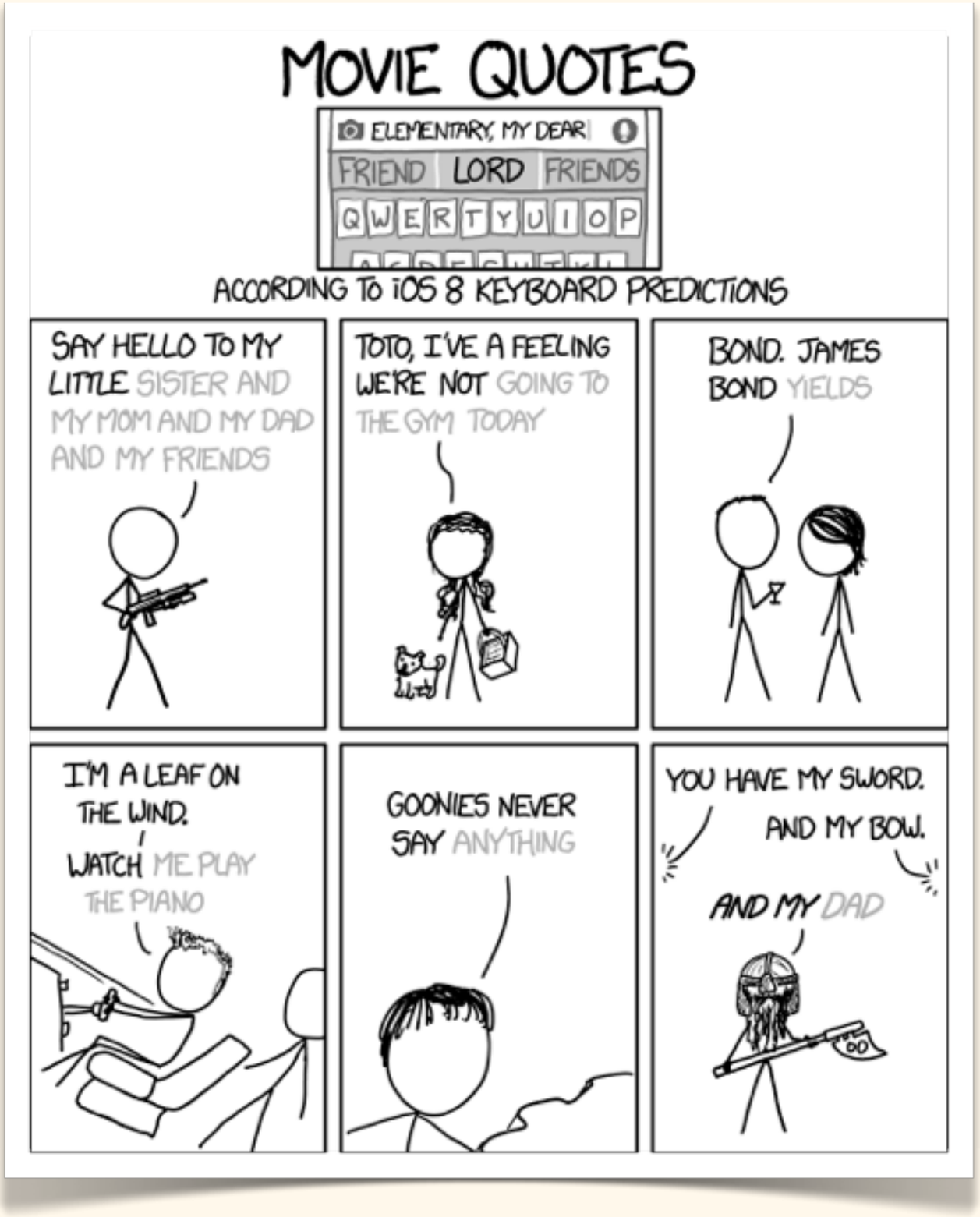
Language Models, part 1



Steven Bedrick

CS/EE 5/655, 10/13/14

http://www.openfst.org/twiki/pub/FST/FstHltTutorial/tutorial_part3.pdf

Plan for the day:

1. Modeling context
2. Probability
3. Language Models
4. Markov Chains
5. Implementation Notes

Complete this sentence:

“I was late assigning the...

Complete this sentence:

"I was late assigning the *homework*."

"I was late assigning the *purple*."

p(purple) < p(homework)

Observation: Language is not random!

Of course, many sequential processes are not totally random:

Today's weather has something to do with yesterday's...

