Entropy
Review

• Insofar as language can be modelled statistically, it might help to think of it in terms of dice.

Fair die

- Vocabulary: numbers
- Vocabulary size: 6

Language

- Vocabulary: words
- Vocabulary size: 2– 200,000
Learning probabilities

• What if the symbols are not equally likely?
  • We have to estimate the bias using training data.

Loaded die

• Observe many rolls of the die.
  • e.g.,
    1, 6, 5, 4, 1, 3, 2, 2, ....

Language

• Observe many words.
  • e.g.,
    ...and then I will...
Training vs. testing

- So you’ve **learned** your **probabilities**.
  - Do they model **unseen** data from the **same** source well?

- **Keep rolling** the same dice.
- **Do sides** keep appearing in the **same proportion** as we expect?

- **Keep reading** words.
- **Do words** keep appearing in the **same proportion** as we expect?
Sequences with dependencies

If you consider *all* of the past, you will never gather enough data in order to be useful in practice.

- Imagine you’ve only seen the Brown corpus.
- The sequence ‘the old car’ never appears therein.
- $P(car|the\ old) = 0 \therefore P(the\ old\ car) = 0$

$$P(2,1,4) = P(2)P(1|2)P(4|2,1)$$

$$P(\text{the old car}) = P(\text{the})P(\text{old}|\text{the})P(\text{car}|\text{the old})$$
Sequences with fewer dependencies

• If you **ignore** the past *entirely*, the probability of a sequence is the product of prior (i.e., unigram) probabilities.

\[
P(2,1,4) = P(2)P(1)P(4)
\]

Language involves context. Ignoring that gives weird results, e.g.,

\[
P(2,1,4) = P(2)P(1)P(4) = P(2)P(4)P(1) = P(2,4,1)
\]
Do we consider the recent past?

- This applies the Markov assumption to bigrams.
  - Imagine you’ve only seen the Brown corpus.
  - The sequences ‘the old’ & ‘old car’ do appear therein!
  - \( P(\text{old}|\text{the}) > 0, P(\text{car}|\text{old}) > 0 \implies P(\text{the old car}) > 0 \)
  - Also, \( P(\text{the old car}) > P(\text{the car old}) \)
We *still* have a 0-probability problem

- We got **lucky** with *the old car*.
- Many bigrams are **never** seen.
- If we’re going to use these models in **practice** (*and that’s the whole point!*), we’ll be dealing with **new** text.
- That new text might easily invoke an unseen bigram!

Bigrams we’ve seen after 100K words
Bigrams we’ve never seen after 100K words
Fixing sparseness with discounting

- **Adjust** probabilities such that
  - we **pretend** we’ve seen words that we haven’t, and
  - the probability of observed data is **diminished**.

![Bar charts showing actual and imaginary counts for trout, salmon, sole, haddock, and catfish.](chart)
Fixing sparseness with interpolation

• As before, our training data does not contain ‘the old car’.

• $\hat{P}(w_t|w_{t-2}w_{t-1}) = \lambda_1 P(w_t|w_{t-2}w_{t-1})$
  $+ \lambda_2 P(w_t|w_{t-1})$
  $+ \lambda_3 P(w_t) \neq 0$

• $\hat{P}(\text{car}|\text{the old}) = \lambda_1 P(\text{car}|\text{the old})$
  $+ \lambda_2 P(\text{car}|\text{old})$
  $+ \lambda_3 P(\text{car})$

• $\hat{P}(\text{car}|\text{the old}) = \lambda_2 P(\text{car}|\text{old}) + \lambda_3 P(\text{car})$
Long versus short sequences

• **Long** sequences will tend to have **lower** probability.

\[
P(2,1,4) = P(2)P(1)P(4) \\
P(2,1,4,5) = P(2)P(1)P(4)P(5) \\
\therefore P(2,1,4) > P(2,1,4,5)
\]

\[
P(\text{the old car}) = P(\text{the})P(\text{old})P(\text{car}) \\
P(\text{the old car is}) = P(\text{the})P(\text{old})P(\text{car})P(\text{is}) \\
\therefore P(\text{the old car}) > P(\text{the old car is})
\]

• This is an **effect** of the fact that we’re modelling **only observable** phenomena (e.g., we’re **ignoring** grammar).
Modelling language

• So far, we’ve modelled language as a surface phenomenon using only our observations (i.e., words).

