natural language computing
This lecture

• An extractive summary of the course.

• Open office hours will follow, TBD
Final Exam Assessment

• A quiz will be posted on Quercus on **23 April**. This is your final assessment.

• You have **3 hours** from start to finish to complete it.

• Those 3 hours must occur between 9h and 21h Eastern Time.

• “No aids allowed”
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.
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.
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).
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

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

\[ P(A, B) = P(A)P(B|A) \]
\[ P(A, B) = P(B)P(A|B) \]
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>Unigram counts:</th>
<th>I</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>Chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2533</td>
<td>927</td>
<td>2417</td>
<td>746</td>
<td>158</td>
<td>1093</td>
<td>341</td>
<td>278</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \text{Count}(w_{t-1}, w_t) )</th>
<th>( w_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>( l )</td>
</tr>
<tr>
<td>( W_{t-1} )</td>
<td></td>
</tr>
<tr>
<td>( I )</td>
<td>5</td>
</tr>
<tr>
<td>( want )</td>
<td>2</td>
</tr>
<tr>
<td>( to )</td>
<td>2</td>
</tr>
<tr>
<td>( eat )</td>
<td>0</td>
</tr>
<tr>
<td>( Chinese )</td>
<td>1</td>
</tr>
<tr>
<td>( food )</td>
<td>15</td>
</tr>
<tr>
<td>( lunch )</td>
<td>2</td>
</tr>
<tr>
<td>( spend )</td>
<td>1</td>
</tr>
</tbody>
</table>
Zipf’s law on the Brown corpus

\[ f \propto \frac{1}{r} \quad \text{i.e., for some } k \quad f \cdot r = k \]
Smoothing as redistribution

- Steal from the rich and give to the poor.
- E.g., $\text{Count}(I\text{ caught } \cdot)$
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.
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**: 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**: \textit{n. the average} amount of information needed to specify one variable \textit{given} that you know another.

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

It’s \textit{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.
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?
Observable Markov model
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>upside</td>
<td>0.1</td>
</tr>
<tr>
<td>down</td>
<td>0.05</td>
</tr>
<tr>
<td>promise</td>
<td>0.05</td>
</tr>
<tr>
<td>friend</td>
<td>0.6</td>
</tr>
<tr>
<td>monster</td>
<td>0.05</td>
</tr>
<tr>
<td>midnight</td>
<td>0.1</td>
</tr>
<tr>
<td>hallow</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>P(word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>upside</td>
<td>0.3</td>
</tr>
<tr>
<td>down</td>
<td>0.0</td>
</tr>
<tr>
<td>promise</td>
<td>0.0</td>
</tr>
<tr>
<td>friend</td>
<td>0.2</td>
</tr>
<tr>
<td>monster</td>
<td>0.05</td>
</tr>
<tr>
<td>midnight</td>
<td>0.05</td>
</tr>
<tr>
<td>hallow</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 $P(\langle up, up \rangle)$

• 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

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 (state \(i\), 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 $\theta = \langle \Pi, A, B \rangle$ so that $P(\mathcal{O}; \theta)$ is maximized for some **training** data $\mathcal{O}$:

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{O}; \theta)$$

• 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 $\theta$, 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

$$\theta = \left\langle \pi_i, \{a_{ij}\}, \{b_i(w)\} \right\rangle$$
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.
The noisy channel

\[ E^* = \arg\max_E P(F|E)P(E) \]
Bilingual evaluation: BLEU

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

\[
\text{brevity}_1 = \frac{17}{18} \quad B_{P1} = 1
\]
\[
\text{brevity}_2 = \frac{16}{14} \quad B_{P2} = e^{1-\left(\frac{8}{7}\right)} = 0.8669
\]

• **Final score** of candidate $C$:

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

where

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

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*

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

- \( p_1 = \frac{\sum_{1\text{gram} \in C} \text{Count}_R(1\text{gram})}{\sum_{1\text{gram} \in C} \text{Count}_C(1\text{gram})} = \frac{1+1+1}{1+1+1} = 1 \)

- \( p_2 = \frac{\sum_{2\text{gram} \in C} \text{Count}_R(2\text{gram})}{\sum_{2\text{gram} \in C} \text{Count}_C(2\text{gram})} = \frac{1}{2} \)

- \( \text{BLEU} = BP(p_1 p_2)^{\frac{1}{2}} = e^{1 - \left(\frac{4}{3}\right)} \left(\frac{1}{2}\right)^{\frac{1}{2}} \approx 0.5067 \)

Assume \( \text{cap}(n) = 2 \) for all \( n \)-grams
Neural language models
Continuous bag of words (1 word context)

Note: we have two vector representations of each word:
\[ \nu_w = x^T W_I \text{ (w}^{th}\text{ row of } W_I) \]
\[ V_w = W_O^T y \text{ (w}^{th}\text{ col of } W_O) \]

'softmax': \[ P(w_o | w_i) = \frac{\exp(V_w^T \nu_{w_i})}{\sum_{w=1}^{W} \exp(V_w^T \nu_{w_i})} \]

Where

\( \nu_w \) 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

Let’s talk about gender at the UofT

However, in word2vec trained on Google News, \textit{man:woman::programmer:homemaker}.

