Introduction to NLP

213.

Evaluation of Language Models

Evaluation of LM

- Extrinsic
  - Use in an application
- Intrinsic
  - Cheaper
  - Based on information theory
- Correlate the two for validation purposes

Information Theory

- It is concerned with data transmission, data compression, and measuring the amount of information.
- Applies to statistical physics, economics, linguistics.

Information and Uncertainty

- The decrease in uncertainty is called information
- Example
  - we know that a certain event will happen next week
  - then we learn that it is more likely to happen on a workday
  - the new information reduces the uncertainty
  - the more new information we get, the smaller the remaining uncertainty

Entropy

- Entropy tells us how informative a random variable is.

$$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Examples

- One symbol (a)
  - uncertainty is 0
- Two symbols (a,b)
  - uncertainty is 1
  - we can reduce it to 0 by using one bit of information (a=0,b=1)
- Four symbols (a,b,c,d)
  - we need two bits of information (e.g., a=00,b=01,c=10,d=11)
- In general we need
  - $\log_2 k$ bits, where k is the number of symbols
  - note: this only holds if all symbols are equiprobable

Amount of Surprise

- Amount of surprise (given a general prob. distribution)

  $-\log_2 p(x)$ - for a specific outcome x

- If the distribution is uniform:

  $p(x) = 1/k$

  $k = 1/p(x)$

  $\log_2 k = \log_2 (1/p(x)) = -\log_2 p(x)$

- Average surprise

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) = E(\log_2 \frac{1}{p(X)})$$

- Information need

  H(X) = 0 means that we have all the information that we need

  H(X) = 1 means that we need one bit of information, etc.

Entropy Example

- Sample 8-character language: A E I O U F G H

$$H(X) = -\sum_{i=1}^{8} p(i)\log_2 p(i) = -\sum_{i=1}^{8} \tfrac{1}{8}\log_2 \tfrac{1}{8} = \log_2 8 = 3$$

- Three bits per character if the characters are equiprobable

A	E	I	O	U	F	G	H
000	001	010	011	100	101	110	111

Simplified Polynesian

- Six characters: P T K A I U, not equiprobable

char:	P	T	K	A	I	U
prob:	1/8	1/4	1/8	1/4	1/8	1/8

$$
\begin{aligned}
H(X) &= -\sum_{i \in L} p(i) \log p(i) \\
&= -[4 \times \frac{1}{8} \log \frac{1}{8} + 2 \times \frac{1}{4} \log \frac{1}{4}] \\
&= 2.5
\end{aligned}
$$

- This number (2.5) can lead one to believe that 3 bits per character are needed
  - e.g. 000, 001, 010, 100, 101, 111

Simplified Polynesian

- More efficient encoding

char:	P	T	K	A	I	U
code:	100	00	101	01	110	111

- Longer codes for less frequent characters
- This can lower the average number of bits per character to the theoretical estimate of 2.5
- Under what assumption, though?

Joint Entropy

- Amount of information to specify both *x* and *y*.

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(x,y)$$

- Measures the amount of surprise of seeing a specific tag bigram.

Conditional Entropy

- If we know *x*, how much additional information is needed to know *y*.

$$
\begin{aligned}
H(Y|X) &= \sum_{x \in X} p(x) H(Y|X = x) \\
&= \sum_{x \in X} p(x) [- \sum_{y \in Y} p(y|x) \log_2 p(y|x)] \\
&= - \sum_{x \in X} \sum_{y \in Y} p(x) p(y|x) \log_2 p(y|x) \\
&= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y|x)
\end{aligned}
$$

Chain Rule for Entropy

- Chain rule for entropy

$$
\begin{aligned}
H(X, Y) &= H(X) + H(Y|X) \\
H(X_1, \ldots, X_n) &= H(X_1) + H(X_2|X_1) \\
&\quad + H(X_3|X_1, X_2) + \ldots \\
&\quad + H(X_n|X_1, \ldots, X_{n-1})
\end{aligned}
$$

Probabilities of Syllables

- P(C,·) and P(·,V) - marginal probabilities

p	t	k	a	i	u
1/8	3/4	1/8	1/2	1/4	1/4

- P(C,V)

	p	t	k	
a	$\frac{1}{16}$	$\frac{3}{8}$	$\frac{1}{16}$	$\frac{1}{2}$
i	$\frac{1}{16}$	$\frac{3}{16}$	0	$\frac{1}{4}$
u	0	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{4}$
	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	

Surprise in Syllables

$$H(C,V) = H(C) + H(V|C) \approx 1.061 + 1.375 \approx 2.44$$

a. $H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \approx 1.061$

b. $H(V|C) = -\sum_{c \in C} \sum_{v \in V} p(c,v) \log p(v|c) = 1.375$

- Example

$$p(V = a|C = p) = \tfrac{1}{2} \text{ because } \tfrac{1}{16} \text{ is half of } \tfrac{1}{8}$$

Polynesian Syllables (cont'd)

$$
\begin{aligned}
H(C) &= -\sum_{i \in L} p(i) \log p(i) \\
&= -[2 \times \frac{1}{8} \log \frac{1}{8} + \frac{3}{4} \log \frac{3}{4}] \\
&= 2 \times \frac{1}{8} \log 8 + \frac{3}{4} (\log 4 - \log 3) \\
&= 2 \times \frac{1}{8} \times 3 + \frac{3}{4} (2 - \log 3) \\
&= \frac{3}{4} + \frac{6}{4} - \frac{3}{4} \log 3 \\
&= \frac{9}{4} - \frac{3}{4} \log 3 \approx 1.061
\end{aligned}
$$