• Language is hugely complex and involves hidden structure (recall: syntax, semantics, pragmatics).

• A ‘true’ model of language would take into account all those things and the proper relations between them.

• Our first hint of modelling hidden structure will come with uncovering grammatical roles (i.e., parts-of-speech)
Parts-of-speech as hidden variables

- Will/MD the/DT chair/NN chair/?? the/DT meeting/NN from/IN that/DT chair/NN?

a) Will the chair chair ...

b) Will the chair chair ...

More later...
Today

- Information theory.
Reminder: rules of logarithms

• You may need these in later assignments (and today):
  • Definition: \( \log_a x = N \leftrightarrow a^N = x \)
  • Product: \( \log_a (xy) = \log_a x + \log_a y \)
  • Quotient: \( \log_a \left( \frac{x}{y} \right) = \log_a x - \log_a y \)
  • Power: \( \log_a (x^p) = p \log_a x \)
  • Base change: \( \log_a x = \frac{\log_b x}{\log_b a} \)

• Reminder: avoid common logarithmotechnic errors:
  • \( \log_a (x + y) \neq \log_a x + \log_a y \)
  • \( \log_a (x - y) \neq \log_a x - \log_a y \)
Information

• Imagine Darth Vader is about to say either “yes” or “no” with equal probability.
  • You don’t know what he’ll say.

• You have a certain amount of uncertainty – a lack of information.
Information

• Imagine you then observe Darth Vader saying “no”
• Your uncertainty is gone; you’ve received information.
• How much information do you receive about event $E$ when you observe it?

\[
I(E) = \log_2 \frac{1}{P(E)}
\]

\[
I(\text{no}) = \log_2 \frac{1}{P(\text{no})} = \log_2 \frac{1}{\frac{1}{2}} = 1 \text{ bit}
\]
Information

- Imagine Darth Vader is about to roll a fair die.
- You have more uncertainty about an event because there are more possibilities.
- You receive more information when you observe it.

\[
I(5) = \log_2 \frac{1}{P(5)} = \log_2 \frac{1}{\frac{1}{6}} \approx 2.59 \text{ bits}
\]
Information is additive

• From $k$ independent, equally likely events $E$,

$$I(E^k) = \log_2 \frac{1}{P(E^k)} = \log_2 \frac{1}{P(E)^k}$$

$$I(k \text{ binary decisions}) = \log_2 \frac{1}{\left(\frac{1}{2}\right)^k} = k \text{ bits}$$

• For a unigram model, with each of 50K words $w$ equally likely,

$$I(w) = \log_2 \frac{1}{\frac{1}{50000}} \approx 15.61 \text{ bits}$$

and for a sequence of 1K words in that model,

$$I(w^k) = \log_2 \frac{1}{\left(\frac{1}{50000}\right)^{1000}}$$
Information with unequal events

• An information source $S$ emits symbols without memory from a vocabulary $\{w_1, w_2, \ldots, w_n\}$. Each symbol has its own probability $\{p_1, p_2, \ldots, p_n\}$

• What is the average amount of information we get in observing the output of source $S$?

• You still have 6 events that are possible – but you’re fairly sure it will be ‘No’.
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:

$$E[X] = \sum_{x \in X} p(x) x$$
Entropy – examples

Yes (0.1)  No (0.7)  Maybe (0.04)  Sure (0.03)  Darkside (0.06)  Destiny (0.07)

\[ H(S) = \sum_i p_i \log_2 \frac{1}{p_i} \]

\[ = 0.7 \log_2 (1/0.7) + 0.1 \log_2 (1/0.1) + \ldots \]

\[ = 1.542 \text{ bits} \]

There is **less** average uncertainty when the probabilities are ‘skewed’.

\[ H(S) = \sum_i p_i \log_2 \frac{1}{p_i} = 6 \left( \frac{1}{6} \log_2 \frac{1}{1/6} \right) \]

\[ = 2.585 \text{ bits} \]
Entropy characterizes the distribution

- ‘Flatter’ distributions have a higher entropy because the choices are more equivalent, on average.
- So which of these distributions has a lower entropy?
Low entropy makes decisions easier

