Introduction to generative models of language

- What are they?
- Why they're important
- Issues for counting words
- Statistics of natural language
- **Unsmoothed n-gram models**

Goals

- Determine the next word in a sequence
  - Probability distribution across all words in the language
  - $P(w_n \mid w_1\ w_2 \ldots w_{n-1})$
- Determine the probability of a sequence of words
  - $P(w_1\ w_2 \ldots w_{n-1}\ w_n)$

Probability of a word sequence

- P(w_1 w_2 … w_{n-1} w_n)

$$P(w_1^n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1^2)\ \ldots P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

- Determine, e.g., P(I see what you mean)
  - calculate w.r.t. a particular corpus

Computing the probabilities

I see what I eat and
I eat what I see .

P(I see what you mean)

$$P(w_1^n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1^2)\ \ \ldots P(w_n|w_1^{n-1})$$

P (w_1) = P (I) = # of I's / # of word tokens = 4 / 12 = .33

P (w_2 I w_1)= P (see I I) = # of 'I see' / # of I's= 2 / 4 = .5

P (w_3 I w_1 w_2)= P (what I I see) = # of 'I see what' / # of 'I see'=

= 1 / 2 = .5

= .33 x .5 x .5 x …

Probability of a word sequence

- P (w_1 w_2 … w_{n-1} w_n)

$$P(w_1^n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1^2)\ \ldots P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

- Problem?

Predict the next word

- $P(w_n \mid w_1\ w_2 \ldots w_{n-1}) = P(w_n \mid w_1^{n-1})$
  - Same problem

- Solution: *approximate* the probability of a word given all the previous words…

N-gram approximations

- Markov assumption: probability of some future event (next word) depends only on a limited history of preceding events (previous words)
- Bigram model

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

predict next word

- Trigram model
  - $P(w_n | w_1^{n-1}) = P(w_n | w_{n-2} w_{n-1})$

Probability of a word sequence: bigram estimation

P (w_1 w_2 … w_{n-1} w_n)

$$P(w_1^n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1^2)\ \ldots P(w_n|w_1^{n-1})$$

= $P(w_1)$ $P(w_2|w_1)$ $P(w_3|w_2)$ $P(w_4|w_3)$ ... $P(w_n|w_{n-1})$

= $P(w_1|\langle s\rangle)$ $P(w_2|w_1)$ $P(w_3|w_2)$ ... $P(w_n|w_{n-1})$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

Training N-gram models

- N-gram models can be trained by counting and normalizing
  - Bigrams

$$P(w_n \mid w_{n-1}) = \frac{Count(w_{n-1} w_n)}{Count(w_{n-1})}$$

  - General case

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{Count(w_{n-N+1}^{n-1} w_n)}{Count(w_{n-N+1}^{n-1})}$$

  - An example of Maximum Likelihood Estimation (MLE)
    - Resulting parameter set is one in which the likelihood of the training set T given the model M (i.e. P(TIM)) is maximized.

Exercises

I see what I eat and
I eat what I see .

P(eat what) =

P(see I what I) =

Bigram grammar fragment

- Berkeley Restaurant Project

eat on	.16	eat Thai	.03
eat some	.06	eat breakfast	.03
eat lunch	.06	eat in	.02
eat dinner	.05	eat Chinese	.02
eat at	.04	eat Mexican	.02
eat a	.04	eat tomorrow	.01
eat Indian	.04	eat dessert	.007
eat today	.03	eat British	.001

- Can compute the probability of a complete string
  - P (I want to eat British food) = P(II<s>) P(wantII) P(tol want) P(eatIto) P(BritishIeat) P(foodIBritish)

Bigram counts

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

- Note the number of 0's…

Bigram probabilities

- Problem for the maximum likelihood estimates: sparse data

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

Quality of N-gram models

- Accuracy increases as N increases
  - Train various N-gram models and then use each to generate random sentences.
  - Corpus: Complete works of Shakespeare
    - **Unigram:** *Will rash been and by I the me loves gentle me not slavish page, the and hour; ill let*
    - **Bigram:** *What means, sir. I confess she? Then all sorts, he is trim, captain.*
    - **Trigram:** *Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.*
    - **Quadrigram:** *They say all lovers swear more performance than they are wont to keep obliged faith unforfeited!*

Strong dependency on training data

- Trigram model from WSJ corpus
  - *They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions*