This lecture

• An extractive summary of the course.

• Reminder: open office hours 25 April 14h-16h30 (location TBD)
Exam

• 30 April from 19h00—22h00.

<table>
<thead>
<tr>
<th>A-Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN 3</td>
<td>BN 3</td>
</tr>
<tr>
<td>Clara Benson</td>
<td>Clara Benson</td>
</tr>
<tr>
<td>320 Huron Street</td>
<td>320 Huron Street</td>
</tr>
</tbody>
</table>

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

*May be subject to change*
Structure

• Following the format of previous years:
  • 20 **multiple-choice** questions [40 marks]
    • 4 options each.
  • 10 **short-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 quiz 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.
2014 Final exam distribution
2015 Final exam distribution
Course outline (approximate)

• Introduction and linguistic data (2 lectures)
• $N$-gram models and features of data (2 lectures) *
• Entropy and information theory (2 lectures) *
• Hidden Markov models (3 lectures) *
• Statistical machine translation (3 lectures) **
• Neural language models (2 lectures) *
• Articulatory and acoustic phonetics (2 lectures) *
• Automatic speech recognition (2 lectures) **
• Speech synthesis (1 lecture) **
• Information retrieval (2 lectures) **
• Text summarization (1 lecture) **

* techniques  ** applications
Categories of linguistic knowledge

- **Phonology**: the study of patterns of speech sounds.
  e.g., “read” $\rightarrow /r\ iy\ d/$

- **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., $\text{NounPhrase} \rightarrow \text{det. adj. n.}$

- **Semantics**: the study of how meaning is created by words and phrases.
  e.g., “book” $\rightarrow$

- **Pragmatics**: the study of meaning in broad contexts.
NLC as Artificial Intelligence

• NLC involves **resolving ambiguity** at all levels.
  • **Reasoning with world knowledge.**
    • In the early days knowledge was **explicitly encoded** in artificial symbolic systems (e.g., context-free grammars) by **experts**.

• Now, algorithms learn using **probabilities** to distinguish subtly different competing hypotheses.
  • E.g., is Google a **noun** or a **verb**?
  • An example where Google ∈ Nouns (**Google makes Android**), does **not** mean that Google is never a verb (**Go Google yourself**).

• \( P(\text{Google} \in \text{Nouns}) > P(\text{Google} \in \text{Verbs}) > 0 \)
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).
Notable corpora


• **Penn treebank**: Syntactically annotated Brown, plus others incl. 1989 *Wall Street Journal*.

• **Switchboard corpus**: 120 hours ≈ 2.4M tokens. 2.4K telephone conversations between US English speakers.

• **Hansard corpus**: Canadian parliamentary proceedings, French/English bilingual.
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., \( P(\text{food} \mid \text{I want Chinese} ) = \frac{P(\text{I want Chinese food})}{P(\text{I want Chinese})} \approx \frac{\text{Count(\text{I want Chinese food})}}{\text{Count(\text{I want Chinese})}} \)
Bayes’ theorem

\[ P(A) \]

\[ P(A,B) \]

\[ P(B) \]

Bayes theorem:

\[
P(A|B) = \frac{P(B|A) P(A)}{P(B)}
\]

\[
P(A, B) = P(A)P(B|A)
\]

\[
P(A, B) = P(B)P(A|B)
\]
Maximum likelihood estimate

• **Maximum likelihood estimate (MLE)** of parameters \( \theta \)
in a **model** \( M \), given **training data** \( T \) is

the estimate that maximizes the likelihood of the

*training data* using the *model*.

• e.g., \( T \) is the Brown corpus,
  \( M \) is the bigram and unigram tables
  \( \theta_{(to|want)} \) is \( P(to|want) \).
Sparsity of unigrams vs. bigrams

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

<table>
<thead>
<tr>
<th>Count($w_{t-1}$,$w_t$)</th>
<th>Unigram counts:</th>
<th>$w_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I</td>
<td>want</td>
</tr>
<tr>
<td>I</td>
<td>2533</td>
<td>927</td>
</tr>
<tr>
<td>want</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>to</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>eat</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Chinese</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>food</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>lunch</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>spend</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
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
Smoothing as redistribution

- Steal from the rich and give to the poor.
- E.g., \textit{Count(I caught)}

![Bar charts showing actual counts vs. adjusted counts for different fish types: trout, salmon, sole, haddock, catfish. The adjusted counts are visually represented as a redistribution of the actual counts.](chart.png)
Add-1 smoothing (Laplace)

- Given a vocab size $||\mathcal{V}||$ and corpus size $N$, just add 1 to all the counts! No more zeros!