• When predicting the next word, e.g., we’d like a distribution with lower entropy.
• Low entropy ≡ less uncertainty
Bounds on entropy

- **Maximum:** uniform distribution $S_1$. Given $M$ choices,
  \[ H(S_1) = \sum_i p_i \log_2 \frac{1}{p_i} = \sum_i \frac{1}{M} \log_2 \frac{1}{1/M} = \log_2 M \]

- **Minimum:** only one choice,
  \[ H(S_2) = p_i \log_2 \frac{1}{p_i} = 1 \log_2 1 = 0 \]
Alternative notions of entropy

- Entropy is *equivalently*:
  - The *average* amount of *information* provided by symbols in a vocabulary,
  - The *average* amount of *uncertainty* you have *before* observing a symbol from a vocabulary,
  - The *average* amount of ‘*surprise*’ you receive when observing a symbol,
  - The number of bits needed to communicate that alphabet
  - Aside: Shannon (the same as yesterday) showed that you *cannot* have a *coding scheme* that can communicate the vocabulary more *efficiently* than $H(S)$
Coding symbols efficiently

• If we want to transmit Vader’s words efficiently, we can encode them so that more probable words require fewer bits.
  • On average, fewer bits will need to be transmitted.

<table>
<thead>
<tr>
<th>Word (sorted)</th>
<th>Linear Code</th>
<th>Huffman Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>Yes</td>
<td>001</td>
<td>11</td>
</tr>
<tr>
<td>Destiny</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>Darkside</td>
<td>011</td>
<td>1001</td>
</tr>
<tr>
<td>Maybe</td>
<td>100</td>
<td>10000</td>
</tr>
<tr>
<td>Sure</td>
<td>101</td>
<td>10001</td>
</tr>
</tbody>
</table>

- Yes (0.1)
- No (0.7)
- Maybe (0.04)
- Sure (0.03)
- Darkside (0.06)
- Destiny (0.07)
Aside: Coding symbols efficiently

• Another way of looking at this is through the (binary) Huffman tree ($r$-ary trees are often flatter, all else being equal):

<table>
<thead>
<tr>
<th>Word</th>
<th>Linear Code</th>
<th>Huffman Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>Yes</td>
<td>001</td>
<td>11</td>
</tr>
<tr>
<td>Destiny</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>Darkside</td>
<td>011</td>
<td>1001</td>
</tr>
<tr>
<td>Maybe</td>
<td>100</td>
<td>10000</td>
</tr>
<tr>
<td>Sure</td>
<td>101</td>
<td>10001</td>
</tr>
</tbody>
</table>
Entropy of several variables

- Consider the vocabulary of a meteorologist describing temperature and wetness.

\[
P(W = \text{dry}) = 0.6, \\
P(W = \text{wet}) = 0.4 \\
H(W) = 0.6 \log_2 \frac{1}{0.6} + 0.4 \log_2 \frac{1}{0.4} = 0.970951 \text{ bits}
\]

\[
P(T = \text{hot}) = 0.3, \\
P(T = \text{mild}) = 0.5, \\
P(T = \text{cold}) = 0.2 \\
H(T) = 1.48548 \text{ bits}
\]

But \(W\) and \(T\) are not independent, \(P(W, T) \neq P(W)P(T)\)
Joint entropy

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

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

• **Hint:** this is **very** similar to univariate entropy – we just replace univariate probabilities with joint probabilities and sum over everything.
Entropy of several variables

• Consider joint probability, $P(W, T)$

<table>
<thead>
<tr>
<th></th>
<th>cold</th>
<th>mild</th>
<th>hot</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dry</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>wet</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

• Joint entropy, $H(W, T)$, computed as a sum over the space of joint events ($W = w, T = t$)

\[
H(W, T) = 0.1 \log_2 \frac{1}{0.1} + 0.4 \log_2 \frac{1}{0.4} + 0.1 \log_2 \frac{1}{0.1} \\
+ 0.2 \log_2 \frac{1}{0.2} + 0.1 \log_2 \frac{1}{0.1} + 0.1 \log_2 \frac{1}{0.1} = 2.32193 \text{ bits}
\]

Notice $H(W, T) \approx 2.32 < 2.46 \approx H(W) + H(T)$
Entropy given knowledge

• In our example, **joint entropy** of two variables together is **lower** than the **sum** of their **individual** entropies
  • \( H(W, T) \approx 2.32 < 2.46 \approx H(W) + H(T) \)

• Why?

