Ngram Review

October13, 2017
Professor Meteer

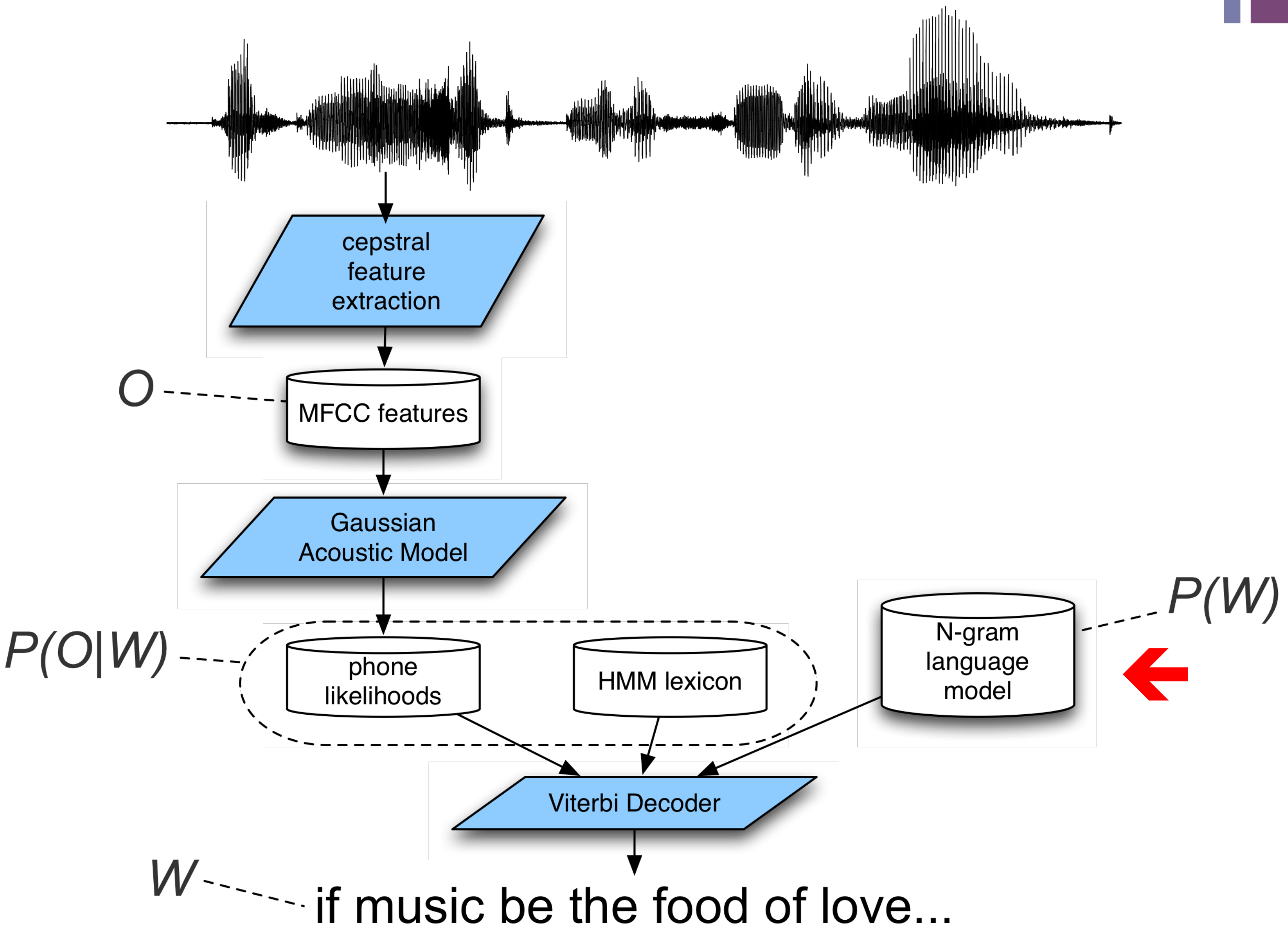
CS 136 Lecture 10
Language Modeling

Thanks to Dan Jurafsky for these slides

+ ASR components

- Feature Extraction, MFCCs, start of Acoustic
- HMMs, the Forward and Viterbi algorithms
- Baum-Welch (Forward-Backward)
- Acoustic Modeling and GMMs
- N-grams and Language Modeling
- Search and Advanced Decoding
- Dealing with Variation

