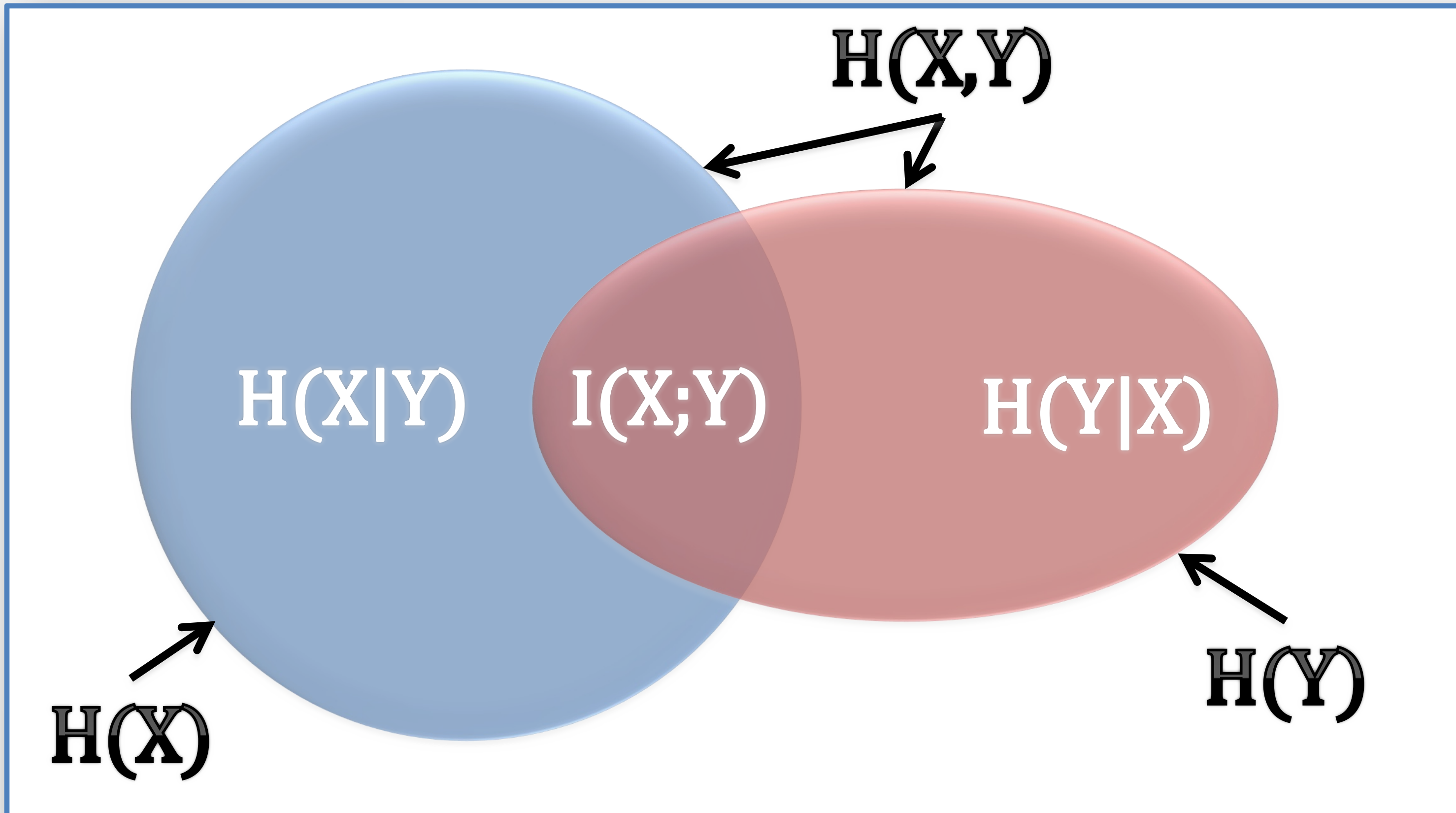
Conditional entropy

- **Conditional entropy**: *n.* the **average** amount of information needed to specify one variable **given that you know another**.

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

It's **an average of entropies** over all possible conditioning values.

Relations between entropies



$$H(X,Y) = H(X) + H(Y) - I(X;Y)$$

Mutual information

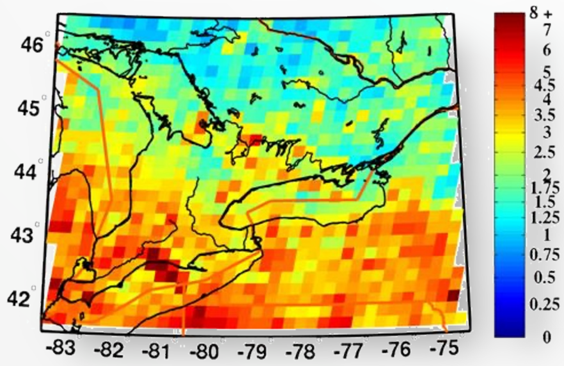
- **Mutual information**: *n.* the **average** amount of information shared between variables.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$
$$= \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

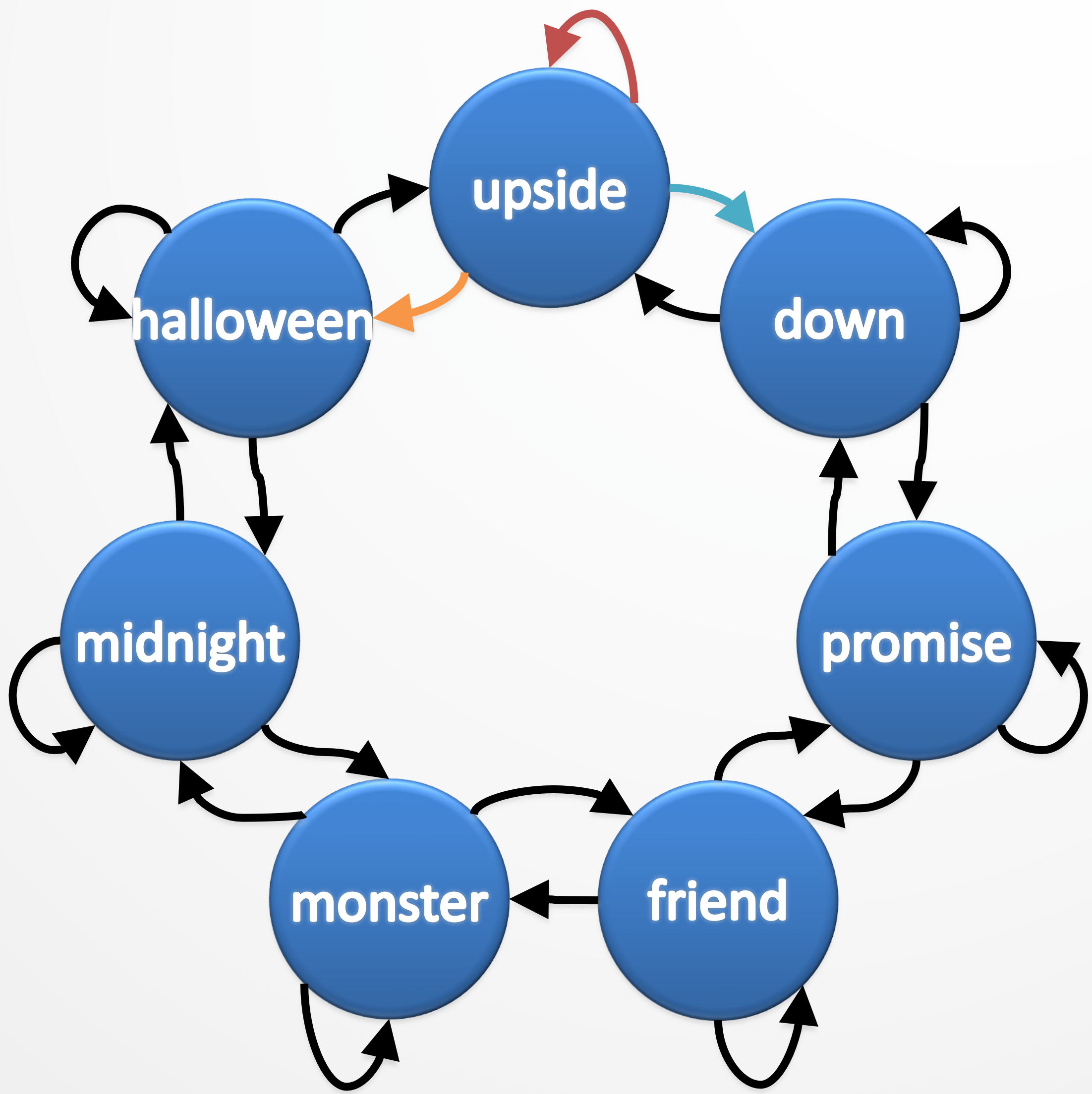
Again, a sum over each variable, but the log fraction is normalized by an assumption that they're independent ($p(x)p(y)$).

Information theory

- In general, lectures includes some walked-through examples of applying the preceding formula.
  - It's probably a good idea to walk through these examples yourself on paper.



Observable Markov model

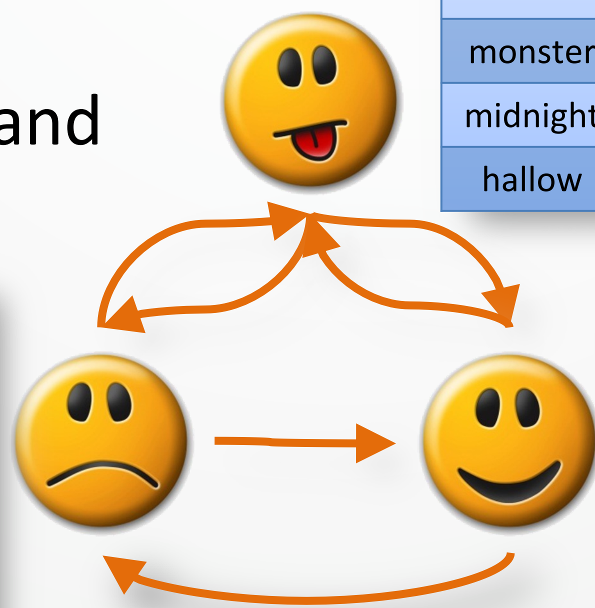


UNIVERSITY OF TORONTO

Multivariate systems

- What if a conditioning variable changes over time?
  - e.g., I'm **happy** one second and **disgusted** the next.
- Here, the **state** is the mood and the **observation** is the word.

word	P(word)
upside	0.1
down	0.05
promise	0.05
friend	0.6
monster	0.05
midnight	0.1
hallow	0.05