• Information is **shared** among variables
  • There are **dependencies**, e.g., between temperature and wetness.
  • E.g., if we knew exactly how **wet** it is, is there **less confusion** about what the **temperature** is ... ?
Conditional entropy

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

• A.k.a ‘equivocation’

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

• **Hint:** this is *very* similar to how we compute expected values in general distributions.
Entropy given knowledge

- Consider **conditional** probability, \( P(T|W) \)

\[
P(T|W) = \frac{P(W,T)}{P(W)}
\]

<table>
<thead>
<tr>
<th>( P(W,T) )</th>
<th>( T = \text{cold} )</th>
<th>mild</th>
<th>hot</th>
<th>( P(W) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W = \text{dry} )</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>wet</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

| \( P(T|W) \) | \( T = \text{cold} \) | mild | hot | \( P(W) \) |
|----------------|-----------------|-----|-----|-----|
| \( W = \text{dry} \) | 0.1/0.6 | 0.4/0.6 | 0.1/0.6 | 1.0 |
| wet | 0.2/0.4 | 0.1/0.4 | 0.1/0.4 | 1.0 |

CSC401/2511 – Spring 2017
Entropy given knowledge

- Consider **conditional** probability, $P(T|W)$

| $P(T|W)$  | $T = $ cold | mild | hot | $W = $ dry |
|-----------|-------------|------|-----|------------|
| $W = $ dry | 1/6         | 2/3  | 1/6 | 1.0        |
| wet       | 1/2         | 1/4  | 1/4 | 1.0        |

- $H(T|W = \text{dry}) = H\left(\left\{\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right\}\right) = 1.25163$ bits

- $H(T|W = \text{wet}) = H\left(\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}\right) = 1.5$ bits

- **Conditional entropy** combines these:

  $$
  H(T|W) = [p(W = \text{dry})H(T|W = \text{dry})] + [p(W = \text{wet})H(T|W = \text{wet})]
  = 1.350978 \text{ bits}
  $$
Equivocation removes uncertainty

• Remember $H(T) = 1.48548$ bits
• $H(W, T) = 2.32193$ bits
• $H(T|W) = 1.350978$ bits

• How much does $W$ tell us about $T$?
  • $H(T) - H(T|W) = 1.48548 - 1.350978 \approx 0.1345$ bits
  • Well, a little bit!

Entropy (i.e., confusion) about temperature is reduced if we know how wet it is outside.
Perhaps $T$ is more informative?

- Consider another conditional probability, $P(W|T)$

| $P(W|T)$ | $T = \text{cold}$ | mild | $T = \text{hot}$ |
|----------|---------------------|------|------------------|
| $W = \text{dry}$ | 0.1/0.3 | 0.4/0.5 | 0.1/0.2 |
| $W = \text{wet}$ | 0.2/0.3 | 0.1/0.5 | 0.1/0.2 |
| 1.0 | 1.0 | 1.0 |

- $H(W|T = \text{cold}) = H\left(\left\{\frac{1}{3}, \frac{2}{3}\right\}\right) = 0.918295$ bits
- $H(W|T = \text{mild}) = H\left(\left\{\frac{4}{5}, \frac{1}{5}\right\}\right) = 0.721928$ bits
- $H(W|T = \text{hot}) = H\left(\left\{\frac{1}{2}, \frac{1}{2}\right\}\right) = 1$ bit

- $H(W|T) = 0.8364528$ bits
Equivocation removes uncertainty

- $H(T) = 1.48548$ bits
- $H(W) = 0.970951$ bits
- $H(W, T) = 2.32193$ bits
- $H(T|W) = 1.350978$ bits
- $H(T) - H(T|W) \approx 0.1345$ bits

• How much does $T$ tell us about $W$ on average?
  - $H(W) - H(W|T) = 0.970951 - 0.8364528 \approx 0.1345$ bits