+ Speech Recognition Architecture



Thanks to Dan Jurafsky for these slides

+ “Prior” probability of word sequence

- The ***language model*** captures the knowledge we have about what words to expect based on the ***context***
  - “Grammar” creates a finite state model of all (and only) possible sequences of words.
  - “Statistical Language Model” (SLM) encodes the probability of sequences of words based on counts from data.
- The ***context*** is both the domain and the immediate previous context
  - Domain: Language modeling data should match the target recognition data
  - Immediate context: Previous “n” words (usually 3-4)

+ Language Modeling

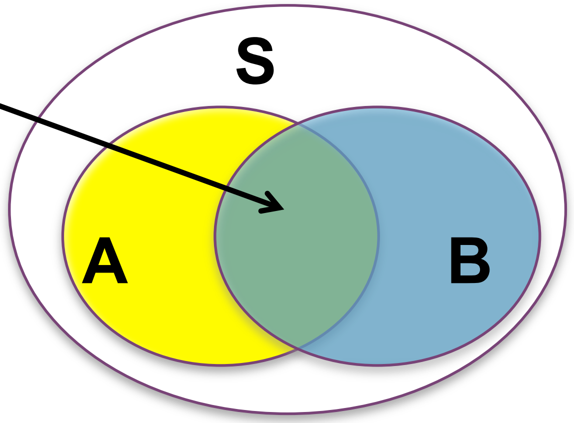
- We want to compute
  - $P(w_1,w_2,w_3,w_4,w_5 \ldots w_n) = P(W)$
  - = the probability of a sequence
- Alternatively we want to compute
  - $P(w_5|w_1,w_2,w_3,w_4)$
  - =the probability of a word given some previous words
- The model that computes
  - P(W) or
  - $P(w_n|w_1,w_2 \ldots w_{n-1})$
- We can model the word prediction task as the ability to assess the conditional probability of a word given the previous words in the sequence
  - $P(w_n|w_1,w_2 \ldots w_{n-1})$

+ Conditional Probability

- Given an experiment, a corresponding sample space S, and a probability law
- Suppose we know that the outcome is within some given event B
- We want to quantify the likelihood that the outcome also belongs to some other given event A.
- We need a new probability law that gives us the conditional probability of A given B $P(A|B)$

+ Conditional Probability

- Let A and B be events
- p(B|A) = the probability of event B occurring given event A occurs
- Definition: $p(B|A) = p(A \cap B) / p(A)$
- Notation & Notes
  - $p(A,B) = p(A \cap B)$
  - $p(A,B) = P(B,A)$
  - $p(A|B) = p(A \cap B) / p(B)$
  - $p(A,B) = p(A|B) * P(B)$



+ Bayes Theorem

$$P(B \mid A) = \frac{P(A|B)P(B)}{P(A)}$$

- Swap the conditioning
- Sometimes it's easier to estimate one kind of dependence than another

+ Language Modeling

- How might we go about calculating such a conditional probability?
  - One way is to use the definition of conditional probabilities and look for counts. So to get
  - P(*the* | *its water is so transparent that*)

- By definition that's

$$\frac{P(\text{its water is so transparent that the})}{P(\text{its water is so transparent that})}$$

We can get each of those from counts in a large corpus.

+ Very Easy Estimate

- How to estimate?
  - P(the | its water is so transparent that) =

$$\frac{\texttt{Count(its water is so transparent that the)}}{\texttt{Count(its water is so transparent that)}}$$

- According to Google those counts are 5/9
  - Unfortunately... 2 of those were to these slides... So maybe it’s really 3/7
  - In any case, that’s not terribly convincing due to the small numbers involved.
  - (actually, it’s 95,800 / 103,000 or .95)

+ Language Modeling

- Unfortunately, for most sequences and for most text collections we won’t get good estimates from this method.
  - What we’re likely to get is 0. Or worse 0/0.
- Clearly, we’ll have to be a little more clever.
  - Let’s use the chain rule of probability
  - And a particularly useful independence assumption.