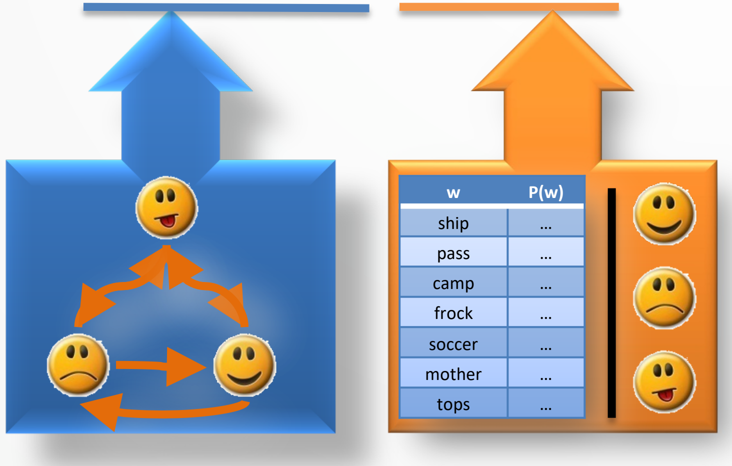


word	P(word)
upside	0.25
down	0.25
promise	0.05
friend	0.3
monster	0.05
midnight	0.09
hallow	0.01

word	P(word)
upside	0.3
down	0
promise	0
friend	0.2
monster	0.05
midnight	0.05
hallow	0.4

Observable multivariate systems

- Q: How do you **learn** these probabilities?
  - $P(w_{0:t}, q_{0:t}) \approx \prod_{i=0}^{t} \underline{P(q_i|q_{i-1})}\,\underline{P(w_i|q_i)}$

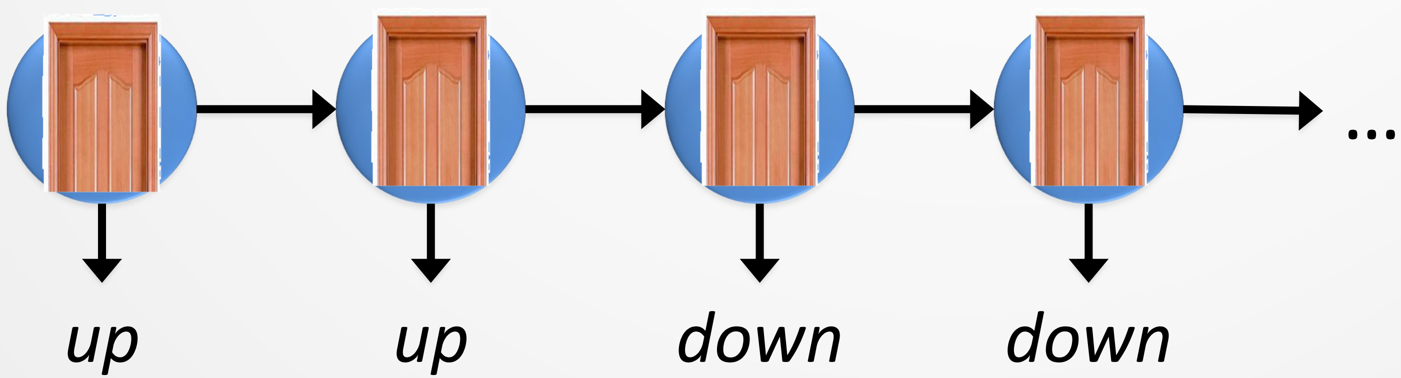


- A: Basically, the same as before.
  - $P(q_i|q_{i-1}) = \frac{P(q_{i-1}q_i)}{P(q_{i-1})}$ is learned with MLE from training data.
  - $P(w_i|q_i) = \frac{P(w_i,q_i)}{P(q_i)}$ is also learned with MLE from training data.

Hidden variables

- Q: What if you **don't** have access to the **state** during testing?
  - e.g., you're asked to compute $P(\langle up, up \rangle)$

- Q: What if you **don't** have access to the **state** during *training*?



UNIVERSITY OF TORONTO

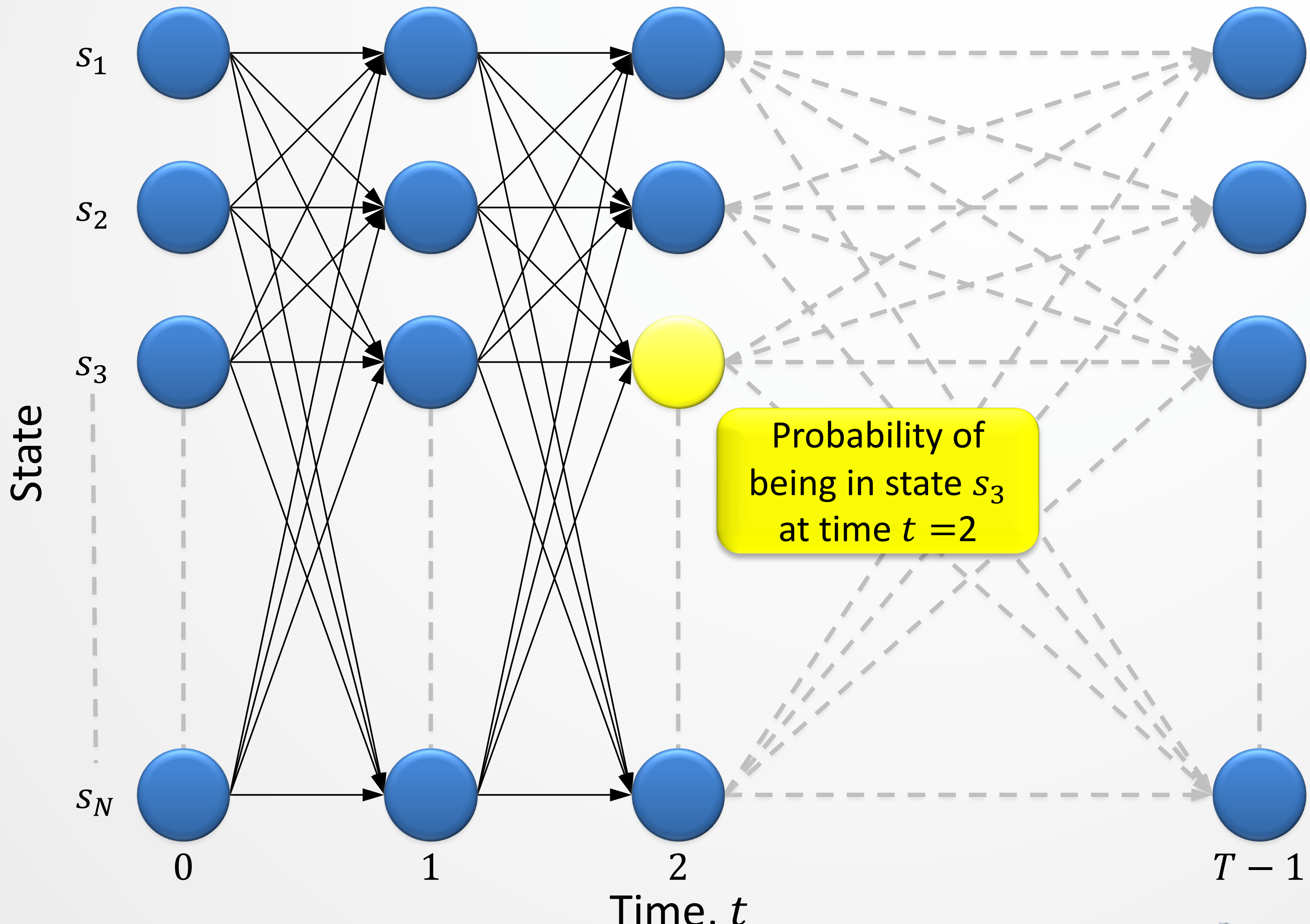
Tasks for HMMs

1. Given a **model** with particular parameters $\theta = \langle \Pi, A, B \rangle$, how do we efficiently compute the likelihood of a *particular* **observation sequence**, $P(\mathcal{O}; \theta)$?

2. Given an **observation sequence** $\mathcal{O}$ and a **model** θ, how do we choose a state sequence $Q = \{q_0, \ldots, q_T\}$ that best explains the observations?

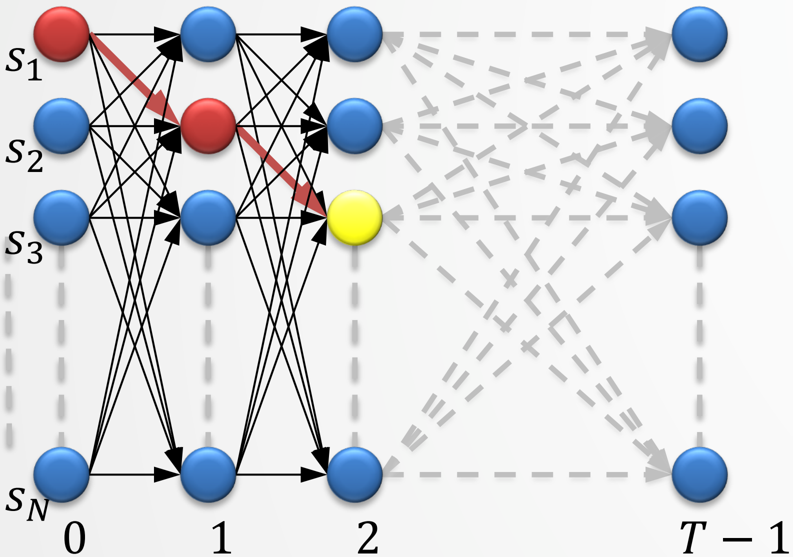
3. Given a large **observation sequence** $\mathcal{O}$, how do we choose the best parameters $\theta = \langle \Pi, A, B \rangle$ that explain the data $\mathcal{O}$?