... each speech audio sample is related to its predecessor, etc.

There's all kinds of structure to protein sequences!

What might we want to do with this information?

Prediction:
Given an observation(s), what is the most likely next observation?

Estimation:
Given an observation(s), how likely is it to have come from whatever process was generating our data?

Classification:
Given a new observation(s), and several sets of old observations, which set did the new ones come from?

Prediction:

"I was late assigning the *homework*."

"I was late assigning the *purple*."

p(purple) > p(homework)

Estimation:

s1 “He really believed, that were it not for the inferiority of her connections...”

s2 "that not it her really of, were connections for inferiority He believed the"

p(s1) > p(s2)

Less silly example:

he briefed to reporters on the chief contents of the statement

he briefed reporters on the chief contents of the statement

he briefed to reporters on the main contents of the statement

he briefed reporters on the main contents of the statement

Which sentence to choose?

Another example:

They are leaving in fifteen minuets to go to the store.

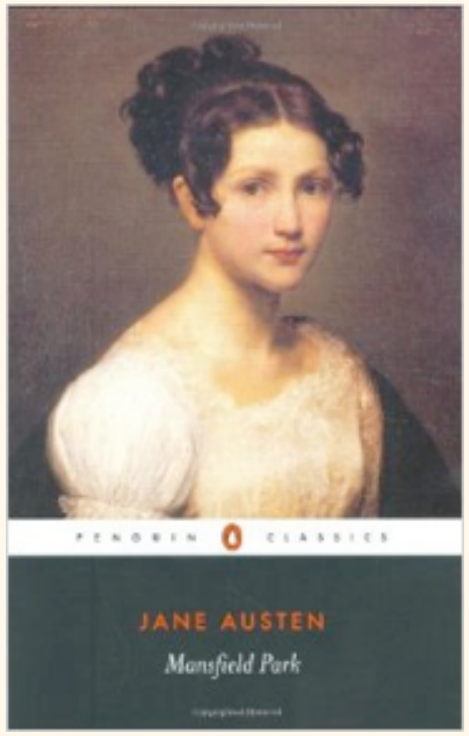
The design an construction will take more than two years.

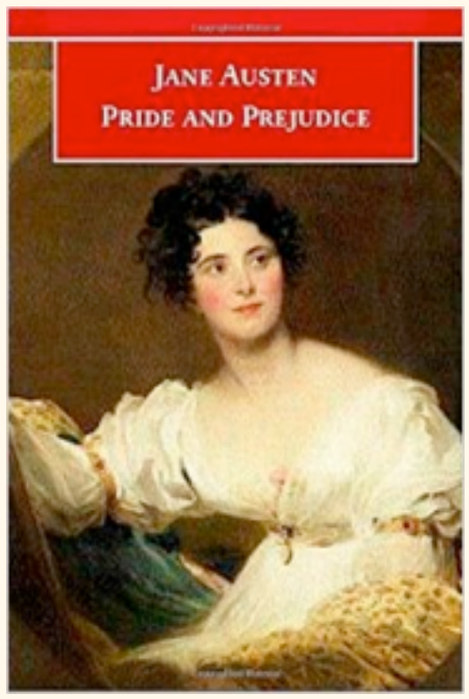
How to spot typos that are valid English words?

Classification:

“About thirty years ago Miss Maria Ward, of Huntingdon, with only seven thousand pounds, had the good luck to captivate...”

“It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.”





Which sentence came from which book?

All of these depend on having some kind of *model* of our data...

... and then formulating our problems as questions that can be answered by the model.

Plan for the day:

1. Modeling context
2. Probability
3. Language Models
4. Markov Chains
5. Implementation Notes

Probability:

Much of what we want to do depends on calculating the probability of an observation given some history:

$$P(obs \mid hist)$$

Probability:

$P(obs \mid hist)$

Consider: “its water is so transparent that”

How likely is “the” to be the next word?

$P(w \mid h)$

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

Probability:

P(w | hist)

$$P(w|h) = \frac{C(w,h)}{C(h)}$$

$$P(w|h) = \frac{C(its\ water\ is\ so\ transparent\ that\ the)}{C(its\ water\ is\ so\ transparent\ that)}$$

This “works”... but makes a lot of assumptions.

Probability:

$$P(w|h) = \frac{C(its\ water\ is\ so\ transparent\ that\ the)}{C(its\ water\ is\ so\ transparent\ that)}$$

This “works”... but makes a lot of assumptions.