+ The Chain Rule

- Recall the definition of conditional probabilities

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$

- Rewriting:

$$P(A \wedge B) = P(A \mid B)P(B)$$

- For sequences...
  - P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)
- In general
  - $P(x_1,x_2,x_3,\ldots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1\ldots x_{n-1})$

+ The Chain Rule

$$
\begin{aligned}
P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) \\
&= \prod_{k=1}^{n} P(w_k|w_1^{k-1})
\end{aligned}
$$

P(its water was so transparent)=

P(its)*

P(water|its)*

P(was|its water)*

P(so|its water was)*

P(transparent|its water was so)

+ Need the Independence Assumption

- There are still a lot of possible sentences
  - In general, we'll never be able to get enough data to compute the statistics for those longer prefixes
  - Same problem we had for the strings themselves
- Make the simplifying assumption
  - P(*the* | *its water is so transparent that*) =

    P(the | that)
- That is, the probability in question is independent of its earlier history.

+ Estimating Bigram Probabilities

- Markov Assumption
  - So for each component in the product replace with the approximation (assuming a prefix of N)

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1})$$

  - Bigram version

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

- The Maximum Likelihood Estimate (MLE)

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Maximum Likelihood Estimates

- The maximum likelihood estimate of some parameter of a model M from a training set T
  - Is the estimate that maximizes the likelihood of the training set T given the model M
- Suppose the word Chinese occurs 400 times in a corpus of a million words (Brown corpus)
- What is the probability that a random word from some other text from the same distribution will be “Chinese”
- MLE estimate is 400/1000000 = .004
  - This may be a bad estimate for some other corpus
- But it is the **estimate** that makes it **most likely** that “Chinese” will occur 400 times in a million word corpus.

+ Berkeley Restaurant Project

- Data collected to create a language model for asking questions about restaurants near Berkeley

Can you tell me about any good cantonese restaurants close by

Mid priced thai food is what i’m looking for

Tell me about chez panisse

Can you give me a listing of the kinds of food that are available

I’m looking for a good place to eat breakfast

When is caffe venezia open during the day

+ Bigram Counts

- Out of 9222 sentences
  - Eg. “I want” occurred 827 times

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

+ Bigram Probabilities

- Divide bigram counts by prefix unigram counts to get probabilities.

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Kinds of Knowledge

- As crude as they are, *N*-gram probabilities capture a range of interesting facts about language.

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

World knowledg

Syntax

Discourse

+ What to count?

- Each word?
- Ums and Uhs?
- Partial words?
- “polywords”? Classes of words?
- What about languages
  - With lots of inflections? (like Russian)
  - With no word boundaries (like Chinese)
  - With lots of compounding (like German)

+ Example from Switchboard

- A.1: Uh, do you have a pet Randy?
- B.2: Uh, yeah, currently we have a poodle.
- A.3: A poodle, miniature or, uh, full size?
- B.4: Yeah, uh, it's, uh miniature.
- A.5: Uh-huh.
- B.6: Yeah.
- A.7: I read somewhere that, the poodles is one of the, the most intelligent dogs, uh, around.
- B.8: Well, um, I wouldn't, uh, I definitely wouldn't dispute that, it, it's actually my wife's dog, uh, I, I became part owner six months ago when we got married, but, uh, it, uh, definitely responds to, uh, to authority and, I've had dogs in the past and, uh, it seems, it seems to, uh, respond real well, it, it - she's, she's picked up a lot of things, uh, just, just by, uh, teaching by force, I guess is what I'd like to say.
- A.9: Oh, uh-huh. So, you, you've only known the dog, wh-, how long did you say.

+ Unknown Words

- But once we start looking at test data, we'll run into words that we haven't seen before (pretty much regardless of how much training data you have.)
- With an Open Vocabulary task
  - Create an unknown word token <UNK>
  - Training of <UNK> probabilities
    - Create a fixed lexicon L, of size V
      - From a dictionary or
      - A subset of terms from the training set
    - At text normalization phase, any training word not in L changed to <UNK>
    - Now we count that like a normal word
  - At test time
    - Use UNK counts for any word not in training