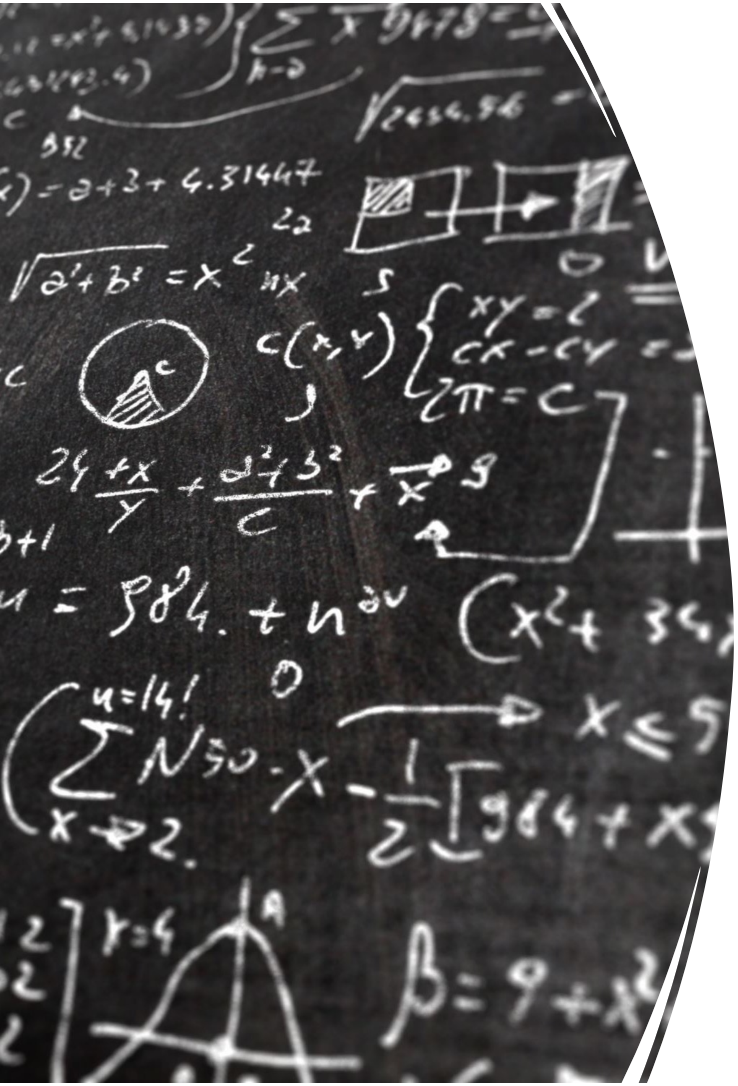


CS 383 – Computational Text Analysis

Lecture 2
Language Modeling

Adam Poliak

01/23/2023

Slides adapted from Philipp Koehn, Jordan Boyd-Graber, Jason Eisner, Dan Jurafsky

Announcements

- Office Hours:
  - Thursdays 3-4:30pm
    - There are a few I will reschedule
  - After lecture on Monday
- HW00 due tonight
- Reading01 due tonight
- HW01 released tonight, due Monday 01/30
  - Based on today's lecture
- Reading02 released tonight, due Monday 01/30

Outline

- NLP/HLT/CTA
- Define LMs
  - Motivate LMs, applications
- Probability review
  - Joint
  - Conditional
  - Chain rule
- N-grams
- Computing LMs
  - MLE
  - Smoothing
- Evaluating LMs

Language Model

Answers the question(s):

- How likely is a piece of text a good example of the language?
- How likely is a given piece of text to be seen in the wild?

“assigns probabilities to sequences of words” - textbook

Why do we want probabilities to sequences of words? What can we do with probabilities of sequences of words?

Classification:

- LanguageID
- Text Categorization
- Authorship attribution

Predict next word

- Texting on phone

Autocorrect/Spelling Correction

Machine Translation

Language Generation

Contextual Spelling Correction

- Which is most probable?
  1. … I think they’re okay …
  2. … I think there okay …
  3. … I think their okay …

- Which is most probable?
  1. … by the way, are they’re likely to …
  2. … by the way, are there likely to …
  3. … by the way, are their likely to …

Machine Translation

	good English? (n-gram)	good match to French?
Jon appeared in TV.		
Appeared on Jon TV.		
In Jon appeared TV.		
Jon is happy today.		
Jon appeared on TV.		
TV appeared on Jon.		
TV in Jon appeared.		
Jon was not happy.		