1. That’s a lot of counting...
2. Language is productive and creative!
3. We’d need a very large corpus (infinite?)

Probability:

A better way:

$P(its = w_1, water = w_2, is = w_3, so = w_4, transparent = w_5, that = w_6)$

$P(w_1, w_2, w_3, w_4, w_5, w_6)$

$w_1 ... w_n$

w_1^n w_1^3 "The first three words in the sequence"

$$P(X_1, ...X_n) = P(X_1)P(X_2|X_1)(PX_3|X_1^2)...P(X_n|X_1^{n-1})$$

$$P(X_1, ...X_n) = \prod_{k=1}^{n} P(X_k|X_1^{k-1})$$

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

Probability:

A better way:

$$P(its = w_1, water = w_2, is = w_3, so = w_4, transparent = w_5, that = w_6)$$

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

This doesn't look much better— we still need to be able to calculate all the intermediate probabilities...

What if we make an assumption?

Instead of the *entire* history for each word, maybe we only need a little bit?

Probability:

Instead of the *entire* history for each word, maybe we only need a little bit?

A bigram model: each word depends only on its preceding word.

Andrei Andreievich Markov
1856–1922

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

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

Probability:

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

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

How many times did we see this word combo...

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

How many times did we see a bigram that started with our bigram's first word?

Probability:

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

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

How many times did we see this word combo...

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

How many times did we see a bigram that started with our bigram's first word?

Fun example:

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

$$P(I|<s>) = \frac{C(<s> \ I)}{C(<s>)} = \frac{2}{3} = 0.67$$

$$P(Sam|am) = \frac{1}{2} = 0.5$$

Example from J&M Chap. 4

A bigger toy corpus:

	i	want	to	eat	chinese	food	lunch	spend
i	5	927	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

Example from J&M Chap. 4

A bigger toy corpus:

	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

Example from J&M Chap. 4

A bigger toy corpus:

	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(i | <s>) = 0.25

p(food | english) = 0.5

p(english | want) = 0.0011

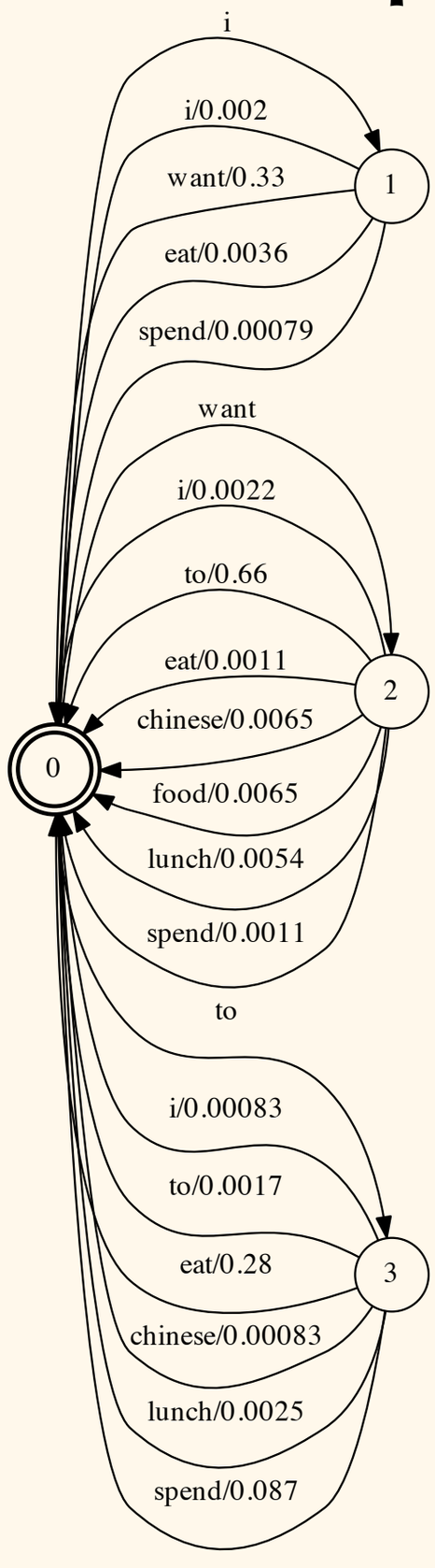
p(</s> | food) = 0.68

P(<s> i want english food </s>)

$$
\begin{aligned}
&= P(i|<s>)P(want|i)...P(</s>|food) \\
&= 0.25 \times 0.33 \times 0.0011 \times 0.5 \times 0.68 \\
&= 0.000031
\end{aligned}
$$

Example from J&M Chap. 4

We can represent this as an FST...