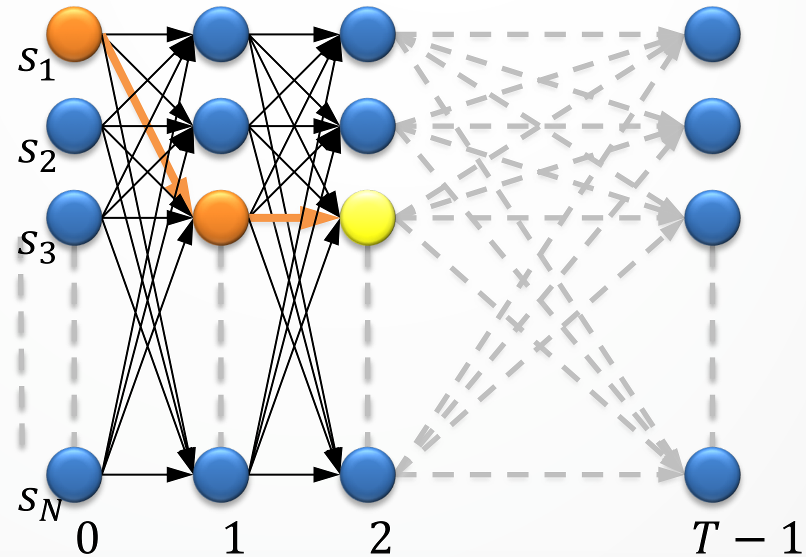
1. Trellis

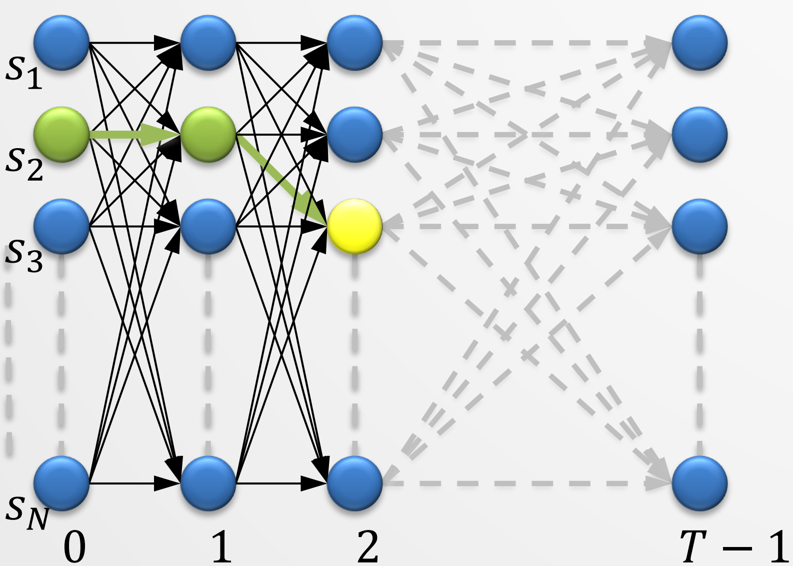


s_1
s_2
s_3
s_N
State
Probability of being in state s_3 at time t =2
0
1
2
$T-1$
Time, t

2. Choosing the best state sequence







I want to guess which sequence of states generated an observation.

E.g., if states are **PoS** and observations are **words**

2. The Viterbi algorithm

- Also an inductive dynamic-programming algorithm that uses the trellis.

- Define the probability of the most probable path leading to the trellis node at $(\text{state } i, \text{time } t)$ as

$$\boldsymbol{\delta_i(t)} = \max_{q_0 \dots q_{t-1}} P(q_0 \dots q_{t-1}, \sigma_0 \dots \sigma_{t-1}, q_t = s_i; \theta)$$

- And the incoming arc that led to this most probable path is defined as $\boldsymbol{\psi_i(t)}$

3. Training HMMs

- We want to **modify** the parameters of our model $\theta = \langle \Pi, A, B \rangle$ so that $P(\mathcal{O};\theta)$ is maximized for some **training** data $\mathcal{O}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(\mathcal{O};\theta)$$

- If we want to choose a **best state sequence** Q^* on previously unseen **test data**, the parameters of the HMM should first be tuned to similar **training data.**

3. Expectation-maximization

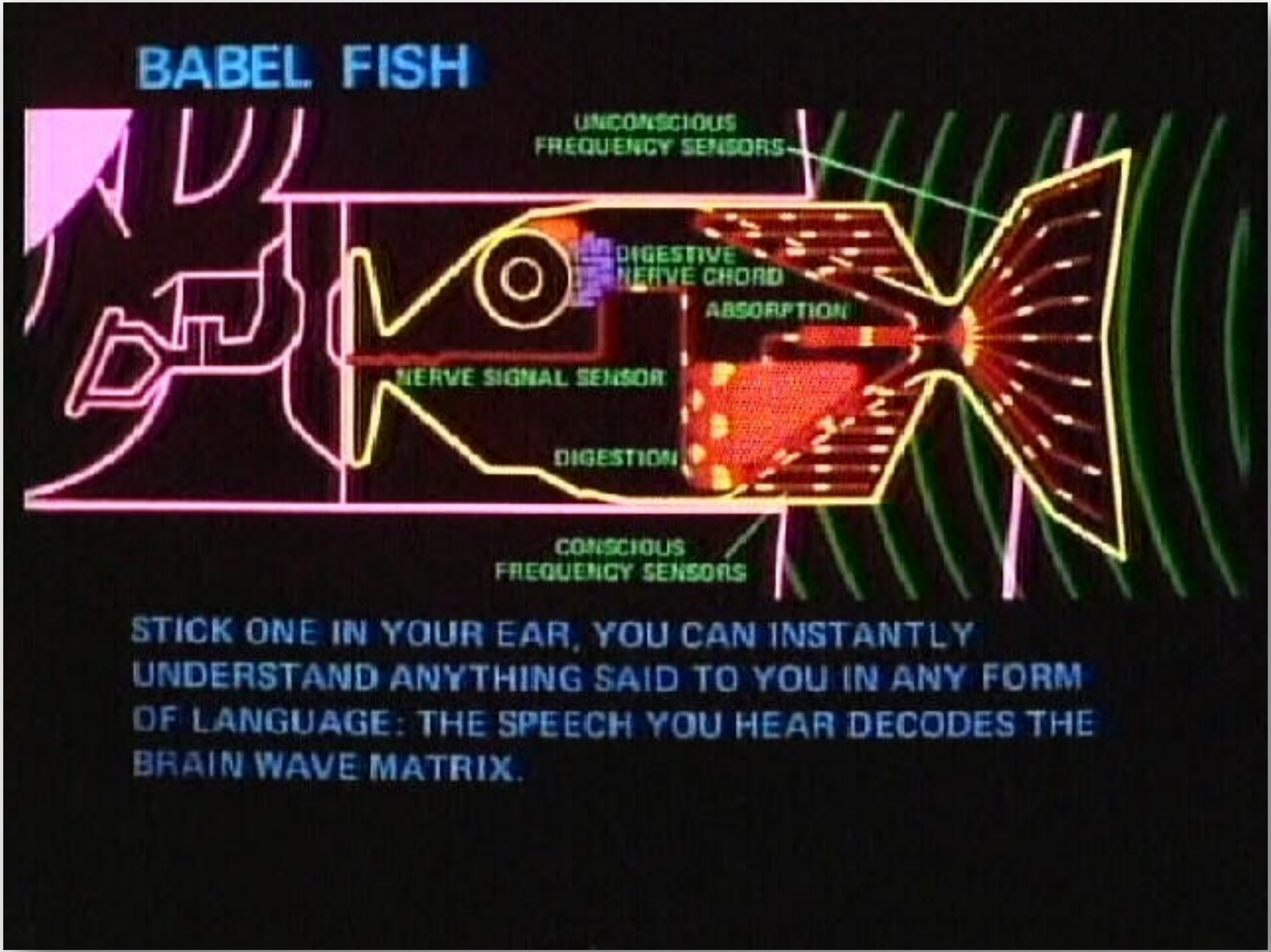
- If we knew θ, we could estimate **expectations** such as
  - Expected number of times in state s_i,
  - Expected number of transitions $s_i \rightarrow s_j$

- If we knew:
  - Expected number of times in state s_i,
  - Expected number of transitions $s_i \rightarrow s_j$

  then we could compute the **maximum likelihood estimate** of

$$\theta = \langle \pi_i, \{a_{ij}\}, \{b_i(w)\} \rangle$$

Statistical machine translation



Challenges of SMT

- Lexical ambiguity (e.g., words are polysemous).
- Differing word orders.
- Syntactic ambiguity.
- Miscellaneous idiosyncracies.

Bilingual evaluation: BLEU

- In lecture, $\|\text{Ref1}\| = 16$, $\|\text{Ref2}\| = 17$, $\|\text{Ref3}\| = 16$, and $\|\text{Cn1}\| = 18$ and $\|\text{Cn2}\| = 14$,

$$brevity_1 = \frac{17}{18} \qquad BP_1 = 1$$

$$brevity_2 = \frac{16}{14} \qquad BP_2 = e^{1-\left(\frac{8}{7}\right)} = 0.8669$$

- **Final score** of candidate C:

$$\boxed{BLEU = BP \times (p_1 p_2 \ldots p_n)^{1/n}}$$

where

$$p_n = \frac{\sum_{ngram \in C} Count_R(ngram)}{\sum_{ngram \in C} Count_C(ngram)}$$

Reference

Candidate

BLEU example

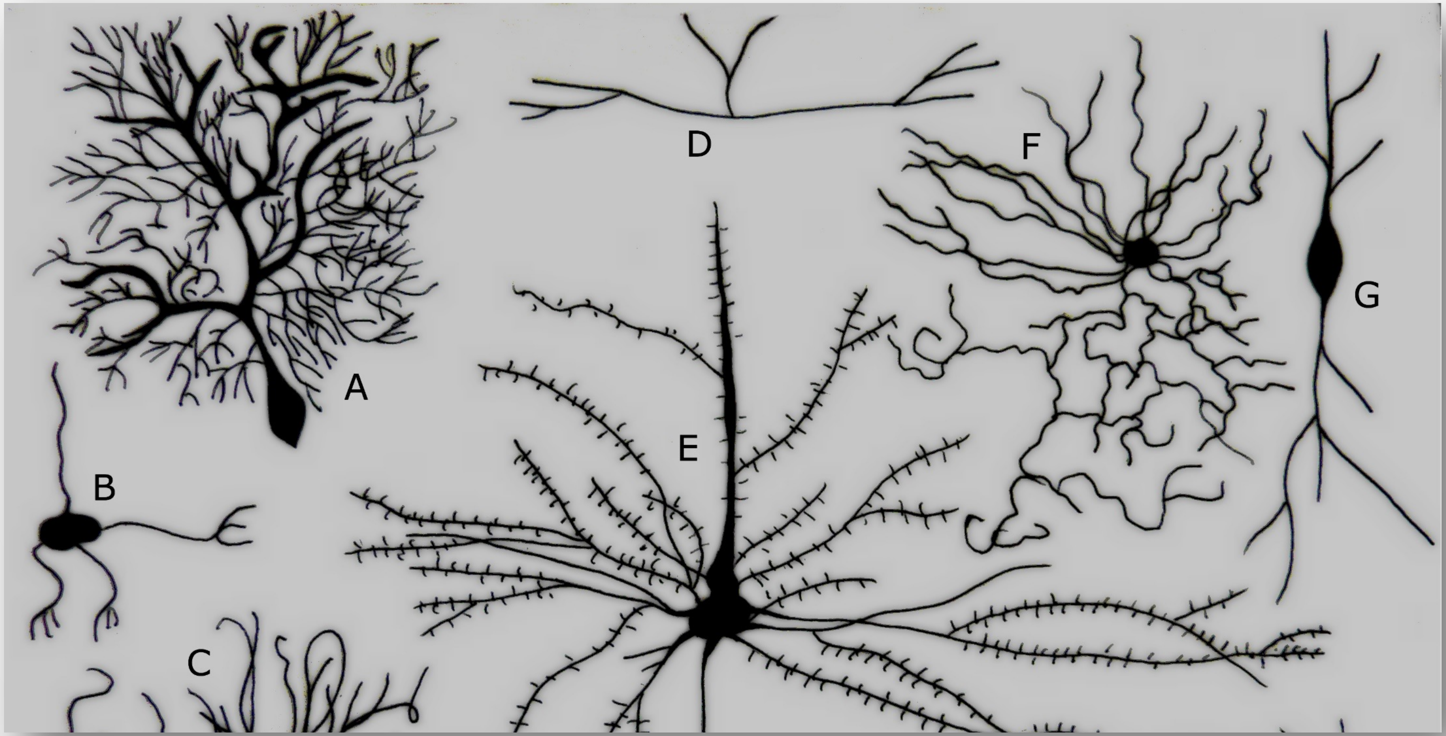
- **Reference 1:** *I am afraid Dave*
  Reference 2: *I am scared Dave*
  Reference 3: *I have fear David*
  Candidate: *I fear David*

