# CS-AD 220 – Spring 2016

# Natural Language Processing

## Session 9: 25-Feb-16

Prof. Nizar Habash

Some slides are adapted from Jurafsky and Martin's course slides on Speech and Language Processing and from Jurafsky Slides for LSA Summer 2007

**NYUAD CS-AD 220 – Spring 2016**
**Natural Language Processing**

**Assignment #2**
**Finite State Machines**
**Assigned Feb 18, 2016**
**Due Mar 10, 2016 (11:59pm)**

## I. Grading & Submission

This assignment is about the development of finite state machines using the OpenFST and Thrax toolkits. The assignment accounts for 15% of the full grade. It consists of three exercises. The first is a simple "machine translation" system for animal sounds to help with learning the tools. The second is about modeling how numbers are read in English and French. And the third is about Spanish verb conjugation. The answers should be placed in a zipped folder with separate sub-directories for each exercise.

The assignment is due on March 10 before midnight (11:59pm). For late submissions, 10% will be deducted from the homework grade for any portion of each late day. The student should upload the answers in a single zipped to NYU Classes (Assignment #2).

*Assignment #2 posted on NYU Classes*

# Moving Legislative Day Class

- Spring Break is March 18 – 25, 2016
- Sat March 26, 2016 is a Legislative *Thursday*
- Move to

**Sat April 2, 2016 at 10am**

**Same Classroom C2-E049**

# Today

- N-Grams

… but first

# Introduction to Probability

- Experiment (trial)
  - Repeatable procedure with well-defined possible outcomes
- Sample Space (S)
  - the set of all possible outcomes
  - *finite or infinite*
  - Example
    - coin toss experiment
    - possible outcomes: S = {heads, tails}
  - Example
    - die toss experiment
    - possible outcomes: S = {1,2,3,4,5,6}

Slides from Sandiway Fong

# Introduction to Probability

- Definition of sample space depends on what we are asking
  - Sample Space (S): the set of all possible outcomes
  - Example
    - die toss experiment for whether the number is even or odd
    - possible outcomes: {even,odd}
    - *not* {1,2,3,4,5,6}

# More definitions

- Events
  - an ***event*** is any subset of outcomes from the ***sample space***
- Example
  - die toss experiment
  - let A represent the event such that the outcome of the die toss experiment is divisible by 3
  - A = {3,6}
  - A is a subset of the sample space S= {1,2,3,4,5,6}
- Example
  - Draw a card from a deck
    - suppose sample space S = {heart,spade,club,diamond} (*four suits*)
  - let A represent the event of drawing a heart
  - let B represent the event of drawing a red card
  - A = {heart}
  - B = {heart,diamond}

# Introduction to Probability

- Some definitions
  - Counting
    - suppose operation $o_i$ can be performed in $n_i$ ways, then
    - a sequence of k operations $o_1 o_2 ... o_k$
    - can be performed in $n_1 \times n_2 \times ... \times n_k$ ways
  - Example
    - die toss experiment, 6 possible outcomes
    - two dice are thrown at the same time
    - number of sample points in sample space = $6 \times 6 = 36$

# Definition of Probability

- The probability law assigns to an event a nonnegative number
- Called P(A)
- Also called the probability of A
- That encodes our knowledge or belief about the collective likelihood of all the elements of A
- Probability law must satisfy certain properties

# Probability Axioms

- Nonnegativity
  - P(A) >= 0, for every event A
- Additivity
  - If A and B are two disjoint events, then the probability of their union satisfies:
  - P(A U B) = P(A) + P(B)
- Normalization
  - The probability of the entire sample space S is equal to 1, I.e. P(S) = 1.