More on this Wednesday...

A bigger toy corpus:

	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(i | <s>) = 0.25

p(food | english) = 0.5

p(english | want) = 0.0011

p(</s> | food) = 0.68

P(<s> i want english food </s>)

$$= P(i| < s >)P(want|i)...P(< /s > |food)$$
$$= 0.25 \times 0.33 \times 0.0011 \times 0.5 \times 0.68$$
$$= 0.000031$$

Example from J&M Chap. 4

Open questions:

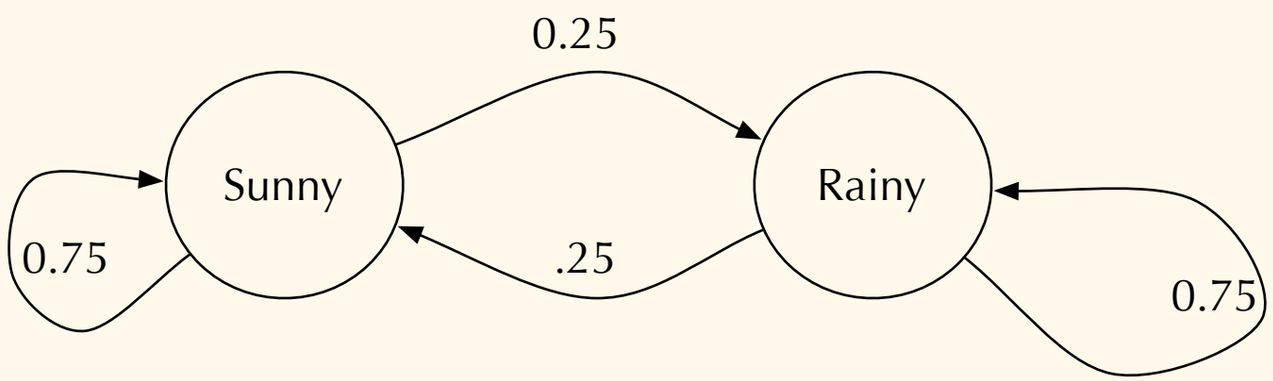
1. We can’t do anything useful with bigrams we’ve never seen before...
2. Our probabilities are tiny!
3. Do bigrams capture enough context to be useful?

Useful illustration: Markov Chain

A Markov Chain is a *memoryless* mathematical system, similar to a wFSA.

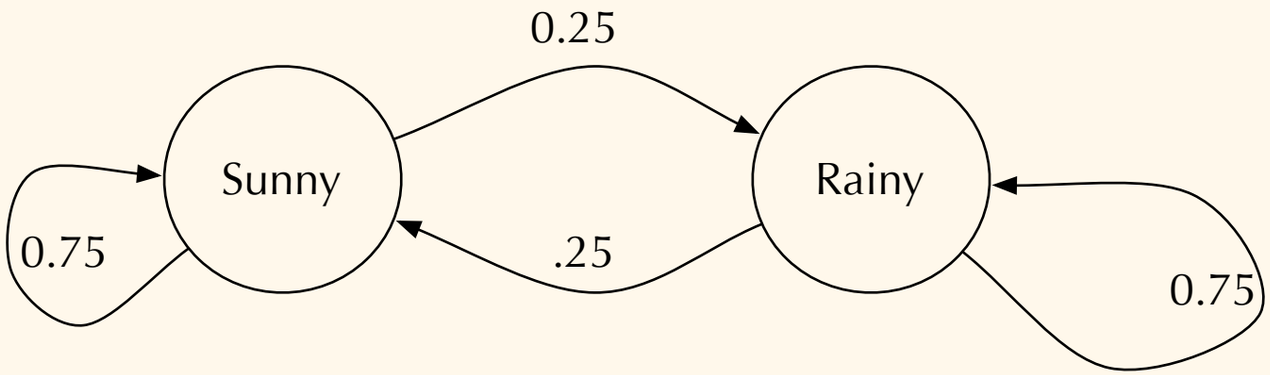
Consider the weather:

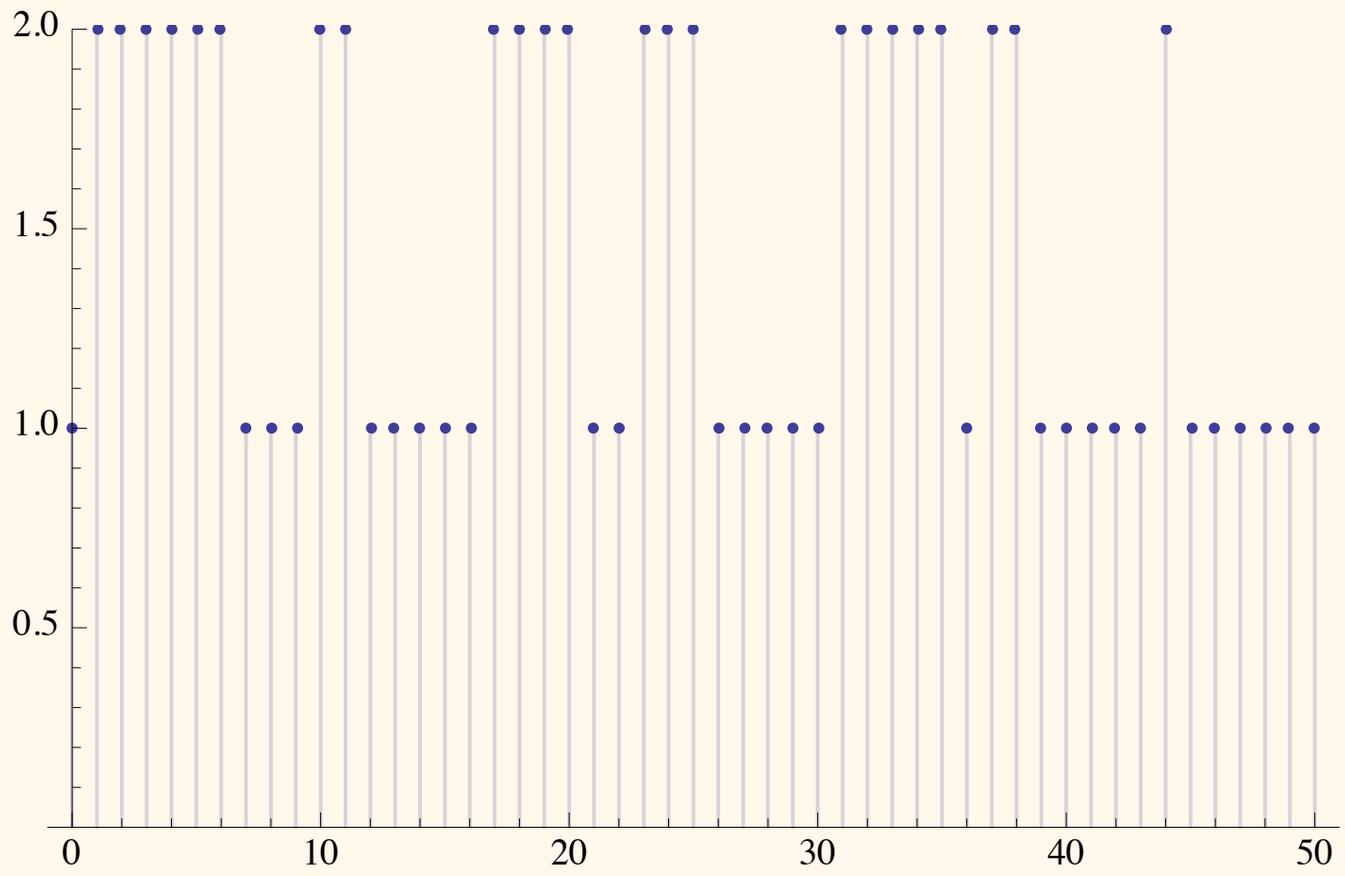
Today's weather is usually a good predictor of tomorrow's:



In Portland, if today was rainy, tomorrow has a 75% chance of the same.