Polynesian Syllables (cont'd)

$$
\begin{aligned}
H(V|C) &= -\sum_{x \in C} \sum_{y \in V} p(x,y) \, log \, p(y|x) \\
&= -[1/16 \, log \, 1/2 + 3/8 \, log \, 1/2 + 1/16 \, log \, 1/2 \\
&\quad +1/16 \, log \, 1/2 + 3/16 \, log \, 1/4 + 0 \, log \, 0 \\
&\quad +0 \, log \, 0 + 3/16 \, log \, 1/4 + 1/16 \, log \, 1/2] \\
&= 1/16 \, log 2 \; + 3/8 \, log \, 2 \; + 1/16 \, log \, 2 \\
&\quad +1/16 \, log \, 2 + 3/16 \, log \, 4 \\
&\quad +3/16 \, log \, 4 + 1/16 \, log \, 2] \\
&= 11/8 \\
&= 1.375
\end{aligned}
$$

Pointwise Mutual Information

- Measured between two points (not two distributions)

$$I(x;y) = \log \frac{p(x,y)}{p(x)p(y)}$$

- If $p(x,y) = p(x)p(y)$, then $I(x;y) = \log 1 = 0$ (independence)

Mutual Information

- Same, but for two distributions (not points)
- How much information does one of the distributions contain about the other one.

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

Kullback-Leibler (KL) Divergence

- Measures how far two distributions are from one another

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

- It measures the number of bits needed to encode p by using q.
- Always non-negative
- D(p||q) = 0, iff p=q
- D(p||q) = ∞, iff $\exists x \in X$ such that p(x)>0 and q(x)=0
- Not symmetric

Divergence as Mutual Information

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

$$I(X;Y) = D(p(x,y) || p(x)p(y))$$

Perplexity

- Does the model fit the data?
  - A good model will give a high probability to a real sentence
- Perplexity
  - Average branching factor in predicting the next word
  - Lower is better (lower perplexity -> higher probability)
  - N = number of words

$$Per = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Perplexity

- Example:
  - A sentence consisting of N equiprobable words: $p(w_i) = 1/k$

$$Per = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

  - Per = $((k^{-1})^N)^{(-1/N)} = k$
- Perplexity is like a (weighted) branching factor
- Logarithmic version
  - the exponent is = #bits to encode each word

$$Per = 2^{-(1/N)\sum \log_2 P(w_i)}$$

The Shannon Game

- Consider the Shannon game:
  - Connecticut governor Ned Lamont said ...
- What is the perplexity of guessing a digit if all digits are equally likely? Do the math.
  - 10
- How about a letter?
  - 26
- How about guessing A ("operator") with a probability of 1/4, B ("sales") with a probability of 1/4 and 10,000 other cases with a probability of 1/2 total
  - example modified from Joshua Goodman.

Perplexity Across Distributions

- What if the actual distribution is very different from the expected one?
- Example:
  - All of the 10,000 other cases are equally likely but P(A) = P(B) = 0.
- Cross-entropy = log (perplexity), measured in bits

$$\mathrm{H}(p,q) = -\sum_x p(x)\,\log q(x).$$

Sample Values for Perplexity

- Wall Street Journal (WSJ) corpus
  - 38 M words (tokens)
  - 20 K types
- Perplexity
  - Evaluated on a separate 1.5M sample of WSJ documents
  - Unigram 962
  - Bigram 170
  - Trigram 109
  - More recent results – 47.7 (Yang et al. 2017 using AWD-LSTM

Word Error Rate

- Another evaluation metric
  - Number of insertions, deletions, and substitutions
  - Normalized by sentence length
  - Same as Levenshtein Edit Distance
- Example:
  - governor Ned Lamont met with the mayor
  - the governor met the senator
  - 3 deletions + 1 insertion + 1 substitution = WER of 5

Issues

- Out of vocabulary words (OOV)
  - Split the training set into two parts
  - Label all words in part 2 that were not in part 1 as <UNK>
- Clustering
  - e.g., dates, monetary amounts, organizations, years

Long Distance Dependencies

- This is where n-gram language models fail by definition
- Missing syntactic information
  - **The students** who participated in the game **are** tired
  - **The student** who participated in the game **is** tired
- Missing semantic information
  - **The pizza** that I had last night was **tasty**
  - **The class** that I had last night was **interesting**

Other Ideas in LM

- Skip-grapm models
  - **Ms.** Jane **Doe**, **Ms.** Mary **Doe**
- Syntactic models
  - Condition words on other words that appear in a specific syntactic relation with them
- Caching models
  - Take advantage of the fact that words appear in bursts

Limitations

- Still no general solution for long-distance dependencies
- Cannot handle linear combinations, e.g.,
  - Cats eat mice
  - People eat broccoli
  - Cats ear broccoli
  - People eat mice
- Possible solution – use phrases or n-grams as features (combinatorial explosion)

External Resources

- Google n-gram corpus
  - http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html
- Google book n-grams
  - http://ngrams.googlelabs.com/

N-gram External Links

- http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html
- http://norvig.com/mayzner.html
- http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
- https://books.google.com/ngrams/
- http://www.elsewhere.org/pomo/
- http://pdos.csail.mit.edu/scigen/
- http://www.magliery.com/Band/
- http://www.magliery.com/Country/
- http://johno.jsmf.net/knowhow/ngrams/index.php
- http://www.decontextualize.com/teaching/rwet/n-grams-and-markov-chains/
- http://gregstevens.com/2012/08/16/simulating-h-p-lovecraft
- http://kingjamesprogramming.tumblr.com/