Assume $cap(n) = 2$ for all n-grams

- $brevity = \frac{4}{3} \geq 1$ so $BP = e^{1-\left(\frac{4}{3}\right)}$

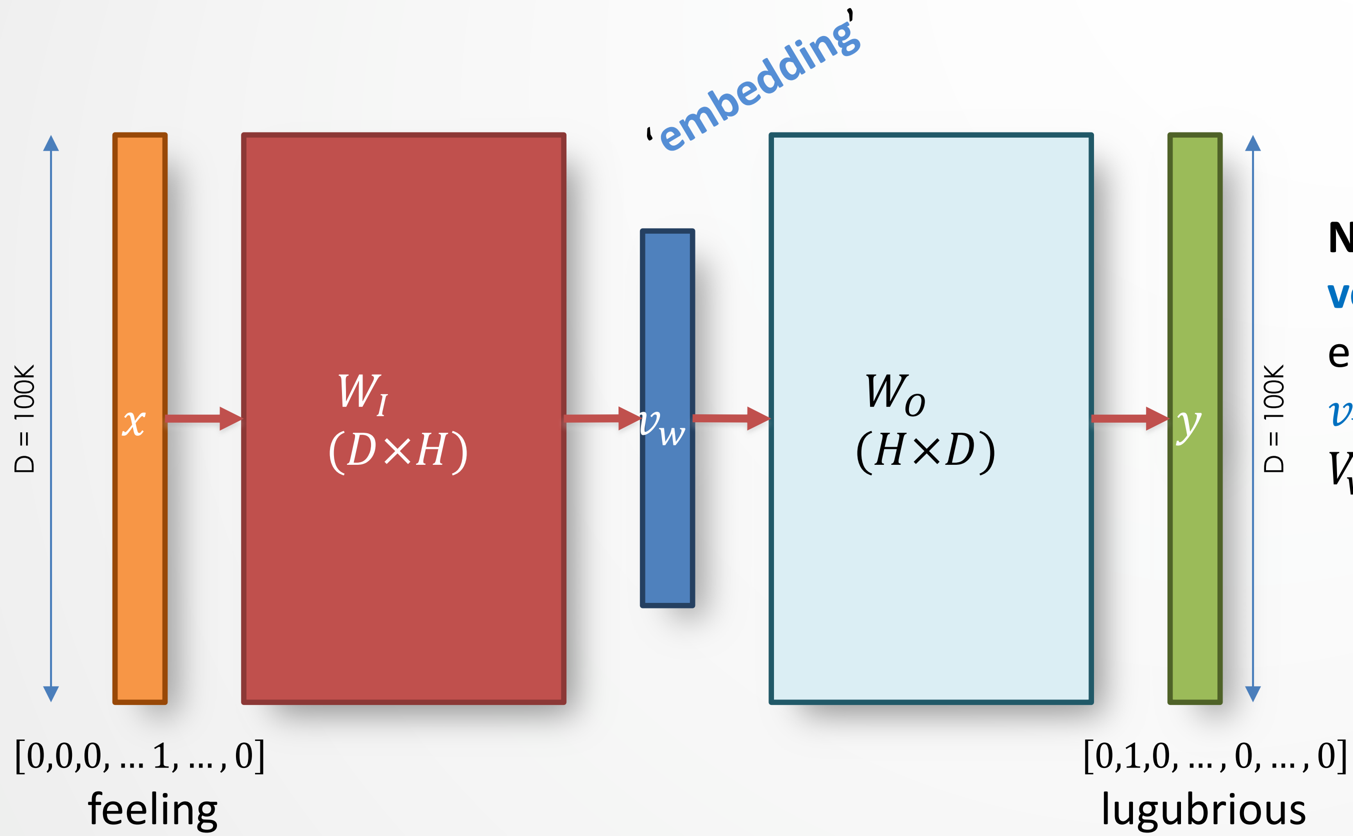
- $p_1 = \frac{\sum_{1gram \in C} Count_R(1gram)}{\sum_{1gram \in C} Count_C(1gram)} = \frac{1+1+1}{1+1+1} = 1$
- $p_2 = \frac{\sum_{2gram \in C} Count_R(2gram)}{\sum_{2gram \in C} Count_C(2gram)} = \frac{1}{2}$
- $BLEU = BP(p_1 p_2)^{\frac{1}{2}} = e^{1-\left(\frac{4}{3}\right)} \left(\frac{1}{2}\right)^{\frac{1}{2}} \approx 0.5067$

Neural language models



UNIVERSITY OF TORONTO

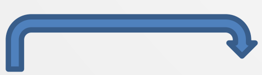
Continuous bag of words (1 word context)



Note: we have *two* **vector representations** of each word:

$v_w = x^\intercal W_I$ (w^{th} row of W_I)

$V_w = W_O^\intercal y$ (w^{th} col of W_O)



feeling	lugubrious	all
a	lugubrious	sadness
	…	

'**softmax**': $P(w_o|w_i) = \frac{\exp(V_{w_o}^\intercal v_{w_i})}{\sum_{w=1}^{W} \exp(V_w^\intercal v_{w_i})}$

Where

v_w is the 'input' vector for word w,

V_w is the 'output' vector for word w,

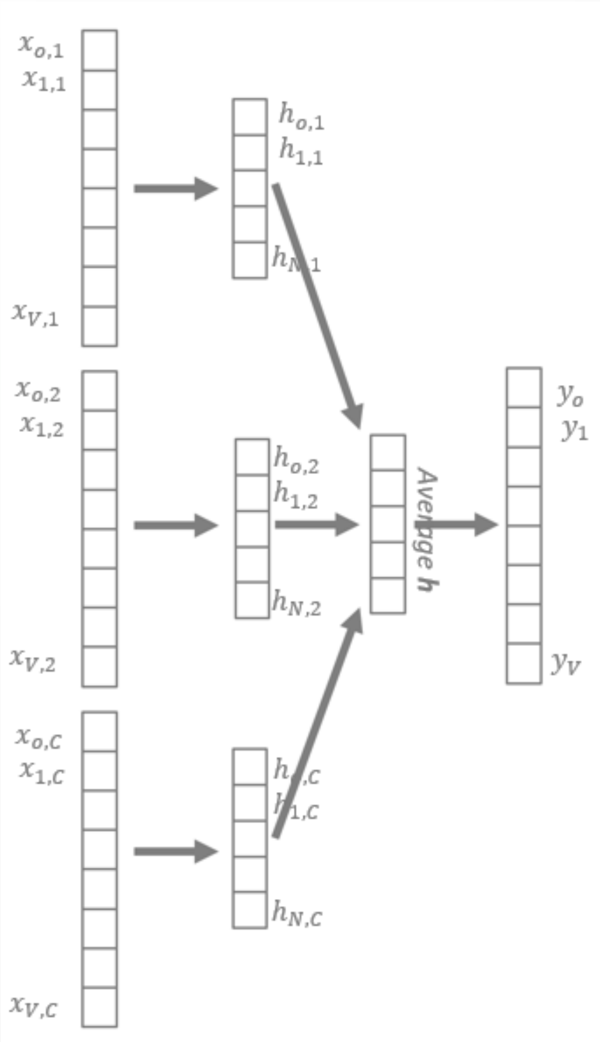
Continuous bag of words (C words context)

- If we want to use **more context**, C, we need to change the network architecture somewhat.
  - Each input word will produce one of C embeddings
  - We just need to add an **intermediate layer**, usually this just averages the embeddings.