# An example

- An experiment involving a single coin toss
- There are two possible outcomes, H and T
- Sample space S is {H,T}
- If coin is fair, should assign equal probabilities to 2 outcomes
- Since they have to sum to 1
- P({H}) = 0.5
- P({T}) = 0.5
- P({H,T}) = P({H})+P({T}) = 1.0

# Another example

- Experiment involving 3 coin tosses
- Outcome is a 3-long string of H or T
- S ={HHH,HHT,HTH,HTT,THH,THT,TTH,TTT}
- Assume each outcome is equiprobable
  - "Uniform distribution"
- What is probability of the event that exactly 2 heads occur?
- A = {HHT,HTH,THH}
- P(A) = P({HHT})+P({HTH})+P({THH})
- = 1/8 + 1/8 + 1/8
- =3/8

# Probability definitions

- In summary:

$$P(E) = \frac{\text{number of outcomes corresponding to event E}}{\text{total number of outcomes}}$$

Probability of drawing a spade from 52 well-shuffled playing cards:

$$\frac{13}{52} = \frac{1}{4} = 0.25$$

# Probabilities of two events

- If two events A and B are independent
- Then
  - P(A and B) = P(A) x P(B)

- Flip two fair coins
  - What is the probability that they are both heads?
- Draw a card from a deck, then put it back, draw a card from the deck again
  - What is the probability that both drawn cards are hearts?
- A coin is flipped twice
  - What is the probability that it comes up heads both times?

# How about non-uniform probabilities? An example

- A biased coin,
  - twice as likely to come up tails as heads,
  - is tossed twice
- What is the probability that at least one head occurs?
- Sample space = {hh, ht, th, tt} (h = heads, t = tails)
- Sample points/probability for the event:
  - ht 1/3 x 2/3 = **2/9**
  - th 2/3 x 1/3 = **2/9**

  hh 1/3 x 1/3= **1/9**

  tt 2/3 x 2/3 = 4/9
- Answer: 5/9 = ≈0.56 (*sum of weights in **red***)

# Moving toward language

- What's the probability of drawing a 2 from a deck of 52 cards with four 2s?

$$P(drawing\ \ a\ \ two) = \frac{4}{52} = \frac{1}{13} = .077$$

- What's the probability of a random word (from a random dictionary page) being a verb?

$$P(drawing\ \ a\ \ verb) = \frac{\#\,of\ \ ways\ \ to\ \ get\ \ a\ \ verb}{all\ \ words}$$

# Probability and part of speech tags

- What's the probability of a random word (from a random dictionary page) being a verb?

$$P(drawing\ \ a\ \ verb) = \frac{\#\,of\ \ ways\ \ to\ \ get\ \ a\ \ verb}{all\ \ words}$$

- How to compute each of these
- All words = just count all the words in the dictionary
- # of ways to get a verb: number of words which are verbs!
- If a dictionary has 50,000 entries, and 10,000 are verbs…. P(V) is 10000/50000 = 1/5 = .20

# Conditional Probability

- A way to reason about the outcome of an experiment based on partial information
  - In a word guessing game the first letter for the word is a “t”. What is the likelihood that the second letter is an “h”?
  - How likely is it that a person has a disease given that a medical test was negative?
  - A spot shows up on a radar screen. How likely is it that it corresponds to an aircraft?

# More precisely

- 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)

# An intuition

- A is “it’s raining now”.
- P(A) in Abu Dhabi is .01
- B is “it was raining ten minutes ago”

- P(A|B) means “what is the probability of it raining now if it was raining 10 minutes ago”
- P(A|B) is probably way higher than P(A)
- Perhaps P(A|B) is .10

- Intuition: The knowledge about B should change our estimate of the probability of A.