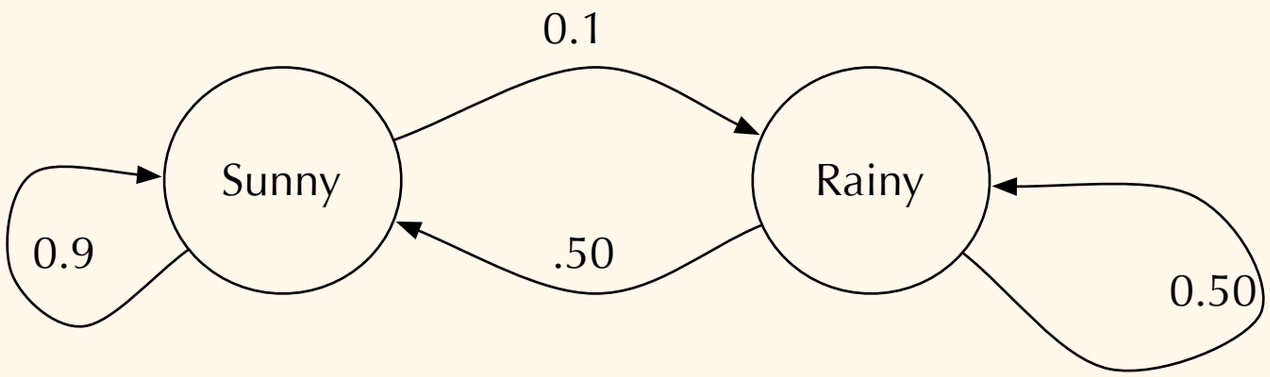
Useful illustration: Markov Chain



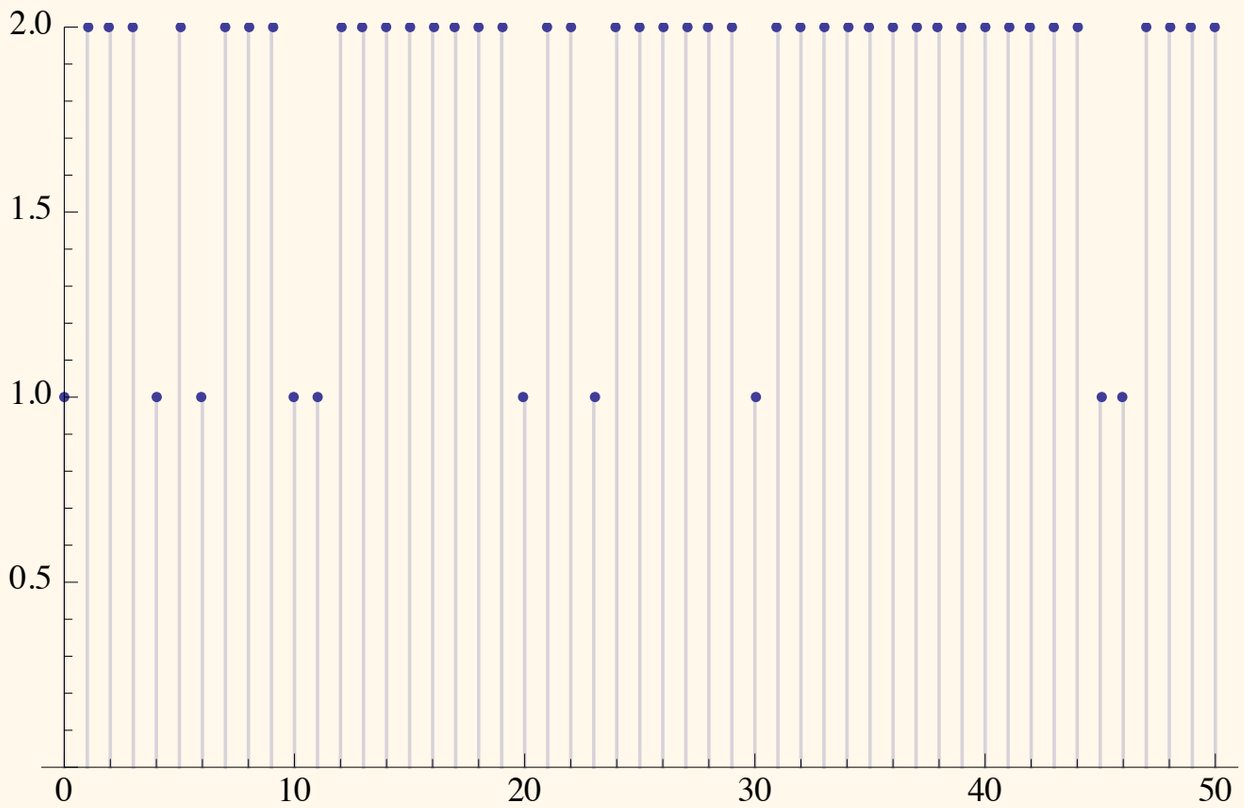


In Portland, if today was rainy, tomorrow has a 75% chance of the same.

Useful illustration: Markov Chain



In Los Angeles, sun is far more common.



Markov chains have lots of uses.