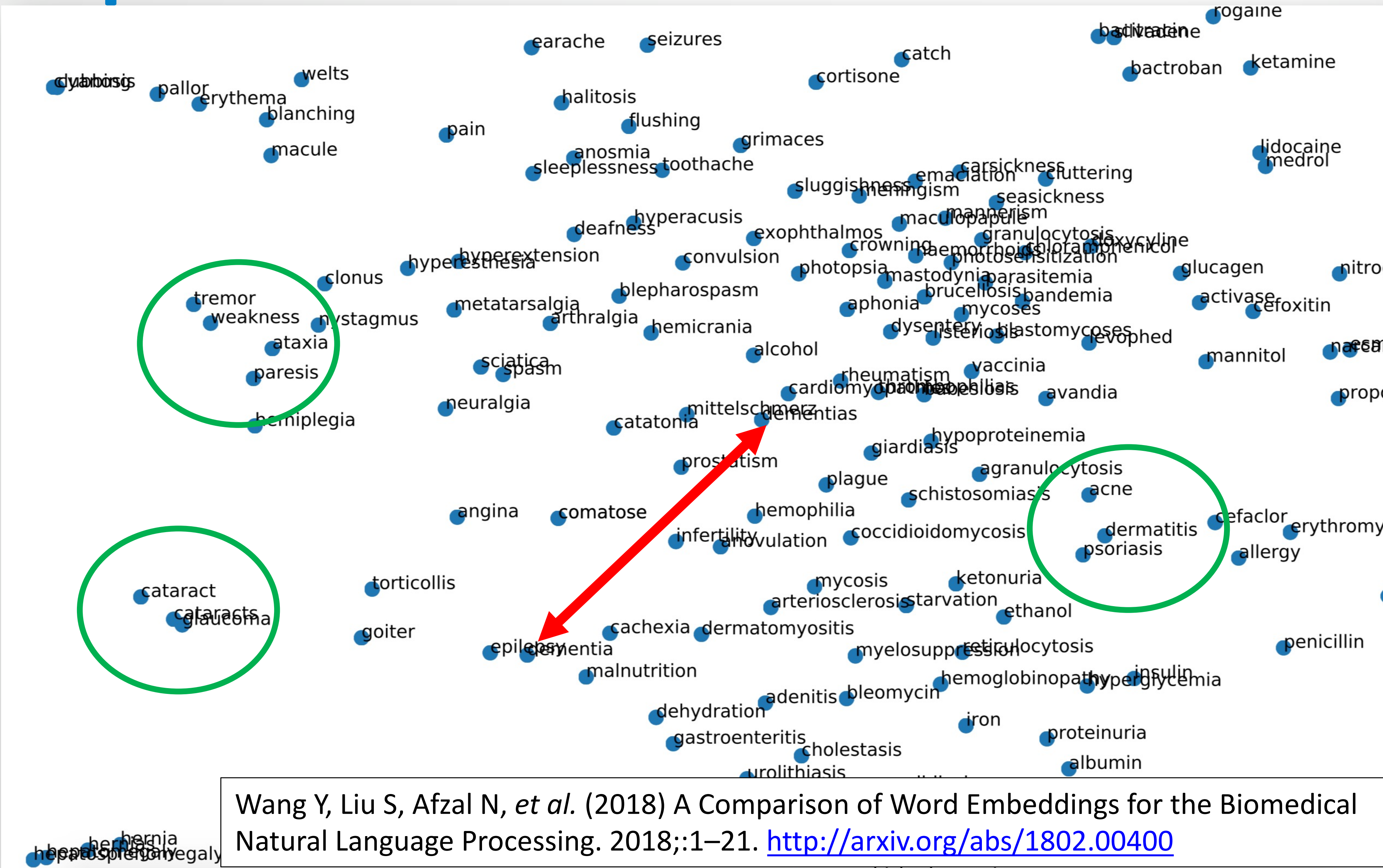
been feeling lugubrious all

felt a lugubrious sadness

…



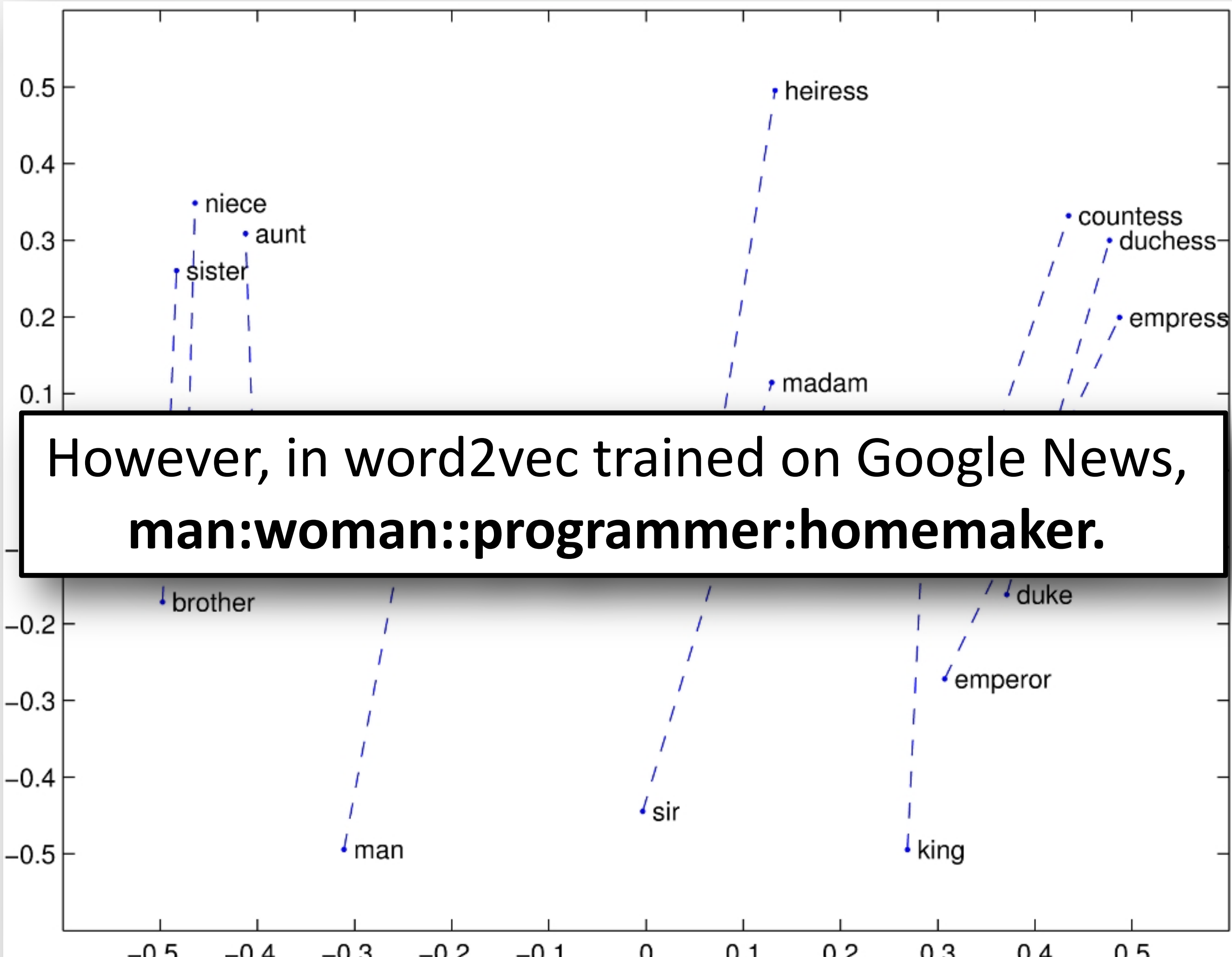
Importance of in-domain data



Wang Y, Liu S, Afzal N, *et al.* (2018) A Comparison of Word Embeddings for the Biomedical Natural Language Processing. 2018;:1–21. http://arxiv.org/abs/1802.00400

UNIVERSITY OF TORONTO

Let's talk about gender at the UofT

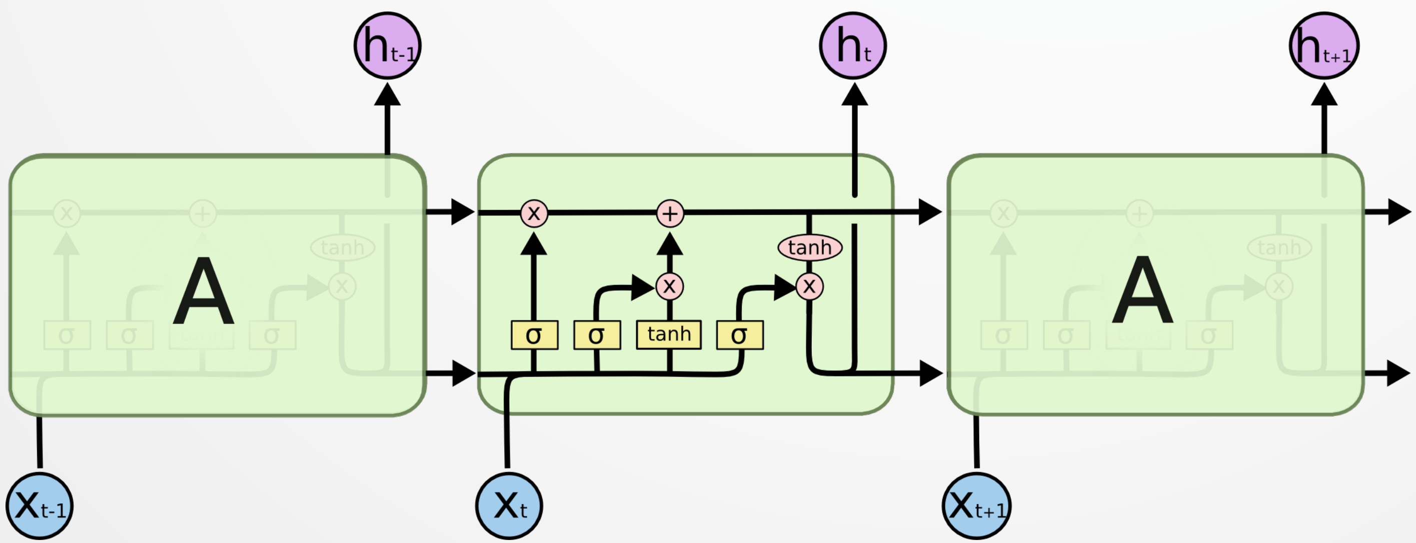


Bolukbasi T, Chang K, Zou J, *et al.* Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In: *NIPS*. 2016. 1–9.

UNIVERSITY OF TORONTO

Recurrent neural networks

- Consider RNNs generally, and LSTMs and others, specifically
- ***Hint***: How do these models differ and how they are similar? What are their **strengths** and **weaknesses**?
- What are the components of an LSTM network?