Machine Translation

	good English? (n-gram)	good match to French?
Jon appeared in TV.		✓
Appeared on Jon TV.		
In Jon appeared TV.		✓
Jon is happy today.	✓	
Jon appeared on TV.	✓	✓
TV appeared on Jon.	✓	
TV in Jon appeared.		
Jon was not happy.	✓	

Language generation

- Choose randomly among outputs:
  - Visitant which came into the place where it will be Japanese has admired that there was Mount Fuji.
- Top 10 outputs according to bigram probabilities:
  - Visitors who came in Japan admire Mount Fuji.
  - Visitors who came in Japan admires Mount Fuji.
  - Visitors who arrived in Japan admire Mount Fuji.
  - Visitors who arrived in Japan admires Mount Fuji.
  - Visitors who came to Japan admire Mount Fuji.
  - A visitor who came in Japan admire Mount Fuji.
  - The visitor who came in Japan admire Mount Fuji.
  - Visitors who came in Japan admire Mount Fuji.
  - The visitor who came in Japan admires Mount Fuji.
  - Mount Fuji is admired by a visitor who came in Japan.

How do we compute probability of a sequence of words?

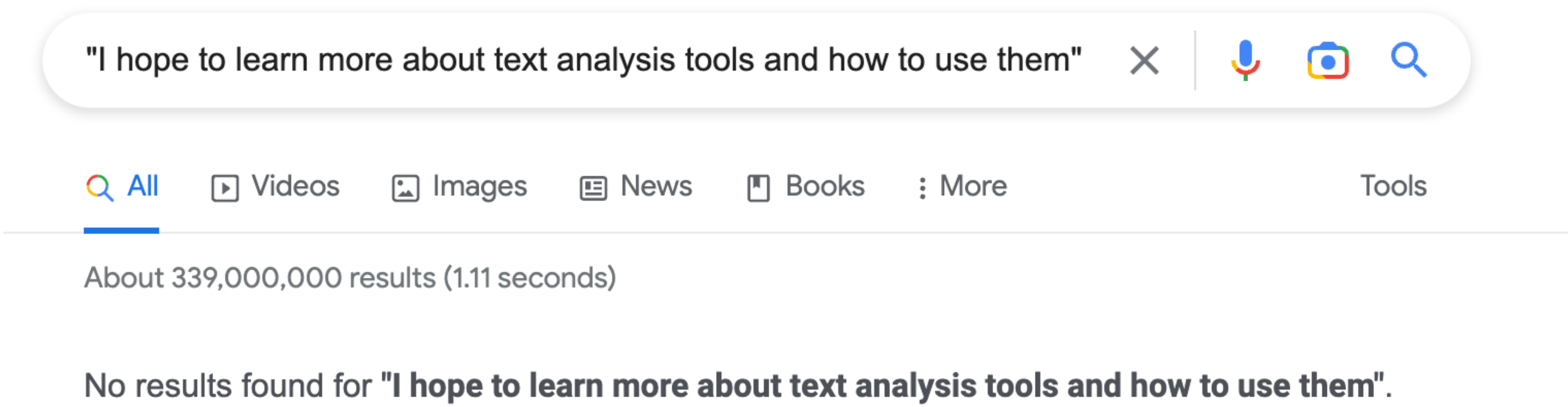
Today’s main topic!

P(“*I hope to learn more about text analysis tools and how to use them*”) = ????

Approach 0: Look up how many times we’ve seen this sentence before?

Issue with Approach 0

Most sentences have never been seen before



Outline

- NLP/HLT/CTA
- Define LMs
  - Motivate LMs, applications
- Probability review
  - Joint
  - Conditional
  - Chain rule
- N-grams
- Computing LMs
  - MLE
  - Smoothing
- Evaluating LMs

Probability side bar

P(*"I hope to learn more about text analysis tools and how to use them*")

What type of probability is this?