PageRank, for example!

Things we can ask Markov Chains:

Most common state(s)?

Most common state(s) given specific starting point?

Expected amount of time spent in each state?

How likely is a given state sequence?

We can also use them to generate pseudo-random text...

... which gets us back to our question:

1. We can't do anything useful with bigrams we've never seen before...
2. Our probabilities are tiny!
3. *Do bigrams capture enough context to be useful?*

Random Markov-order-1 (unigram) text:

. charge he -- not Darcy and `` in is . is home justice your to can and the gratitude even more question Wickham at tidings get aware of . can the , in Mrs. I moved know to partiality a Such though be however be ! have in seen comparison you , of surprise the up believes laid very

'' out not had fifteen . '' . father . spoke and from soon determined everybody , kind weighed to dear , heard , of long word ask first But was Kitty , , whole is stopped up is

! her had was be such be is disposition be spirits you of for Jane that above account her people visit really of to doing , had Miss from And and him tempt all I should you you seeing rest

-- `` him sorrow they sentiment not you in it says the Nothing attachment visit us '' uneasy '' in on that ! business that . be than which to in of as Allow but one My no the , ; made had and might her noticed the him been represented not after he I '' or that through Elizabeth could him , preservation on merely , ? . however and , late now of the

awkward downstairs and so of ! do to She . Jane A the amazing beauty , , receive

Random Markov-order-2 (bigram) text:

It was no man who knew her way -- poor regiment , had once occurred to find me if you the same expression of this happy prospect of her with a marriage .

At any conception of your cousin , and then changed colour .

And you look somewhat better acquainted with her that money which she can not betrayed him , or views of the town ; and despise me ? ''

Though Elizabeth and tell my vanity is a comfortable house , Lizzy ; but the free from a brother Gardiner left under frequent as a way , the very unwell this sad business he is probable credit of Elizabeth recollected their virtue in the Netherfield in her hand .

Though Darcy was good-looking and imposing manners indicated respectability of hearing this , how his misfortunes have conducted herself the evening Mrs. Bennet , and summer months , which her new scene at a uniform cheerfulness in the desire for what she would not kind of first thought her handkerchief , `` That the business , and is disposed to avoid seeing Bingley does the glories of such terrific ideas connected with quick in love with his profession , for Pulvis Lodge to discover at St. James 's ruin him , and she , who , at the case , and for an epithet .

Random Markov-order-3 (trigram) text:

I know my mother ; '' she exclaimed , `` I have been designed for him , to make some sort of girl is Miss King , and all her own room , and the continual presents in money matters were then to receive ; and Mr. Bingley did not mean that his flight was rendered necessary by distress of the impropriety of conduct , the advice and entreaty of so many servants , contrived to have , and she had need to be , and leave her very little , and welcomed to them , on his returning no more .

Her aunt assured her with a call from her books , and brought its master , '' replied her husband 's incivility ; though , at last to go so often turned towards Mr. Collins would have taken up their abode with her sister .

To Kitty , for I know not ; she had a letter . ''

With all these threats in a scarlet coat , and tries to persuade her to walk into the saloon , whose manners were in general .

Though I _know_ it must be done too much .

Random Markov-order-4 (quadgram) text:

Mr. Bennet was among the earliest of those who best knew the easiness of his temper , whether he might not spend the remainder of his days at Netherfield , and he had the opportunity of enjoying herself as much as her nature will allow .

She often tried to provoke Darcy into disliking her guest , by talking of their supposed marriage , and planning his happiness in such an alliance .

The girls grieved over such a number of ladies , but were not missed till yesterday morning at eight .

Though her manner varied , however , at her having any such fears now , because he is trying to get a large party .

Four weeks were to pass away before her uncle and aunt stopped also , and , unshackled by business , occupy himself solely in being civil to all the comfort of having a daughter well married ; and she was preparing to see him again .

Usually, a trigram model strikes the right balance:

Enough context to capture patterns...

... but not too corpus-specific.

Open questions:

1. We can’t do anything useful with bigrams we’ve never seen before...
2. Our probabilities are tiny!
3. ~~Do bigrams capture enough context to be useful?~~

Open questions:

1. We can't do anything useful with bigrams we've never seen before...
2. Our probabilities are tiny!
3. ~~Do bigrams capture enough context to be useful?~~

	i	want	to	eat	chinese	food	lunch	spend
i	5	927	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

Problems with the zeros:

1. We can't do anything useful with bigrams we've never seen before...
2. It doesn't reflect "reality"- the zeros are probably just due to insufficient training data...
3. Many useful metrics and algorithms will blow up if we feed them zeros...