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?
**ELMo: Embeddings from Language Models**

- Instead of a fixed embedding for each word *type*, ELMo considers the entire sentence before embedding each *token*.
- It uses a bi-directional LSTM trained on a specific task.
- Outputs are softmax probabilities on words, as before.
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., $F_1$, $F_2$, $F_3$).

![Formant Diagram](image-url)
The vowel trapezoid

If I asked you about phonemes, I’d probably give you example words. e.g., iy as in sheet

If $F_1$ increases...

If $F_2$ increases...
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 \textit{differently} in MFCC space than speech from another speaker.
• We can \textit{∴} decide if a given observation comes from one speaker or another.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Time, } t & 0 & 1 & \ldots & T \\
\hline
\text{MFCC} & 1 & \text{...} & \text{...} & \text{...} \\
2 & \text{...} & \text{...} & \text{...} & \text{...} \\
3 & \text{...} & \text{...} & \text{...} & \text{...} \\
\vdots & \text{...} & \text{...} & \text{...} & \text{...} \\
42 & \text{...} & \text{...} & \text{...} & \text{...} \\
\hline
\end{array}
\]

\[P(\quad\quad\quad\quad) > P(\quad\quad\quad\quad)\]
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. **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.
Singular value decomposition (SVD)

\[ A = U \cdot \Sigma \cdot V^* \]
SVD example

\[ A_{t \times d} = T_{t \times n} S_{n \times n} (D_{d \times n})^\top \]

<table>
<thead>
<tr>
<th>( A )</th>
<th>( T )</th>
<th>( S )</th>
<th>( D^\top )</th>
</tr>
</thead>
<tbody>
<tr>
<td>natural</td>
<td>1 0 1 0 0 0</td>
<td>nat.</td>
<td>-0.44 -0.30 0.57 0.58 0.25</td>
</tr>
<tr>
<td>language</td>
<td>0 1 0 0 0 0</td>
<td>lang.</td>
<td>-0.13 -0.33 -0.59 0 0.73</td>
</tr>
<tr>
<td>processing</td>
<td>1 1 0 0 0 0</td>
<td>proc.</td>
<td>-0.48 -0.51 -0.37 0 -0.61</td>
</tr>
<tr>
<td>car</td>
<td>1 0 0 1 1 0</td>
<td>car</td>
<td>-0.70 0.35 0.15 -0.58 0.16</td>
</tr>
<tr>
<td>truck</td>
<td>0 0 0 1 0 1</td>
<td>truck</td>
<td>-0.26 0.65 -0.41 0.58 -0.09</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
2.16 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.59 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.28 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.39
\end{bmatrix}
\]

\[
\begin{bmatrix}
-0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\
-0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\
0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\
0 & 0 & 0.58 & 0 & -0.58 & 0.58 \\
-0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22
\end{bmatrix}
\]

• 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.

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

$$
T = \begin{bmatrix}
-0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\
-0.13 & -0.33 & -0.59 & 0 & 0.73 \\
-0.48 & -0.51 & -0.37 & 0 & -0.61 \\
-0.70 & 0.35 & 0.15 & -0.58 & 0.16 \\
-0.26 & 0.65 & -0.41 & 0.58 & -0.09
\end{bmatrix}
$$

$$
D^T = \begin{bmatrix}
-0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\
-0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\
0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\
0 & 0 & 0.58 & 0 & -0.58 & 0.58 \\
-0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22
\end{bmatrix}
$$
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 $A$.

$S =$

\[
\begin{pmatrix}
2.16 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.59 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.28 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.39 \\
\end{pmatrix}
\]

$T =$

\[
\begin{pmatrix}
\cosm & -0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\
\astro & -0.13 & -0.33 & -0.59 & 0 & 0.73 \\
\moon & -0.48 & -0.51 & -0.37 & 0 & -0.61 \\
\car & -0.70 & 0.35 & 0.15 & -0.58 & 0.16 \\
\truck & -0.26 & 0.65 & -0.41 & 0.58 & -0.09 \\
\end{pmatrix}
\]

$D^T =$

\[
\begin{pmatrix}
\begin{array}{cccccc}
\begin{array}{ccccccc}
& d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
\begin{array}{ccccccc}
-0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\
-0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\
0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\
0 & 0 & 0.58 & 0 & -0.58 & 0.58 \\
-0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \\
\end{array}
\end{array}
\end{array}
\end{array}
\]
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