CSCI 5832
Natural Language Processing

Lecture 6
Jim Martin

Today 2/1

- **Review**
- **Noisy Channel Model**
- **Basic Probability Review**
- **Break**
- **N-Gram language models**

Review

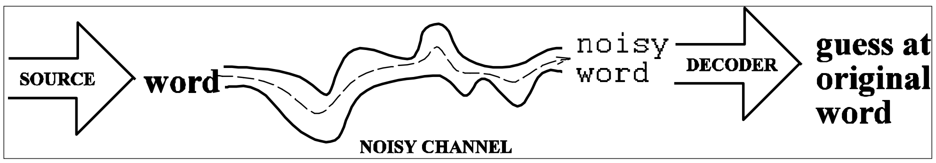
- **FSAs/FSTs can do lots of cool stuff but… they can't do it all.**
  - **In many cases they simply don't have the power to handle the facts (e.g a^n b^n)**
    - **More on this later (e.g. CFGs)**
  - **In the case of global ambiguity, they can't tell us which output is more likely to be the correct one**

So…

- **We'll modify finite state machines so they can tell us more about how likely various (correct) outputs are.**
  - **By applying some simple probability theory**

Noisy Channel

- An influential metaphor in language processing is the noisy channel model



Noisy Channel

- Obvious applications include
  - Speech recognition
  - Optical character recognition
  - Spelling correction
- Not so obvious
  - Semantic analysis
  - Machine translation
    - I.e German to English is a matter of uncorrupting the original signal

Probability Basics

- Prior (or unconditional) probability
  - Written as P(A)
  - For now think of A as a proposition that can turn out to be True or False
  - P(A) is your belief that A is true given that you know nothing else relevant to A
  - In NLP applications, this is the normalized count of some linguistic event
    - Priors for words, NPs, sentences, sentence types, names, etc

Basics

- Conditional (or posterior) probabilities
- Written as P(A|B)
- Pronounced as the probability of A given B
- Think of it as your belief in A given that you know absolutely that B is true.
- In NLP applications this is the count of some event conditioned on some other (usually) linguistic event

And…

- P(A|B)… your belief in A given that you know B is true
- AND B is all you know that is relevant to A

Conditionals Defined

- Conditionals

[illegible]

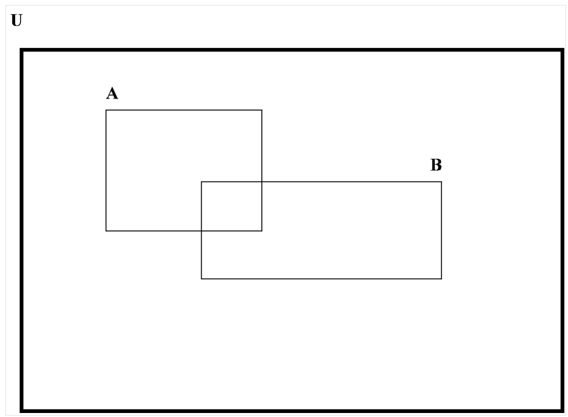
- Rearranging

[illegible]

- And also

[illegible]

Conditionals Defined



Bayes

- **We know…**

[illegible]

[illegible]

[illegible]

- **So rearranging things**

[illegible]

[illegible]

Bayes

- **Memorize this**

Bayes and the Noisy Channel

- **In applying Bayes to the noisy channel we want to compute the most likely source given some observed (corrupt) output signal**

  $\text{Argmax}_i\ P(\text{Source}_i|\text{Signal})$

- **Often (not always) this is hard to get, so we apply Bayes**

Bayes and Noisy Channel

- So… argmax this instead

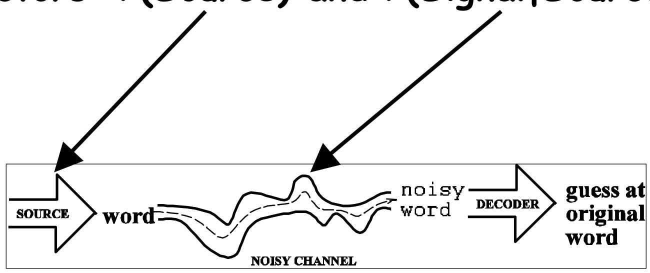
[illegible]

Argmax and Bayes

- What does this mean?
  - Argmax [illegible]
- Plug in each possible source and compute the corresponding probability. Pick the one with the highest
- Note the denominator is the same for each source candidate so we can ignore it for the purposes of the argmax.

Argmax and Bayes

- Ignoring the denominator leaves us with two factors: P(Source) and P(Signal|Source)



Bayesian Decoding

- P(Source): This is often referred to as a language model. It encodes information about the likelihood of particular sequences (or structures) independent of the observed signal.
- P(Signal | Source): This encodes specific information about how the channel tends to introduce noise. How likely is it that a given source would produce an observed signal.

Note

- **This framework is completely general; it makes minimal assumptions about the nature of the application, the source, or the channel.**

Transition

- **Up to this point we've mostly been discussing words in isolation (and their insides)**
- **Now we'll switch to looking at sequences of words**
- **And we're going to worry about *assigning probabilities to sequences of words***

Who Cares?