The solution: "Smoothing"

The solution: “Smoothing”

The basic idea: take probability mass from counts we *have* seen and shift it to counts we *haven’t* seen.

This will make our probability distribution less less jagged (smoother).

The solution: “Smoothing”

There are lots of ways to do this...

We’ll hear about several on Wednesday.

Today, we’ll introduce a *very* simple (and not very good) method.

Laplace (“add-one”) smoothing:

Simply add 1 to every observation.

Unsmoothed (“Maximum Likelihood”) estimator: $P(w_i) = \frac{c_i}{N}$

Laplace-smoothed estimator: $P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$

Laplace ("add-one") smoothing:

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$$

What will this do?

Previously-unseen counts will go from 0 to 1...

Existing MLE probabilities will *decrease* a bit...

... since we are "stealing" some of their mass, and giving it to the unseen observations!

It can be useful to talk about the "adjusted counts"

Laplace ("add-one") smoothing:

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$$

When working with a smoothing algorithm, It can be useful to talk about the "adjusted counts."

For unigram Laplace smoothing: $c_i^* = (c_i + 1)\frac{N}{N + V}$

Laplace is easy to extend to the bigram case:

$$P^*_{Laplace}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

$$c^*(w_{n-1}w_n) = \frac{(C(w_{n-1}w_n) + 1) \times C(w_{n-1})}{C(w_{n-1}) + V}$$

Unsmoothed counts:

	i	want	to	eat	chinese	food	lunch	spend
i	5	927	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

Laplace-smoothed counts:

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	84	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Unsmoothed probabilities:

	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

Laplace-smoothed probabilities:

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Unsmoothed counts:

	i	want	to	eat	chinese	food	lunch	spend
i	5	927	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

Laplace-smoothed reconstructed:

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

Laplace ("add-one") smoothing:

Very important note: Laplace smoothing is *NOT* a useful smoothing algorithm outside of a lecture hall.

Note that the *discount* factor can be huge:

$$P_{MLE}(to|want) = 0.66$$

$$P_{Laplace}(to|want) = 0.26$$

The algorithm is moving *too much* probability mass off of our observed counts.

Other algorithms do a better job: stay tuned for Wednesday!

Open questions:

1. We can't do anything useful with bigrams we've never seen before...
2. Our probabilities are tiny!
3. ~~Do bigrams capture enough context to be useful?~~

Open questions:

1. ~~We can't do anything useful with bigrams we've never seen before...~~
2. Our probabilities are tiny!
3. ~~Do bigrams capture enough context to be useful?~~

Open questions:

1. ~~We can't do anything useful with bigrams we've never seen before...~~

2. Our probabilities are tiny!

3. ~~Do bigrams capture enough context to be useful?~~

Plan for the day:

1. Modeling context
2. Probability
3. Language Models
4. Markov Chains
5. Implementation Notes

Example:

	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(i | <s>) = 0.25

p(food | english) = 0.5

p(english | want) = 0.0011

p(</s> | food) = 0.68

P(<s> i want english food </s>)

$$
\begin{aligned}
&= P(i|<s>)P(want|i)...P(</s>|food) \\
&= 0.25 \times 0.33 \times 0.0011 \times 0.5 \times 0.68 \\
&= 0.000031
\end{aligned}
$$

Example from J&M Chap. 4

P(<s> i want english food </s>)

$$
\begin{aligned}
&= P(i| < s >)P(want|i)...P(< /s > |food) \\
&= 0.25 \times 0.33 \times 0.0011 \times 0.5 \times 0.68 \\
&= 0.000031
\end{aligned}
$$

This is with a *tiny* corpus, and a very short sentence containing some very common bigrams.

The solution: Store log-probabilities, and work in log-space.

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

$$p_1 \times p_2 \times p_3 ... \times p_n = exp(\log p_1 + \log p_2 + \log P_3 ... + \log p_n)$$

BitWeight to the rescue:

```
class BitWeight(object):
    def __init__(self, val=0., is_neg_log=False):
        if is_neg_log:
            self.bw = val
        else:
            self.bw = -log2(val)
```

```
>>> a_real = .5
>>> b_real = .25
>>> a_bw = BitWeight(a_real)
>>> b_bw = BitWeight(b_real)
>>> (a_bw * b_bw).to_real == a_real * b_real
True
```