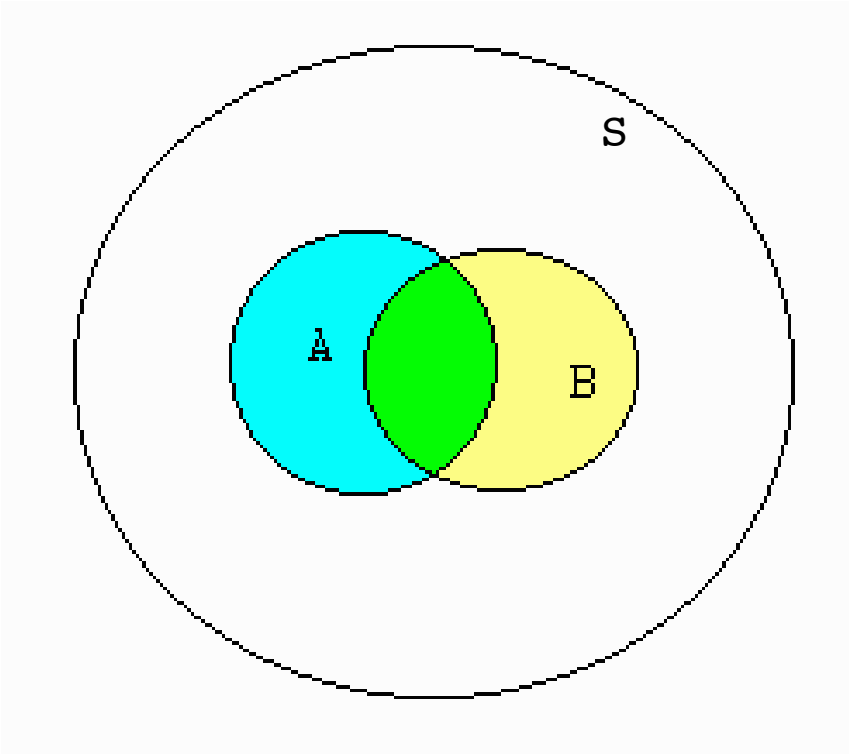
# Conditional probability

- One of the following 30 items is chosen at random
- What is P(X), the probability that it is an X?
- What is P(X|red), the probability that it is an X given that it is red?

| | | | | | |
|---|---|---|---|---|---|
| O | X | X | X | O | O |
| O | X | X | O | X | O |
| O | O | O | X | O | X |
| O | O | O | O | X | O |
| O | X | X | X | X | O |

# Conditional Probability

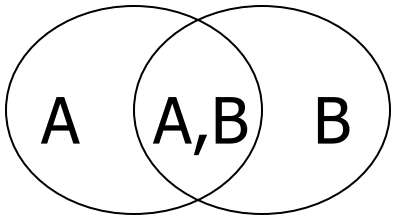
- let A and B be events
- p(B|A) = the *probability* of event B *occurring given* event A *occurs*
- *definition:* p(B|A) = p(A ∩ B) / p(A)



# Conditional probability

- P(A|B) = P(A ∩ B)/P(B)
- Or

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




*Note: P(A,B)=P(A|B) · P(B)*
*Also: P(A,B) = P(B,A)*

# Independence

- What is P(A,B) if A and B are independent?
- P(A,B)=P(A) · P(B) iff A,B independent.

P(heads,tails) = P(heads) · P(tails) = .5 · .5 = .25

*Note: P(A|B)=P(A) iff A,B independent*
*Also: P(B|A)=P(B) iff A,B independent*

Thomas Bayes
18th century English clergyman

# Bayes Theorem

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

- Swap the conditioning
- Sometimes easier to estimate one kind of dependence than the other

# Deriving Bayes Rule

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

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

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

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

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

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

# Word Prediction

- Guess the next word...
  - *... I notice three guys standing on the ???*
- There are many sources of knowledge that can be used to inform this task, including arbitrary world knowledge.
- But it turns out that you can do pretty well by simply looking at the preceding words and keeping track of some fairly simple counts.

# Word Prediction

- We can formalize this task using what are called *N*-gram models.
- *N*-grams are token sequences of length *N*.
- Our earlier example contains the following 2-grams (aka bigrams)
  - (I notice), (notice three), (three guys), (guys standing), (standing on), (on the)
- Given knowledge of counts of N-grams such as these, we can guess likely next words in a sequence.

# *N*-Gram Models

- More formally, we can use knowledge of the counts of *N*-grams to assess the conditional probability of candidate words as the next word in a sequence.
- Or, we can use them to assess the probability of an entire sequence of words.
  - Pretty much the same thing as we'll see...