UNIVERSITY OF TORONTO

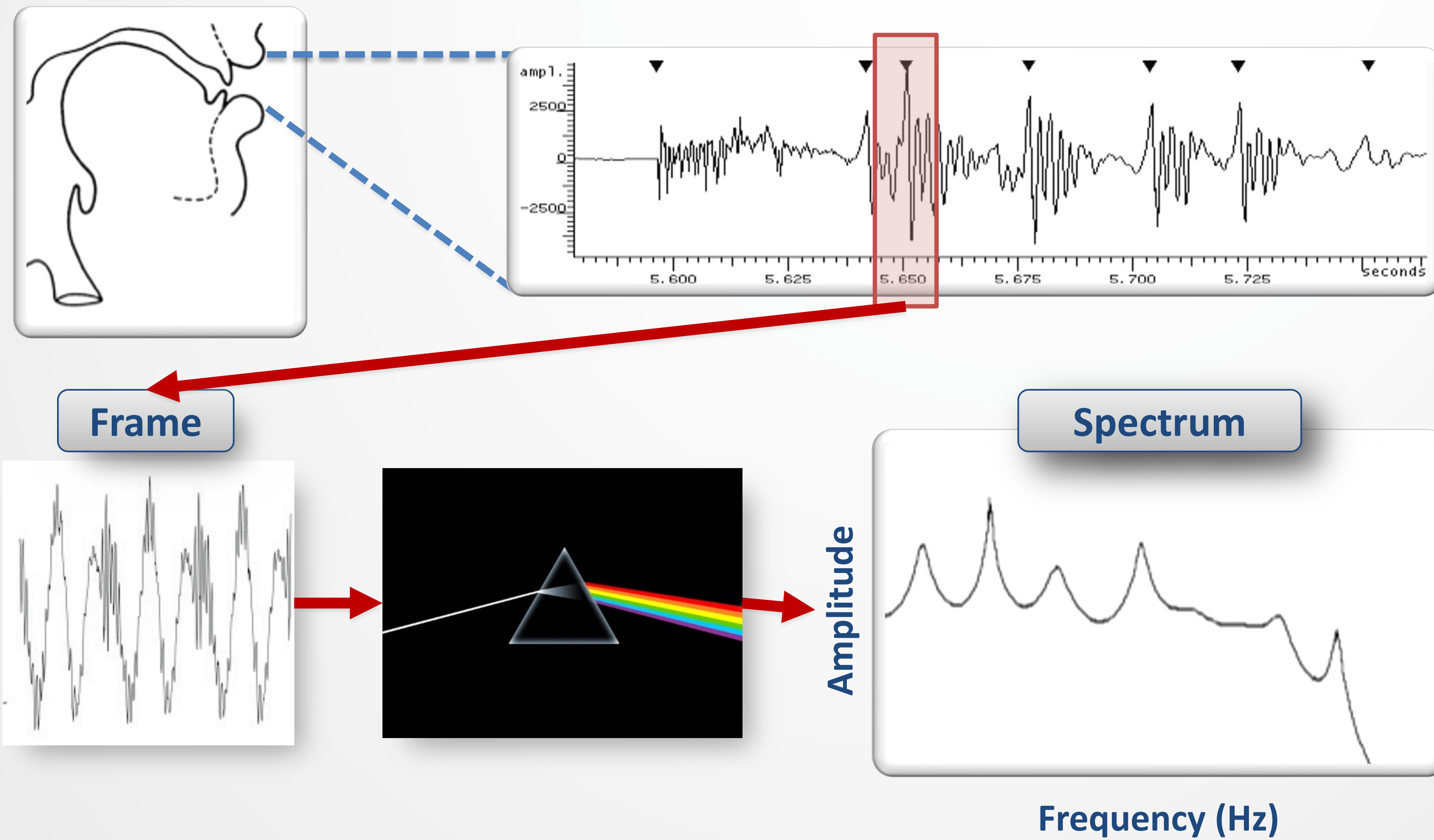
Automatic speech recognition

Manners of articulation

- **Phoneme**: *n.* a distinctive unit of speech sound.

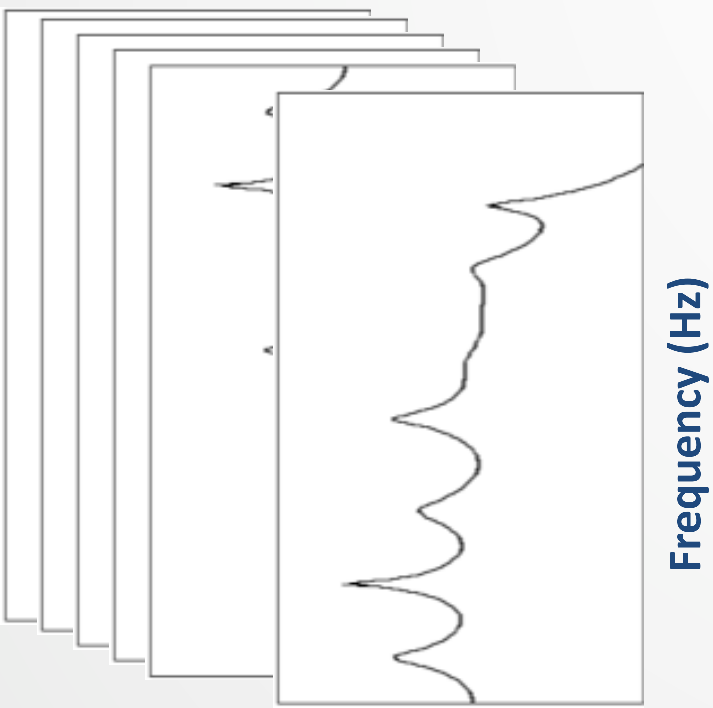
- English phonemes can be partitioned into groups, e.g.,:
  - **Stops/plosives:** complete vocal tract constriction and burst of energy (e.g., '*papa*').
  - **Fricatives:** noisy, with air passing through a tight constriction (e.g., '*shift*').
  - **Nasals**: involve air passing through the nasal cavity (e.g., '*mama*').
  - **Vowels**: open vocal tract, no nasal air.
  - **Glides/liquids**: similar to vowels, but typically with more constriction (e.g., '*wall*').

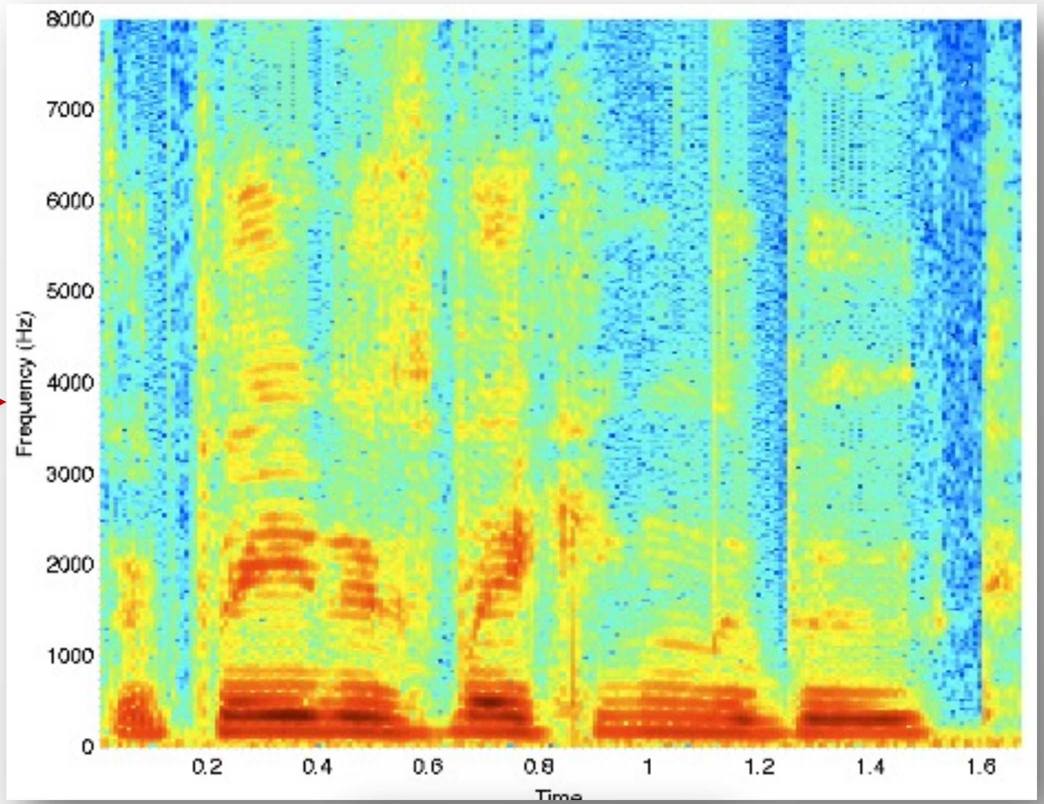
Windowing and spectra



Spectrograms

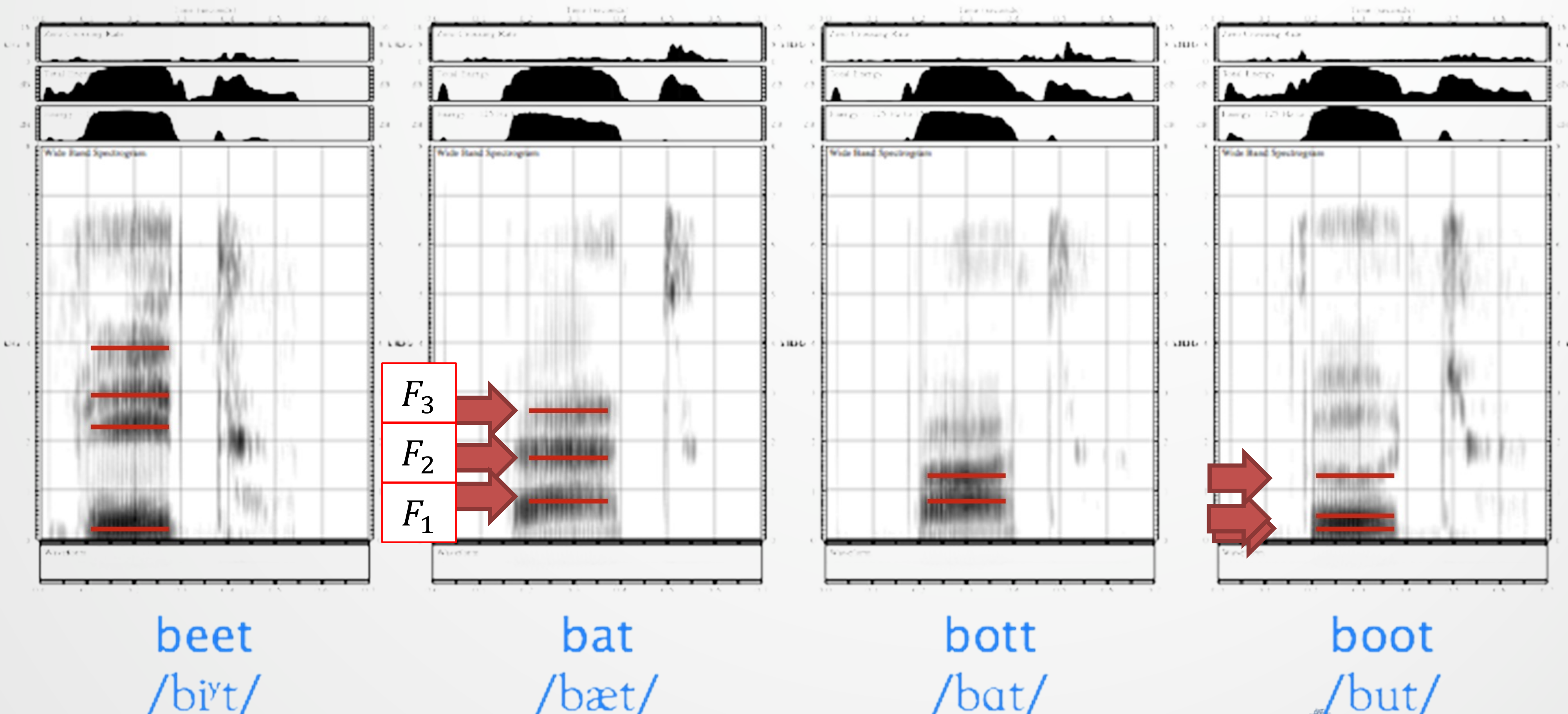
- **Spectrogram**: *n.* a 3D plot of amplitude and frequency over time.