Joint probability

What's the probability of event A and event B of both happening

Event A = "I"

Event B = "hope"

Event C = "learn"

…

Probability side bar: Joint

$P(A,B)$: Probability of event A and event B both happening

$$P(A,B) \leq P(A)$$
$$P(A,B) \leq P(B)$$

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

Probability side bar: Conditional

$P(A \mid B)$: Probability of event A happening if we know event B is happening

$$P(A \mid B) = {}^{P(A,B)}\!\big/_{P(B)}$$

Therefore,

$$P(A\,, B) = P(A\,|B)\; * P(B)$$

Probability side bar: more variables

$P(A, B, C, D)$:

Probability of A and B and C and D happening

Recall, $P(A, B) = P(A)P(B|A)$

(Probability of A and B) and C and D happening

$P(A)P(B|A)$ and C and D happening

$P(A)P(B|A)\ P(C \mid A, B)$ and D happening

$$P(A, B, C, D) = P(A)P(B|A)\ P(C \mid A, B)P(D|A, B, C)$$

Probability Chain Rule

$$P(x_1, x_2, x_3, \dots, x_n) =$$
$$P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(xn|x_1, \dots, xn\ \_\ 1)$$

More compactly

$$P(x_1, x_2, x_3, \dots, x_n) =$$
$$\prod_i P(x_i\ |x_1, x_2, \dots, x_{i-1})$$

Back to language

What are our random variables?

The words in our sentence

Probability of $w_1, w_2, \ldots, w_n =$

$\prod_n P(w_n \mid w_1, w_2, \ldots, w_{n-1})$

P(*“I hope to learn more about text analysis tools and how to use them”*)

$P(\text{“}I\text{”})$
$* P(\text{“}hope \mid I\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\text{”})$
$* P(\text{“}learn\text{”} \mid \text{”}I\ hope\ to\text{”})$
$* P(\text{“}more\text{”} \mid \text{“}I\ hope\ to\ learn\text{”}) * P(\text{“}about\text{”} \mid \text{“}I\ hope\ to\ learn\ more\text{”})$
$* P(\text{“}text\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\text{”})$
$* P(\text{”}analysis\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\text{”})$
$* P(\text{“}tools\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})$
$* P(\text{“}and\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})$
$* P(\text{“}how\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{”})$

…

Compute P("I")

$P(\text{"}I\text{"})$
$* P(\text{"}hope \mid I\text{"})$
$* P(\text{"}to\text{"} \mid \text{"}I\ hope\text{"})$
$* P(\text{"}learn\text{"} \mid \text{"}I\ hope\ to\text{"})$
$* P(\text{"}more\text{"} \mid \text{"}I\ hope\ to\ learn\text{"}) * P(\text{"}about\text{"} \mid \text{"}I\ hope\ to\ learn\ more\text{"})$
$* P(\text{"}text\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\text{"})$
$* P(\text{"}analysis\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\ text\text{"})$
$* P(\text{"}tools\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{"})$
$* P(\text{"}and\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{"})$
$* P(\text{"}how\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{"})$
$* P(\text{"}to\text{"} \mid \text{"}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{"})$

…

Compute P("I")

$$P(\text{"}I\text{"}) = \frac{count(\text{"}I\text{"})}{N}$$ (where N is the number of tokens)

Compute P(“hope” | “I”)

$P(\text{“}I\text{”})$
$* P(\text{“}hope \mid I\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\text{”})$
$* P(\text{“}learn\text{”} \mid \text{”}I\ hope\ to\text{”})$
$* P(\text{“}more\text{”} \mid \text{“}I\ hope\ to\ learn\text{”}) * P(\text{“}about\text{”} \mid \text{“}I\ hope\ to\ learn\ more\text{”})$
$* P(\text{“}text\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\text{”})$
$* P(\text{”}analysis\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\text{”})$
$* P(\text{“}tools\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})$
$* P(\text{“}and\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})$
$* P(\text{“}how\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{”})$

…

Compute P(“hope” | “I”)

$$P(\text{''}hope \mid \text{``}I\text{''}) =$$

$$\frac{count(\text{``}I\ hope\text{''})}{count\ (\text{``}I\text{''})}$$

Compute P(“to” | “I hope”)

$P(\text{“}I\text{”})$
$* P(\text{“}hope \mid I\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\text{”})$
$* P(\text{“}learn\text{”} \mid \text{”}I\ hope\ to\text{”})$
$* P(\text{“}more\text{”} \mid \text{“}I\ hope\ to\ learn\text{”}) * P(\text{“}about\text{”} \mid \text{“}I\ hope\ to\ learn\ more\text{”})$
$* P(\text{“}text\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\text{”})$
$* P(\text{”}analysis\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\text{”})$
$* P(\text{“}tools\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})$
$* P(\text{“}and\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})$
$* P(\text{“}how\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{”})$