• Interesting ... is that a coincidence?
Mutual information

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

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

- **Hint:** The amount of uncertainty **removed** in variable \(X\) if you know \(Y\).
- **Hint2:** If \(X\) and \(Y\) are **independent**, \(p(x, y) = p(x)p(y)\), then

\[
\log_2 \frac{p(x,y)}{p(x)p(y)} = \log_2 1 = 0 \ \forall x, y \text{ – there is no mutual information!}
\]
Relations between entropies

\[ H(X, Y) = H(X) + H(Y) - I(X; Y) \]
Preview – the noisy channel

• Messages can get **distorted** when passed through a **noisy** conduit – *how much information is lost/retained?*

• **Signals**

• **Symbols**
  - Sexual abuse
  - Locker room talk

• **Languages**
  - Hello, computer
  - Bonjour, ordinateur
Relating corpora
Relatedness of two distributions

- How similar are the two probability distributions?
- e.g., Distribution $P$ learned from *Shakespeare*
  Distribution $Q$ learned from *Wall Street Journal*
Kullback-Leibler divergence

• **KL divergence:** \( n. \) the *average log difference* between the distributions \( P \) and \( Q \), relative to \( Q \).
  a.k.a. *relative entropy*.

\textit{caveat:} we assume \( 0 \log 0 = 0 \)
Kullback-Leibler divergence

\[
D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}
\]
Example Kullback-Leibler divergence

- KL divergence generalizes to **continuous** distributions.
- Below, $D_{KL}(\text{green} \mid \mid \text{blue}) > D_{KL}(\text{purple} \mid \mid \text{blue})$
Kullback-Leibler divergence

• KL divergence:
  • is not symmetric, $D_{KL}(P \| Q) \neq D_{KL}(Q \| P)$
  • is the average number of extra bits required to code samples from $P$ when using a code based on $Q$.
  • is (almost) like a ‘distance’:
    • $D_{KL}(P \| Q) \geq 0 \ \forall P, Q$
    • $D_{KL}(P \| Q) = 0$ iff $P$ and $Q$ are identical.

• Aside:

$$I(P; Q) = D_{KL}(P(X, Y) \| P(X)P(Y))$$
Applications of KL divergence

- Often used towards some **other purpose**, e.g.,
  - In **evaluation** to say that **purple** is a **better** model than **green** of the **true distribution** **blue**.
  - In **machine learning** to adjust the parameters of **purple** to be, e.g., less like **green** and more like **blue**.
Entropy as intrinsic LM evaluation

- **Cross-entropy**, measures how difficult it is to encode an event drawn from a *true probability* $p$ given a *model* based on a distribution $q$.

- What if we don’t know the *true probability* $p$?
  - We’d have to estimate $p$.
  - We estimate $p$ by estimating the *probability of a test corpus* $C$ using the distribution $q$:
    $$P_q(C)$$
Probability of a corpus?

- The probability $P(C)$ of a corpus $C$ requires the same assumptions that allow us to compute the probability $P(s_i)$ of a sentence $s_i$.

<table>
<thead>
<tr>
<th>Chain rule</th>
<th>Sentence</th>
<th>Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(s_i) = P(w_1) \prod_{t=2}^{n} P(w_t</td>
<td>w_1:(t-1))$</td>
<td>$P(C) = P(w_1) \prod_{t=2}^{</td>
</tr>
</tbody>
</table>

- Regardless of the LM used for $P(s_i)$, we assume complete independence between sentences.
Intrinsic evaluation – Cross-entropy

• **Cross-entropy** of a LM $M$ and a *new* test corpus $C$ with size $\|C\|$ (total number of words), where sentence $s_i \in C$, is *approximated* by:

$$H(C; M) = - \frac{\log_2 P_M(C)}{\|C\|} = - \frac{\sum_i \log_2 P_M(s_i)}{\sum_i \|s_i\|}$$

• **Perplexity** is similar:

$$PP_M(C) = 2^{H(C; M)}$$
Reading

• Manning & Schütze: Section 2.2, 5.3-5.3.2

• Thanks to Roni Rosenfeld (CMU) for the meteorological examples.