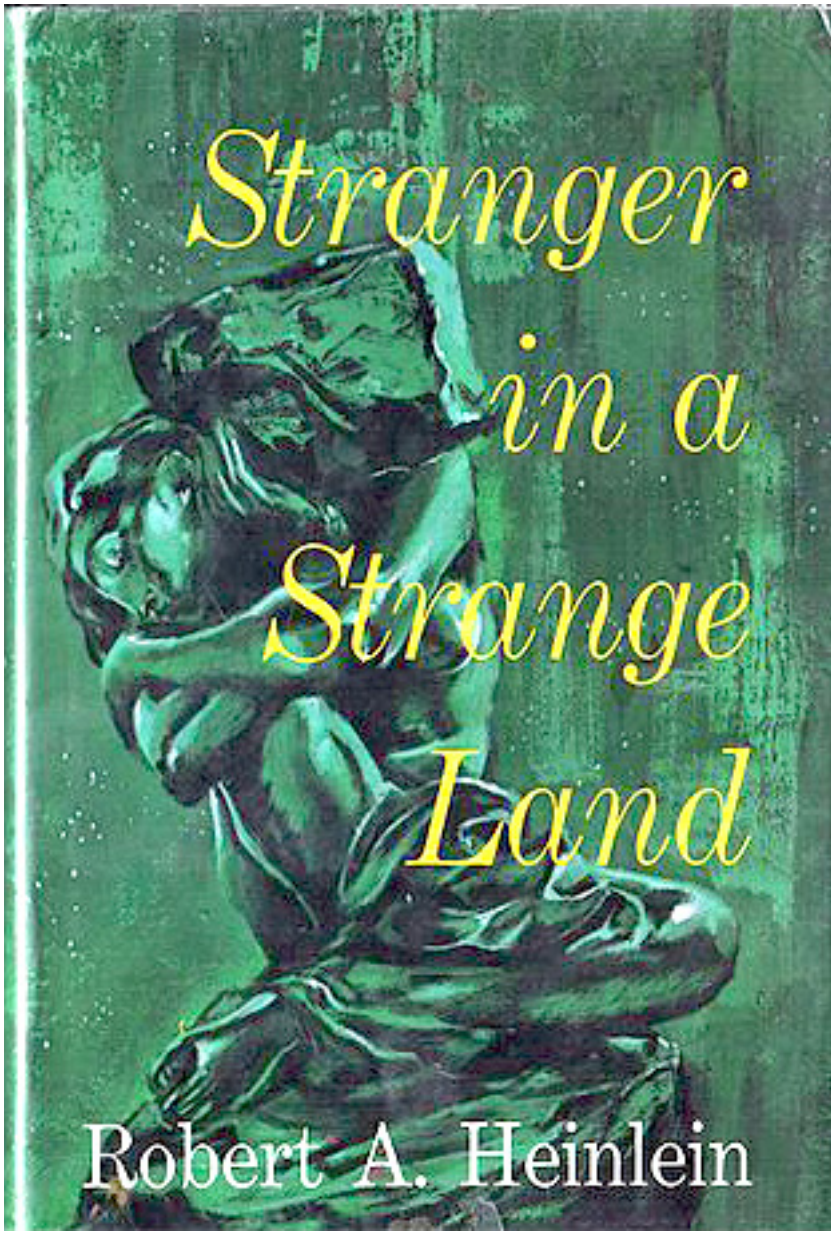
# Applications

- It turns out that being able to predict the next word (or any linguistic unit) in a sequence is an extremely useful thing to be able to do.
- As we'll see, it lies at the core of the following applications
  - Automatic speech recognition
  - Handwriting and character recognition
  - Spelling correction
  - Machine translation
  - And many more.

# Counting

- Simple counting lies at the core of any probabilistic approach. So let's first take a look at what we're counting.
  - *He stepped out into the hall, was delighted to encounter a water brother.*
    - 13 tokens, 15 if we include "," and "." as separate tokens.
    - Assuming we include the comma and period, how many bigrams are there?

- Water Brother???



# Counting

- Not always that simple
  - *I do uh main- mainly business data processing*
- Spoken language poses various challenges.
  - Should we count "uh" and other fillers as tokens?
  - What about the repetition of "mainly"? Should such do-overs count twice or just once?
  - The answers depend on the application.
    - If we're focusing on something like ASR to support indexing for search, then "uh" isn't helpful (it's not likely to occur as a query).
    - But filled pauses are very useful in dialog management, so we might want them there.

# Counting: Types and Tokens

- How about
  - *They picnicked by the pool, then lay back on the grass and looked at the stars.*
    - 18 tokens (again counting punctuation)
- But we might also note that "*the*" is used 3 times, so there are only 16 unique types (as opposed to tokens).
- In going forward, we'll have occasion to focus on counting both types and tokens of both words and *N*-grams.