- **Why would you want to assign a probability to a sentence or…**
- **Why would you want to predict the next word…**
- **Lots of applications**
  - **Historically it was first used effectively in automatic speech recognition**

Break

- **Quiz will be 2/8**
  - **Focus on 2,3,4, maybe the start of 5**
- **Review past quizzes**
  - **Question relate to lectures, readings and the assignment**
  - **Yes, even stuff in the readings not covered in class**
- **HW 2 to be posted asap**

Chain Rule

- Recall the definition of conditional probabilities [illegible]

- Rewriting [illegible]

- Or… [illegible]
- Or… [illegible]

Example

- The big red dog

- P(The)*P(big|the)*P(red|the big)*P(dog|the big red)

- Better P(The| <Beginning of sentence>) written as
  P(The | <S>)

General Case

- The word sequence from position 1 to n is [illegible]
- So the probability of a sequence is

[illegible]

[illegible]

Unfortunately

- That doesn't help since its unlikely we'll ever gather the right statistics for the prefixes.

Markov Assumption

- Assume that the entire prefix history isn't necessary.
- In other words, an event doesn't depend on all of its history, just a fixed length near history

Markov Assumption

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

[illegible]

N-Grams
The big red dog

- **Unigrams: P(dog)**
- **Bigrams: P(dog|red)**
- **Trigrams: P(dog|big red)**
- **Four-grams: P(dog|the big red)**

In general, we'll be dealing with P(Word| Some fixed prefix)

Caveat

- **The formulation P(Word| Some fixed prefix) is not really appropriate in many applications.**
- **It is if we're dealing with real time speech where we only have access to prefixes.**
- **But if we're dealing with text we already have the right and left contexts. There's no a priori reason to stick to left contexts.**

BERP Table: 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

Counts/Bigram Probs

- **Recall… if we want P(want | I) that's the**

P(I want)/P(want) and that's just

Count(I want)/Count(want)

BERP Table: Bigram Probabilities

	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

Some Observations

- **The following numbers are very informative. Think about what they capture.**
  - **P(want|I) = .32**
  - **P(to|want) = .65**
  - **P(eat|to) = .26**
  - **P(food|Chinese) = .56**
  - **P(lunch|eat) = .055**

Some More Observations

- P(I | I)
- P(I | want)
- P(I | food)

- I I I want
- I want I want to
- The food I want is

Generation

- Choose N-Grams according to their probabilities and string them together

BERP

- **I want**
  want to
  to eat
  eat Chinese
  Chinese food
  food .

Shakespeare: Unigrams

- To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
- Every enter now severally so, let
- Hill he late speaks; or! a more to leg less first you enter
- Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Shakespeare: Bigrams

- What means, sir. I confess she? then all sorts, he is trim, captain.
- Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
- What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?
- Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt

Shakespeare: Trigrams

- Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
- This shall forbid it should be branded, if renown made it empty.
- Indeed the duke; and had a very good friend.
- Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Shakespeare: 4-Grams

- King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
- Will you not tell me who I am?
- It cannot be but so.
- Indeed the short and the long. Marry, 'tis a noble Lepidus.

WSJ: Bigrams

bigram: Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Some Useful Observations

- A small number of events occur with high frequency
  - You can collect reliable statistics on these events with relatively small samples
  - Generally you should believe these numbers
- A large number of events occur with small frequency
  - You might have to wait a long time to gather statistics on the low frequency events
  - You should treat these numbers with skepticism

Some Useful Observations

- Some zeroes are really zeroes
  - Meaning that they represent events that can’t or shouldn’t occur
- On the other hand, some zeroes aren’t really zeroes
  - They represent low frequency events that simply didn’t occur in the corpus

An Aside on Logs

- **You don't really do all those multiplications. They're expensive to do (relatively), the numbers are too small, and they lead to underflows.**
- **Convert the probabilities to logs and then do additions.**
- **To get the real probability (if you need it) go back to the antilog.**

Problem

- **Let's assume we're using N-grams**
- **How can we assign a probability to a sequence where one of the component n-grams has a value of zero**
- **Assume all the words are known and have been seen**
  - **Go to a lower order n-gram**
  - **Back off from bigrams to unigrams**
  - **Replace the zero with something else**

Smoothing Solutions

- **Lots of solutions… All based on different intuitions about how to think about events that haven't occurred (yet).**
- **They range from the very simple to very convoluted. We'll cover**
  - **Add 1**
  - **Good-Turing**

Add-One (Laplace)

- **Make the zero counts 1.**
- **Rationale: They're just events you haven't seen yet. If you had seen them, chances are you would only have seen them once… so make the count equal to 1.**
- **Caveat: Other than the name there's no reason to add 1, you can just as easily add some other fixed amount.**

Original BERP 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

Add-One Smoothed BERP Reconstituted

	I	want	to	eat	Chinese	food	lunch
I	6	740	.68	10	.68	.68	.68
want	2	.42	331	.42	3	4	3
to	3	.69	8	594	3	.69	9
eat	.37	.37	1	.37	7.4	1	20
Chinese	.36	.12	.12	.12	.12	15	.24
food	10	.48	9	.48	.48	.48	.48
lunch	1.1	.22	.22	.22	.22	.44	.22

Huh?

- The P(to | I) was 0 since "I to" never happened.
- Now we added 1 to Count("I to") so its probability is what?

  Count("I to")/(Count("I")+N) =

  1/(Count("I")+N)
- Now we know its probability and the sample size we can compute the number of times it should have occurred in the corpus

Add-One Comments

- Pros
  - Easy
- Cons
  - Doesn't work very well.
  - Technical: Moves too much of the probability mass to the zero events and away from the events that actually occurred.
  - Intuitive: Makes too many of the zeroes too big, making the things that occurred look less likely than they really are.

Next Time

- **More smoothing (Good-Turning) and back-off**
- **Start on part-of-speech tagging (Chapter 5)**