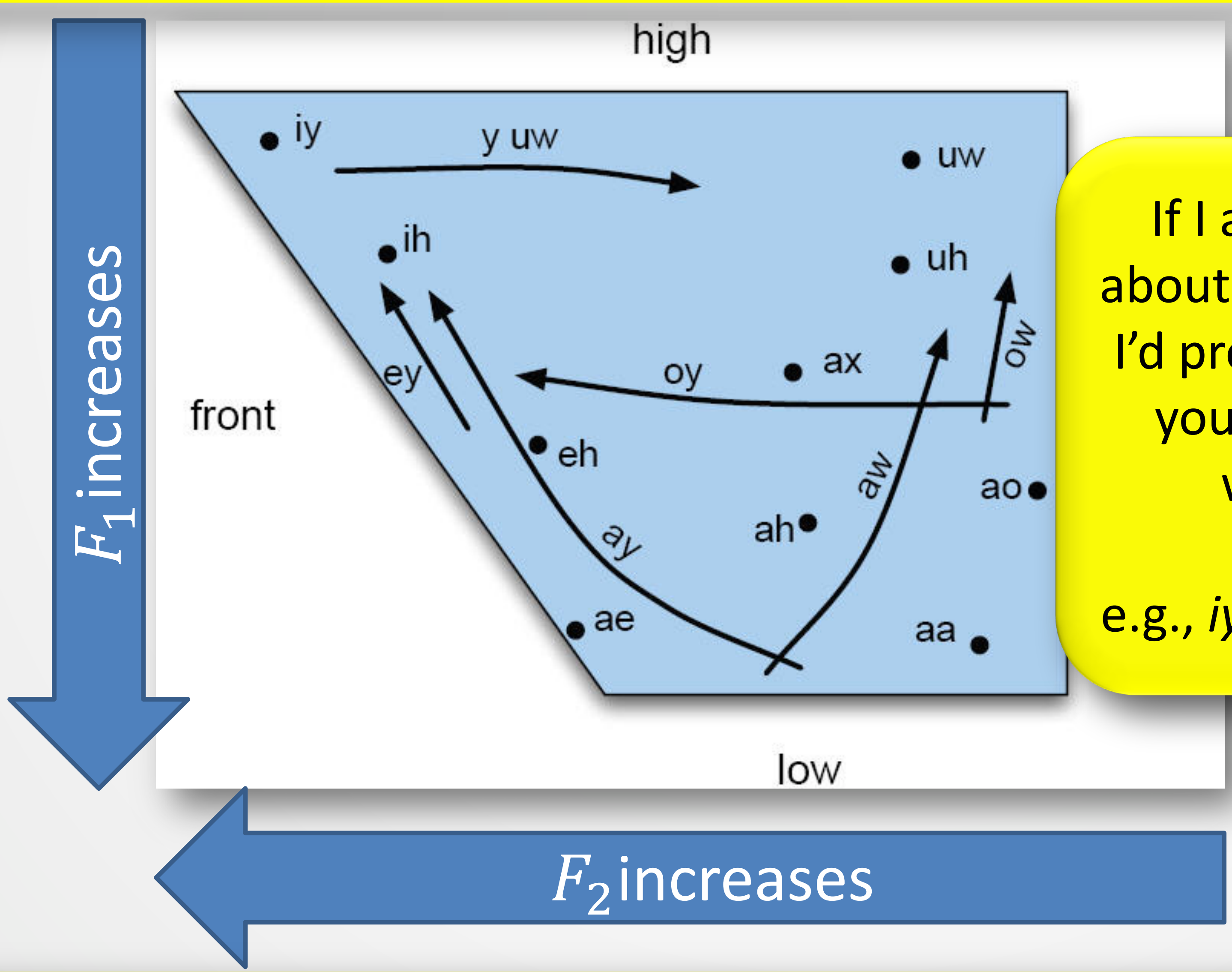


Formants and phonemes

- **Formant**: *n.* A large concentration of energy within a band of frequency (e.g., F_1, F_2, F_3).



The vowel trapezoid



If I asked you about phonemes, I'd probably give you example words.

e.g., *iy* as in *sh<u>ee</u>t*

UNIVERSITY OF TORONTO

Prosody

- **Sonorant**: *n.* Any **sustained** phoneme in which the **glottis** is vibrating (i.e., the phoneme is '**voiced**').
  - Includes some consonants (e.g., */w/*, */m/*, */g/*).

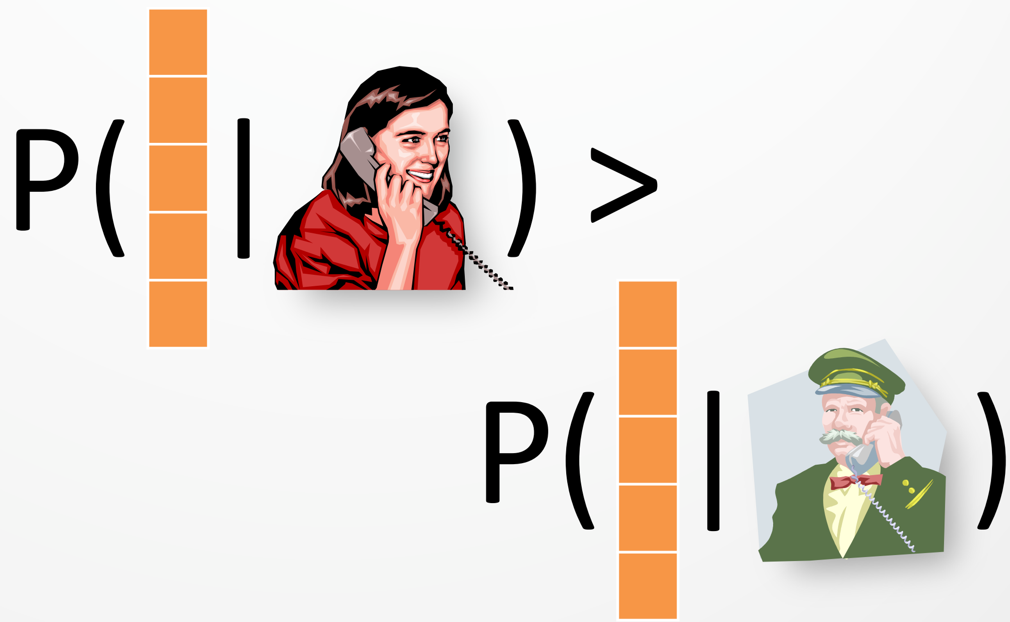
- **Prosody**: *n.* the **modification** of speech acoustics in order to convey some **extra-lexical** meaning:
  - **Pitch**: Changing of F_0 over time.
  - **Duration**: The length in time of sonorants.
  - **Loudness**: The amount of **energy** produced by the **lungs**.

Classifying speakers

- The speech produced by **one speaker** will cluster *differently* in MFCC space than speech from **another speaker**.
  - We can ∴ decide if a given observation comes from one speaker or another.

		Time, *t*			
		0	1	…	T
MFCC	1			…	
	2			…	
	3			…	
	…	…	…	…	…
	42			…	

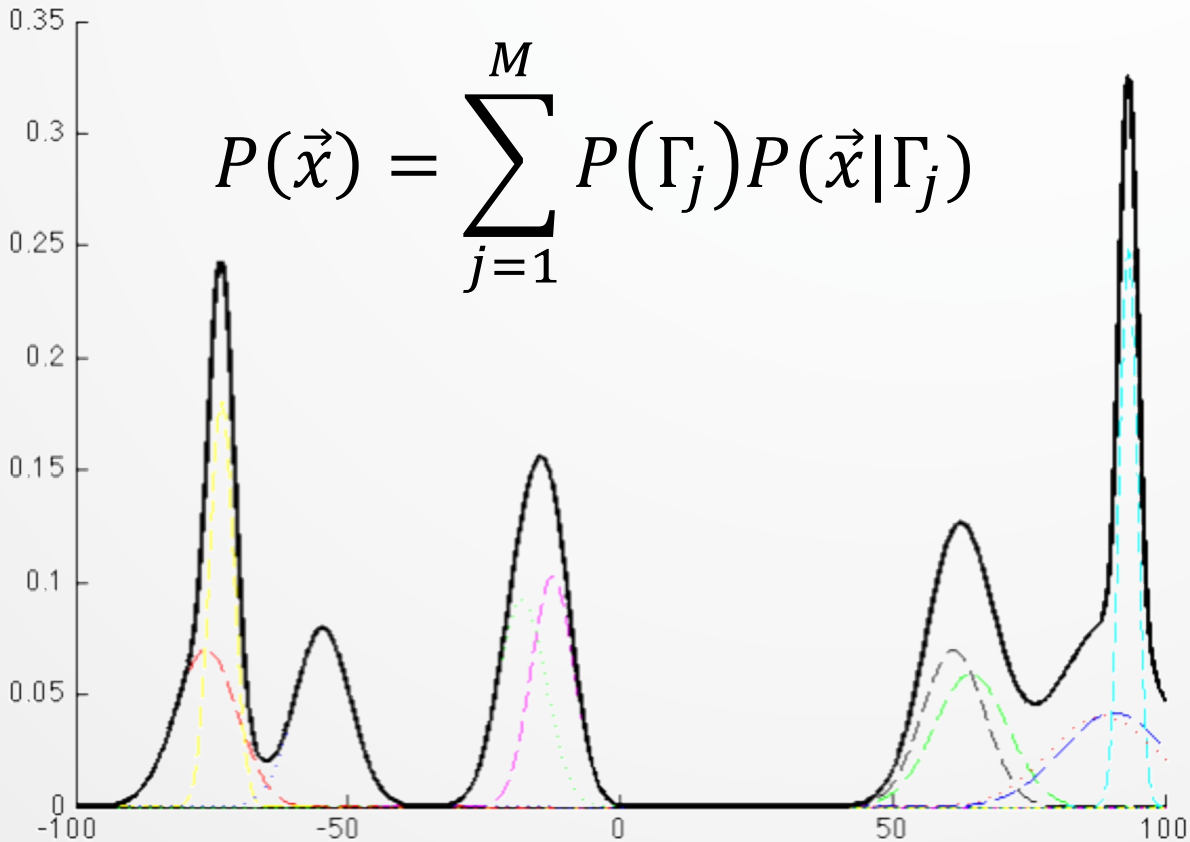
Observation matrix



Mixtures of Gaussians

- **Gaussian mixture models (GMMs)** are a weighted linear combination of M component Gaussians, $\langle \Gamma_1, \Gamma_2, \ldots, \Gamma_M \rangle$ such that