# Counting: Wordforms

- Should “cats” and “cat” count as the same when we’re counting?
- How about “geese” and “goose”?
- Remember…
  - Lemma: a set of lexical forms having the same stem, major part of speech, and rough word sense
  - Wordform: fully inflected surface form
- Again, we’ll have occasion to count both lemmas and wordforms

# Counting: Corpora

- So what happens when we look at large bodies of text instead of single utterances?
- Brown et al (1992) large corpus of English text
  - 583 million wordform tokens
  - 293,181 wordform types
- Google
  - Crawl of 1,024,908,267,229 English tokens
  - 13,588,391 wordform types
    - That seems like a lot of types... After all, even large dictionaries of English have only around 500k types. Why so many here?

- Numbers
- Misspellings
- Names
- Acronyms
- etc

# Language Modeling

- Back to word prediction
- 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
  - $P(w_n|w_1,w_2...w_{n-1})$
- We'll call a statistical model that can assess this a *Language Model*

# 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

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

P(its water is so transparent that the)

P(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)

P(the | its water is so transparent that) =

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

- *According to Google those counts are 25/38.*

# 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
- Rewriting:

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

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

- 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

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

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)

# Unfortunately

- 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

# Independence Assumption

- Make the simplifying assumption
  - P(lizard| the,other,day,I,was,walking,along,and,saw,a) = P(lizard|a)
- Or maybe
  - P(lizard| the,other,day,I,was,walking,along,and,saw,a) = P(lizard|saw,a)
- That is, the probability in question is independent of its earlier history.

# Independence Assumption

- This particular kind of independence assumption is called a ***Markov*** *assumption* after the Russian mathematician Andrei Markov.

Andrey Andreyevich Markov
Russian Mathematician (1856-1922)

# 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

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

# Estimating Bigram Probabilities

- The Maximum Likelihood Estimate (MLE)

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

# An Example

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

$P(\text{I}\,|\,\text{<s>}) =$ $P(\text{Sam}\,|\,\text{<s>}) =$ $P(\text{am}\,|\,\text{I}) =$

$P(\text{</s>}\,|\,\text{Sam}) =$ $P(\text{Sam}\,|\,\text{am}) =$ $P(\text{do}\,|\,\text{I}) =$

# An Example

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

$$P(\text{I}\,|\,\text{<s>}) = \tfrac{2}{3} = .67 \qquad P(\text{Sam}\,|\,\text{<s>}) = \tfrac{1}{3} = .33 \qquad P(\text{am}\,|\,\text{I}) = \tfrac{2}{3} = .67$$

$$P(\text{</s>}\,|\,\text{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\text{Sam}\,|\,\text{am}) = \tfrac{1}{2} = .5 \qquad P(\text{do}\,|\,\text{I}) = \tfrac{1}{3} = .33$$

# An Example

- <s> I am Sam </s>
- <s> Sam I am </s>
- <s> I do not like green eggs and ham </s>

$$P(\text{I}\,|\,\text{<s>}) = \tfrac{2}{3} = .67 \qquad P(\text{Sam}\,|\,\text{<s>}) = \tfrac{1}{3} = .33 \qquad P(\text{am}\,|\,\text{I}) = \tfrac{2}{3} = .67$$
$$P(\text{</s>}\,|\,\text{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\text{Sam}\,|\,\text{am}) = \tfrac{1}{2} = .5 \qquad P(\text{do}\,|\,\text{I}) = \tfrac{1}{3} = .33$$

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

# 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 Sentences

**BeRP: The Berkeley Restaurant Project (Jurafsky et al., 1994)**

- Examples
  - *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 | 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 |

- P(want | i) = 0.33

# Bigram Estimates of Sentence Probabilities

- P(<s> I want english food </s>) =
  P(i|<s>)*
  P(want|I)*
  P(english|want)*
  P(food|english)*
  P(</s>|food)*
  =.000031

# Bigram Estimates of Sentence Probabilities

- P(<s> I want english food </s>) =
  P(i|<s>)*
  P(want|I)*
  P(english|want)*
  P(food|english)*
  P(</s>|food)*
  =.000031

| | |
|---|---|
| P(i\|<s>) | =0.25 |
| P(want\|i) | =0.33 |
| P(english\|want) | =0.0011 |
| P(food\|english) | =0.5 |
| P(</s>\|food) | =0.68 |

# 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 knowledge

Syntax

Discourse

# Next Time

- Read J+M Chap 4 (4.5 to 4.9)

- Assignment #2 due March 10 before midnight
  - Start early!