- MLE: $P(w_i) = \frac{C(w_i)}{N}$

- Laplace estimate: $P_{Lap}(w_i) = \frac{C(w_i)+1}{N+||\mathcal{V}||}$

- Does this give a proper probability distribution? Yes:

$$\sum_w P_{Lap}(w) = \sum_w \frac{C(w) + 1}{N + ||\mathcal{V}||} = \frac{\sum_w C(w) + \sum_w 1}{N + ||\mathcal{V}||} = \frac{N + ||\mathcal{V}||}{N + ||\mathcal{V}||} = 1$$
Add-\(\delta\) smoothing

• Laplace’s method generalizes to the add-\(\delta\) estimate:

\[
P_{\delta}(w_i) = \frac{C(w_i) + \delta}{N + \delta||V||}
\]

• Consider also:
  • Simple interpolation
  • Katz smoothing
  • Good-Turing smoothing
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*)
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.
mRMR feature selection

• Minimum-redundancy-maximum-relevance (mRMR) can use correlation, distance scores (e.g., $D_{KL}$) or mutual information to select features as in

• For feature set $S$ of features $f_i$, class $c$, $D(S, c)$ : a measure of relevance of $S$ for $c$, and $R(S)$ : a measure of the redundancy of $S$,

$$S_{mRMR} = \arg\max_S [D(S, c) - R(S)]$$
Information and entropy
Entropy

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

\[
H(S) = \sum_{i} p_i I(w_i) = \sum_{i} p_i \log_2 \frac{1}{p_i}
\]

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 \( \text{given that you know another}. \)

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

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, lecture3 includes some walked-through examples of applying the preceding formula.
• It’s probably a good idea to walk through these examples yourself on paper.
Collocations

- **Collocation**: *n.* a ‘turn-of-phrase’ or usage where a sequence of words is **perceived** to have a meaning ‘**beyond**’ the sum of its parts.

- E.g., ‘*disk drive*’, ‘*video recorder*’, and ‘*soft drink*’ are collocations. ‘*cylinder drive*’, ‘*video measurer*’, ‘*weak drink*’ are **not** despite some near-synonymy between alternatives.

- Collocations are **not** just highly frequent bigrams, otherwise ‘*of the*’, and ‘*and the*’ would be collocations.

- **How can we test if a bigram is a collocation or not?**
Decision trees

• Consists of **rules** for classifying data that consists of many **attributes/features**.

• Walk through the Simpsons example from 4.
Markov models
Observable Markov model

- Probabilities on all outgoing arcs must sum to 1.

- \( P(\text{ship}|\text{ship}) + P(\text{tops}|\text{ship}) + P(\text{pass}|\text{ship}) = 1 \)

- \( P(\text{ship}|\text{tops}) + P(\text{tops}|\text{tops}) + P(\text{mother}|\text{tops}) = 1 \)

- ...
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.

<table>
<thead>
<tr>
<th>word</th>
<th>P(word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>0.1</td>
</tr>
<tr>
<td>pass</td>
<td>0.05</td>
</tr>
<tr>
<td>camp</td>
<td>0.05</td>
</tr>
<tr>
<td>flock</td>
<td>0.6</td>
</tr>
<tr>
<td>soccer</td>
<td>0.05</td>
</tr>
<tr>
<td>mother</td>
<td>0.1</td>
</tr>
<tr>
<td>tops</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>P(word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship</td>
<td>0.3</td>
</tr>
<tr>
<td>pass</td>
<td>0.0</td>
</tr>
<tr>
<td>camp</td>
<td>0.0</td>
</tr>
<tr>
<td>flock</td>
<td>0.2</td>
</tr>
<tr>
<td>soccer</td>
<td>0.05</td>
</tr>
<tr>
<td>mother</td>
<td>0.05</td>
</tr>
<tr>
<td>tops</td>
<td>0.4</td>
</tr>
</tbody>
</table>
Observable multivariate systems

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

- 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 $P(\langle ship, ship \rangle)$

• Q: What if you don’t have access to the state during training?
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, $P(O; \theta)$?

