CS474 Natural Language Processing

- Last class
  - Introduction to generative models of language
    - » What are they?
    - » Why they're important
    - » Issues for counting words
    - » Statistics of natural language
- Today
  - N-gram models (unsmoothed)
  - Part-of-speech tagging intro

N-gram model

- Uses the previous N-1 words to predict the next one
  - 2-gram: bigram
  - 3-gram: trigram
- In speech recognition, these statistical models of word sequences are referred to as a **language model**

Models of word sequences

- Simplest model
  - Let any word follow any other word
    - » P (word1 follows word2) = 1/# words in English
- Probability distribution at least obeys actual relative word frequencies
    - » P (word1 follows word2) =
      # occurrences of word1 / # words in English
- Pay attention to the preceding words
  - "Let's go outside and take a [ ]"
    - » walk very reasonable
    - » break quite reasonable
    - » lion less reasonable
  - Compute conditional probability P (walk| let's go…)

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

N-gram approximations

- Bigram model

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

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

- Trigram model
  - Conditions on the two preceding words
- N-gram approximation

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

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

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(I|<s>) P(want|I) P(to|want) P(eat|to) P(British|eat) P(food|British)

Training N-gram models

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

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

  - General case

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

  - 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(T|M)) is maximized.

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

Accuracy 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*

CS474 Natural Language Processing

- Last class
  - Introduction to generative models of language
    - What are they?
    - Why they're important
    - Issues for counting words
    - Statistics of natural language
- Today
  - N-gram models (unsmoothed)
  - Part-of-speech tagging intro

Part of speech tagging

"There are 10 parts of speech, and they are all troublesome."
-Mark Twain

- POS tags are also known as word classes, morphological classes, or lexical tags.

- Typically much larger than Twain's 10:
  - Penn Treebank: 45
  - Brown corpus: 87
  - C7 tagset: 146

Part of speech tagging

- Assign the correct part of speech (word class) to each word/token in a document

  "The/DT planet/NN Jupiter/NNP and/CC its/PPS moons/NNS are/VBP in/IN effect/NN a/DT mini-solar/JJ system/NN ,/, and/CC Jupiter/NNP itself/PRP is/VBZ often/RB called/VBN a/DT star/NN that/IN never/RB caught/VBN fire/NN ./."

- Needed as an initial processing step for a number of language technology applications
  - Answer extraction in Question Answering systems
  - Base step in identifying syntactic phrases for IR systems
  - Critical for word-sense disambiguation (WordNet apps)
  - Information extraction
  - …

Why is p-o-s tagging hard?

- Ambiguity
  - He will race/VB the car.
  - When will the race/NOUN end?
  - The boat floated/' VBN down the river sank.

- Average of ~2 parts of speech for each word

- The number of tags used by different systems varies a lot. Some systems use < 20 tags, while others use > 400.

Hard for Humans

- particle vs. preposition
  - He talked *over* the deal.
  - He talked *over* the telephone.
- past tense vs. past participle
  - The horse *walked* past the barn.
  - The horse *walked* past the barn fell.
- noun vs. adjective?
  - The *executive* decision.
- noun vs. present participle
  - *Fishing* can be fun.

To obtain gold standards for evaluation, annotators rely on a set of tagging guidelines.

From Ralph Grishman, NYU

Penn Treebank Tagset

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	*+,%, &*
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Among easiest of NLP problems

- State-of-the-art methods achieve ~97% accuracy.
- Simple heuristics can go a long way.
  - ~90% accuracy just by choosing the most frequent tag for a word (MLE)
  - To improve reliability: *need to use some of the local context.*
- But defining the rules for special cases can be time-consuming, difficult, and prone to errors and omissions

Approaches

1. rule-based**: involve a large database of hand-written disambiguation rules, e.g. that specify that an ambiguous word is a noun rather than a verb if it follows a determiner.**
2. probabilistic**: resolve tagging ambiguities by using a training corpus to compute the probability of a given word having a given tag in a given context.**

   - HMM tagger
3. hybrid corpus-/rule-based**: E.g. transformation-based tagger (Brill tagger); learns symbolic rules based on a corpus.**
4. ensemble methods**: combine the results of multiple taggers.**