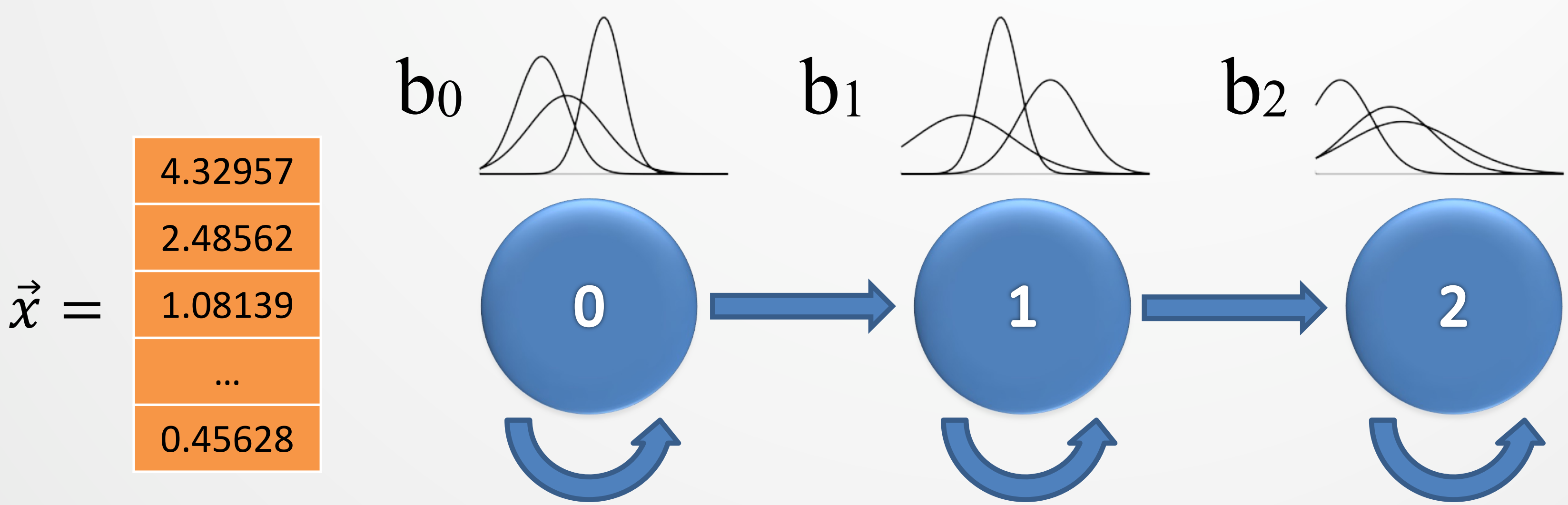
$$P(\vec{x}) = \sum_{j=1}^{M} P(\Gamma_j) P(\vec{x} \mid \Gamma_j)$$



UNIVERSITY OF TORONTO

Continuous HMMs

- Previously we saw **discrete HMMs**: at each state we observed a discrete symbol from a finite set of discrete symbols.
- A **continuous HMM** has observations that are distributed over continuous variables.
  - Observation probabilities, b_i, are also continuous.



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**

- See the example in lecture. **Work it out yourself.**

UNIVERSITY OF TORONTO

Speech synthesis

UNIVERSITY OF TORONTO

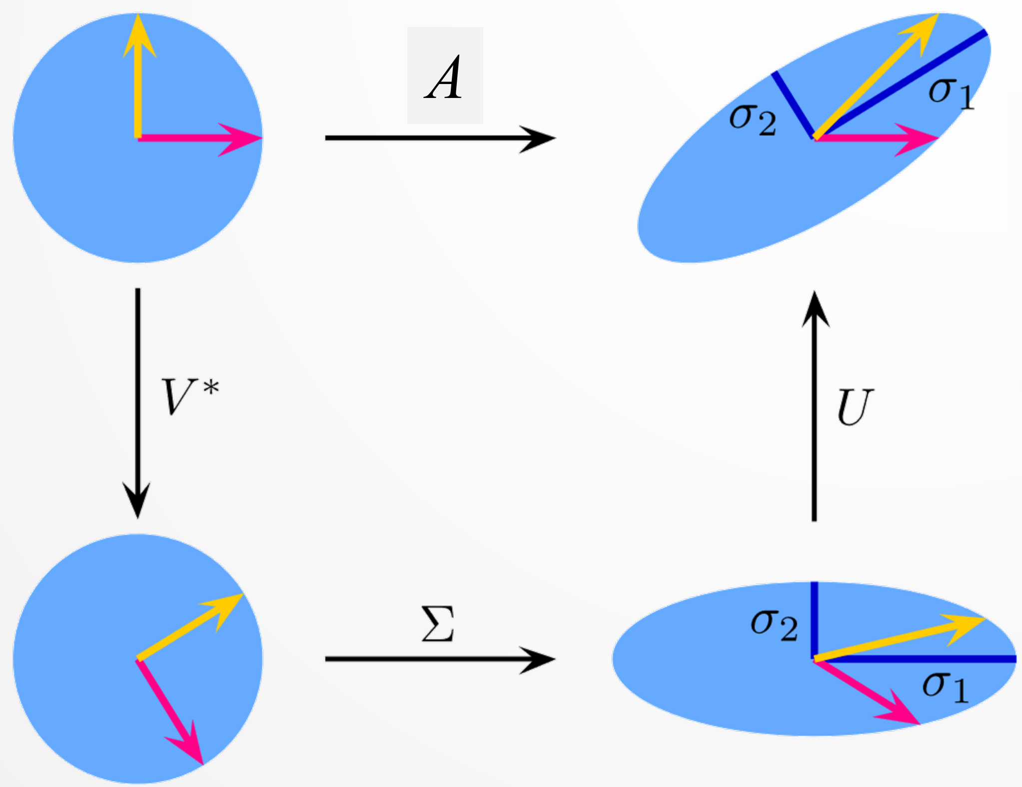
Speech synthesis

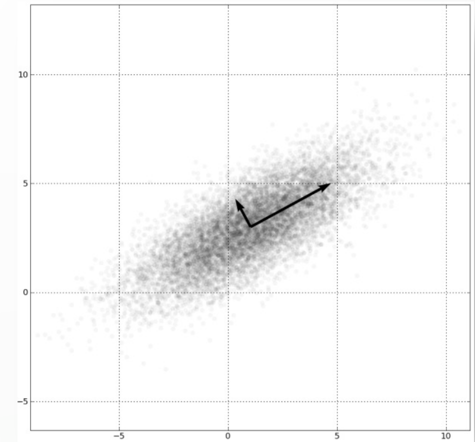
- **Text-to-speech**: *n.* the conversion of electronic text into equivalent, audible speech waveforms.

- Three **architectures** for performing speech synthesis:
  - Formant synthesis,
  - Concatenative synthesis,
  - Articulatory synthesis.
- How do they differ? What are their (dis)advantages?

- Common **components** of speech synthesis:
  - **Letter-to-sound rules** and dictionaries,
  - Acoustic prosody modification.

Singular value decomposition (SVD)





$$A = U \cdot \Sigma \cdot V^*$$

SVD example

$$A_{t\times d} = T_{t\times n} S_{n\times n} (D_{d\times n})^{\intercal}$$

$A =$

	d_1	d_2	d_3	d_4	d_5	d_6
natural	1	0	1	0	0	0
language	0	1	0	0	0	0
processing	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

$T =$

nat.	-0.44	-0.30	0.57	0.58	0.25
lang.	-0.13	-0.33	-0.59	0	0.73
proc.	-0.48	-0.51	-0.37	0	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-0.26	0.65	-0.41	0.58	-0.09

$S =$

2.16	0	0	0	0
0	1.59	0	0	0
0	0	1.28	0	0
0	0	0	1	0
0	0	0	0	0.39

$D^{\intercal} =$

d_1	d_2	d_3	d_4	d_5	d_6
-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
-0.29	-0.53	-0.19	0.63	0.22	0.41
0.28	-0.75	0.45	-0.20	0.12	-0.33
0	0	0.58	0	-0.58	0.58
-0.53	0.29	0.63	0.19	0.41	-0.22

- What do these matrices mean?

SVD example

- Matrices T and D represent **terms** and **documents**, respectively in this ***new*** space.
  - E.g., the first row of T corresponds to the first row of A, and so on.

$T =$

nat..	-0.44	-0.30	0.57	0.58	0.25
lang.	-0.13	-0.33	-0.59	0	0.73
proc.	-0.48	-0.51	-0.37	0	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-0.26	0.65	-0.41	0.58	-0.09

$D^\intercal =$

d_1	d_2	d_3	d_4	d_5	d_6
-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
-0.29	-0.53	-0.19	0.63	0.22	0.41
0.28	-0.75	0.45	-0.20	0.12	-0.33
0	0	0.58	0	-0.58	0.58
-0.53	0.29	0.63	0.19	0.41	-0.22

- T and D are **orthonormal**, so all columns are orthogonal to each other and $T^\intercal T = D^\intercal D = I$.

UNIVERSITY OF TORONTO

SVD example

- By restricting T, S, and D to their first $k < n$ columns, their product gives us $\hat{A}$, a 'best least squares' approximation of A.

$T =$

cosm.	-0.44	-0.30	0.57	0.58	0.25
astro.	-0.13	-0.33	-0.59	0	0.73
moon	-0.48	-0.51	-0.37	0	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-0.26	0.65	-0.41	0.58	-0.09

$S =$

2.16	0	0	0	0
0	1.59	0	0	0
0	0	1.28	0	0
0	0	0	1	0
0	0	0	0	0.39

$D^{\intercal} =$

d_1	d_2	d_3	d_4	d_5	d_6
-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
-0.29	-0.53	-0.19	0.63	0.22	0.41
0.28	-0.75	0.45	-0.20	0.12	-0.33
0	0	0.58	0	-0.58	0.58
-0.53	0.29	0.63	0.19	0.41	-0.22

Final thoughts

(not thoughts on the final)

NLC in industry

👍

Final thoughts

- This course **barely** scratches the surface of these beautiful topics. Talk to these people:

Graeme Hirst

Gerald Penn

Frank Rudzicz

Suzanne Stevenson

Yang Xu

- Many of the techniques in this course are applicable **generally**.
- Now is a great time to make fundamental **progress** in this and adjacent areas of research.

Aside – Knowledge

- **Anecdotes** are often useless except as proofs by contradiction.
  - E.g., *"I saw Google used as a verb"* does **not** mean that *Google* is **always** (or even **likely** to be) a verb, just that it is **not always** a noun.

- **Shallow statistics** are often not enough to be truly meaningful.
  - E.g., *"My ASR system is 95% accurate on my test data. Yours is only 94.5% accurate, you horrible knuckle-dragging idiot."*
    - What if the test data was **biased** to favor my system?
    - What if we only used a **very small** amount of data?

- We need a **test** to see if our statistics actually **mean** something.

Find some way to be *comfortable* making *mistakes*


Thank you