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

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

State

$S_1$

$S_2$

$S_3$

$S_N$

Time, $t$

0

1

2

$T - 1$

Probability of being in state $S_3$ at time $t = 2$
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

\[
\delta_i(t) = \max_{q_0 \ldots q_{t-1}} P(q_0 \ldots q_{t-1}, \sigma_0 \ldots \sigma_{t-1}, q_t = s_i; \theta)
\]

- And the incoming arc that led to this most probable path is defined as \(\psi_i(t)\)
3. Training HMMs

• We want to modify the parameters of our model \( \theta = (\Pi, A, B) \) so that \( P(O; \theta) \) is maximized for some training data \( O \):

\[
\hat{\theta} = \arg\max_{\theta} P(O; \theta)
\]

• 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 $\theta$, 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

STICK ONE IN YOUR EAR, YOU CAN INSTANTLY UNDERSTAND ANYTHING SAID TO YOU IN ANY FORM OF LANGUAGE: THE SPEECH YOU HEAR DECODES THE BRAIN WAVE MATRIX.
Challenges of SMT

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

• Sentence alignment.
  • **Gale&Church**: alignment by length (minimize costs).
  • **Church**: *cognates* approximated by 4-graphs.
  • **Melamed**: *cognates* approximated by longest common subsequences.
The noisy channel

$E^* = \arg\max_E P(F|E)P(E)$
Word alignment

- **Word alignments** can be 1:1, N:1, 1:N, 0:1, 1:0,... E.g.,

  - "zero fertility" word: not translated (1:0)
  - "spurious" words: generated from 'nothing' (0:1)
  - One word translated as several words (1:N)

**Note** that this is only one possible alignment.
IBM Model 1 assumption

\[ P(\text{Canada's program has been implemented}) \]

\[ \Rightarrow \]

\[ P(\text{Canada's program has been implemented}) \]
IBM Model 1: EM

1. **Initialize** translation parameters randomly (or uniformly).

2. **Expectation**: Compute expected value of $\text{Count}(e, f)$ for all words in training data $\mathcal{O}$, given your current translation parameters, $\theta_k$.

3. **Maximization**: Compute the maximum likelihood estimate of the parameters based on the expected counts, giving improved parameters, $\theta_{k+1}$. 
IBM Model 1: EM

1. Take the **product** of each $p(e)$ with each alignments and sentence pair.

2. **Normalize** by summing over all alignments for each sentence.

3. **Add** the appropriate normalized counts for each French/English word pair to find $t_{\text{count}}$ (and total).

4. Use $t_{\text{count}}$ and total to re-estimate $p(f|e)$.

(See lecture 6-1)

\[
P(F|a,E) = P(\text{maison}|\text{blue}) \times P(\text{bleue}|\text{house}) = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}
\]

\[
P(F|a,E) = P(\text{la}|\text{the}) \times P(\text{maison}|\text{house}) = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}
\]
Decoding IBM Model 1; phrases

• How does greedy decoding work at an abstract level?
  • Consider some of the transformation functions.

• How does phrase-based translation differ from word-based translation?
  • E.g., we learn alignments given fully observable models in which word alignments are given.
Bilingual evaluation: BLEU

• In lecture 6-2, $\|\text{Ref1}\| = 16$, $\|\text{Ref2}\| = 17$, $\|\text{Ref3}\| = 16$, and $\|\text{Cn1}\| = 18$ and $\|\text{Cn2}\| = 14$,

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

$$brevity_2 = \frac{16}{14} \quad BP_2 = e^{1-\frac{8}{7}} = 0.8669$$

• Final score of candidate $C$:

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

where

$$p_n = \frac{\sum_{ngram \in C} \text{Count}_R(ngram)}{\sum_{ngram \in C} \text{Count}_C(ngram)}$$

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

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

\[
p_1 = \frac{\sum_{1\text{gram} \in C} Count_R(1\text{gram})}{\sum_{1\text{gram} \in C} Count_C(1\text{gram})} = \frac{1+1+1}{1+1+1} = 1
\]

\[
p_2 = \frac{\sum_{2\text{gram} \in C} Count_R(2\text{gram})}{\sum_{2\text{gram} \in C} Count_C(2\text{gram})} = \frac{1}{2}
\]

\[
BLEU = BP(p_1 p_2)^{\frac{1}{2}} = e^{1 - \left(\frac{4}{3}\right)} \left(\frac{1}{2}\right)^{\frac{1}{2}} \approx 0.5067
\]
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**’).
What is sound?

• A single **tone** is a sinusoidal function of pressure and time.
  
  • **Amplitude**: \( n \). The degree of the displacement in the air. This is similar to ‘loudness’.
    Often measured in **Decibels (dB)**.
  
  • **Frequency**: \( n \). The number of cycles within a unit of time.
    e.g., 1 Hertz (Hz) = 1 vibration/second
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 sheet
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**.
Mel-frequency cepstral coefficients

• **Mel-frequency cepstral coefficients (MFCCs)** are the most popular representation of speech used in ASR.
  • They are the spectra of the logarithms of the mel-scaled filtered spectra of the windows of the waveform.

• Based on what we know about human perception of sound and the source-filter model.
Classifying speakers

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

<table>
<thead>
<tr>
<th>MFCC</th>
<th>Time, $t$</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$P(\text{Image 1} | \text{Image 2}) > P(\text{Image 3} | \text{Image 4})$$
Continuous distributions

• In the past, we used discrete probability functions.
• Since we are now operating with continuous variables, we need to fit continuous probability functions to a discrete number of observations.

• If we assume the 1-dimensional data in this histogram is Normally distributed, we can fit a continuous Gaussian function simply in terms of the mean $\mu$ and variance $\sigma^2$. 
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

$$P(\tilde{x}) = \sum_{j=1}^{M} P(\Gamma_j)P(\tilde{x}|\Gamma_j)$$
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

<table>
<thead>
<tr>
<th>Reference</th>
<th>-</th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>how</td>
<td>∞</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>to</td>
<td>∞</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>recognize</td>
<td>∞</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>speech</td>
<td>∞</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

- See the example in lecture 8-2. *Work it out yourself.*
Speech synthesis
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.
Information retrieval
Information retrieval (IR)

• Given **queries** in natural language, search for documents or information that answers those queries.
  • Returning documents vs. answering the questions directly.

• Evaluating multiple IR systems using **precision** and **recall**.

• The vector space model.

• High-level aspects of singular-value decomposition
The cosine measure

• The **cosine measure** (a.k.a., ‘normalized correlation coefficient’) is

\[
\cos(\mathbf{q}, \mathbf{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} d_i^2}}
\]

where \(\mathbf{q}\) and \(\mathbf{d}\) are \(n\)-dimensional vectors for the query and document, respectively.

• **Larger** values of \(\cos(\mathbf{q}, \mathbf{d})\) means **stronger** correlation, so \(\mathbf{q}\) is ‘closer’ to \(\mathbf{d}_1\) than \(\mathbf{d}_2\) iff \(\cos(\mathbf{q}, \mathbf{d}_1) > \cos(\mathbf{q}, \mathbf{d}_2)\).
Summarization
Summarization

- Reducing a single document or multiple documents down to their most important or salient elements.

  - **Extractive** summarization vs. **synthetic** summaries.

- What features are useful in identifying important phrases or sections? What are their properties?
Determining relevance

• The relevance of sentences and phrases within the text can be approximated by:

  • **Position:** The location of the phrase in the document.
  • **Cues:** The presence of certain words that indicate relevance (e.g., “crucially”, “in conclusion”).
  • **Cohesion:** The distribution of words and their co-occurrences across the document.
ROUGE-2 example

**Candidate:** An egg falls off a wall.

Don’t sit on a wall if you’re an egg.

Horses fail to perform surgery upon an egg.

Humpty Dumpty had a great fall.

Ref 1

Ref 2

Ref 3

\[
ROUGE2 = \frac{\sum_{S \in \{Ref\,Summ\}} \sum_{\text{bigram} \in S} \text{Count}_{match}(\text{bigram})}{\sum_{S \in \{Ref\,Summ\}} \sum_{\text{bigram} \in S} \text{Count}(\text{bigram})} = \frac{2 + 1 + 0}{8 + 7 + 5} = \frac{3}{20}
\]
Miscellaneous
Miscellaneous classification

• Understand the high-level aspects of these models:
  • Support vector machines, and
  • Word-vector representations.
• **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?
Final thoughts

(not thoughts on the final)
NLC in industry

Google

Maluuba

Receptiviti

Nielsen BuzzMetrics

cymfony harnessing influence 2.0™

Wattpad

Twitter

Hakia

Apple

OpenText

IBM

Microsoft

At&T

Nuance

Amazon

Thomson Reuters
Final thoughts

• This course **barely** scratches the surface of **natural language computing**. Talk to these people:

  - Graeme Hirst
  - Gerald Penn
  - Frank Rudzicz
  - Suzanne Stevenson

• Most of the techniques in this course are applicable **generally**.
• Hidden Markov models, e.g., are used almost universally, including in finance, biology, medicine, and robotics.
My 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