…

Compute P("to" | "I hope")

$$P(\text{"}to \mid \text{"}I\ hope\text{"}) =$$

$$\frac{count(\text{"}I\ hope\ to\text{"})}{count\ (\text{"}I\ hope\text{"})}$$

Compute P(“hope” | “I”)

$$
\begin{aligned}
&P(\text{“}I\text{”})\\
&* P(\text{“}hope \mid I\text{”})\\
&* P(\text{“}to\text{”} \mid \text{“}I\ hope\text{”})\\
&* P(\text{“}learn\text{”} \mid \text{”}I\ hope\ to\text{”})\\
&* P(\text{“}more\text{”} \mid \text{“}I\ hope\ to\ learn\text{”}) * P(\text{“}about\text{”} \mid \text{“}I\ hope\ to\ learn\ more\text{”})\\
&* P(\text{“}text\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\text{”})\\
&* P(\text{”}analysis\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\text{”})\\
&* P(\text{“}tools\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})\\
&* P(\text{“}and\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})\\
&* P(\text{“}how\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{”})\\
&* P(\text{“}to\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{”})
\end{aligned}
$$

…

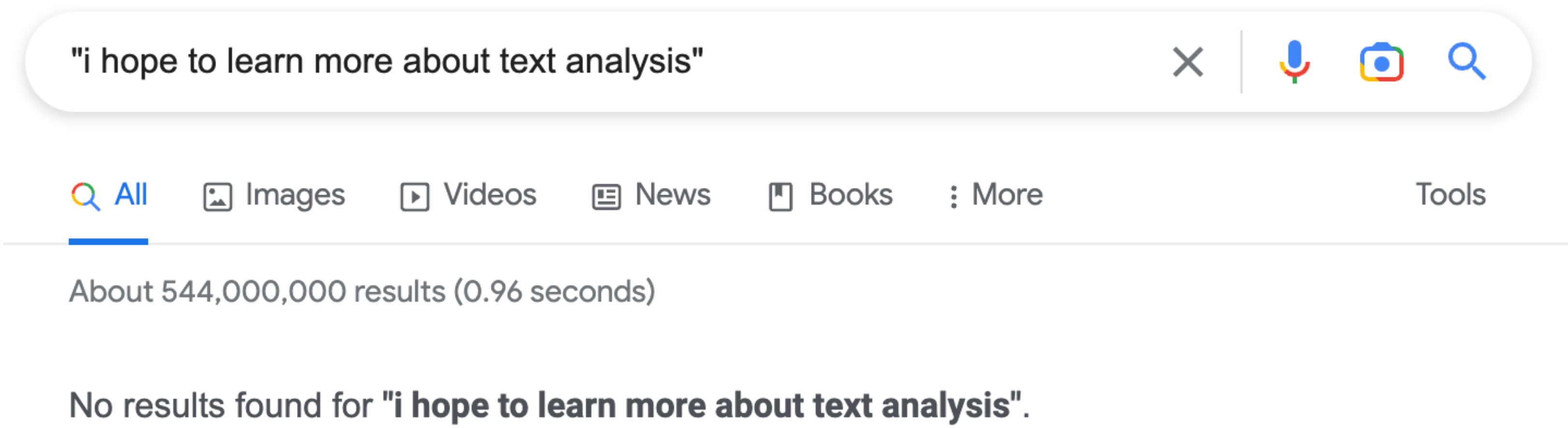
Compute P("tools" | "I hope to learn more about text analysis")

$P(\text{“}I\text{”})$
$* P(\text{“}hope \mid I\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\text{”})$
$* P(\text{“}learn\text{”} \mid \text{”}I\ hope\ to\text{”})$
$* P(\text{“}more\text{”} \mid \text{“}I\ hope\ to\ learn\text{”}) * P(\text{“}about\text{”} \mid \text{“}I\ hope\ to\ learn\ more\text{”})$
$* P(\text{“}text\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\text{”})$
$* P(\text{”}analysis\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\text{”})$
$* P(\text{“}tools\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})$
$* P(\text{“}and\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})$
$* P(\text{“}how\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\text{”})$
$* P(\text{“}to\text{”} \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\ and\ how\text{”})$

…

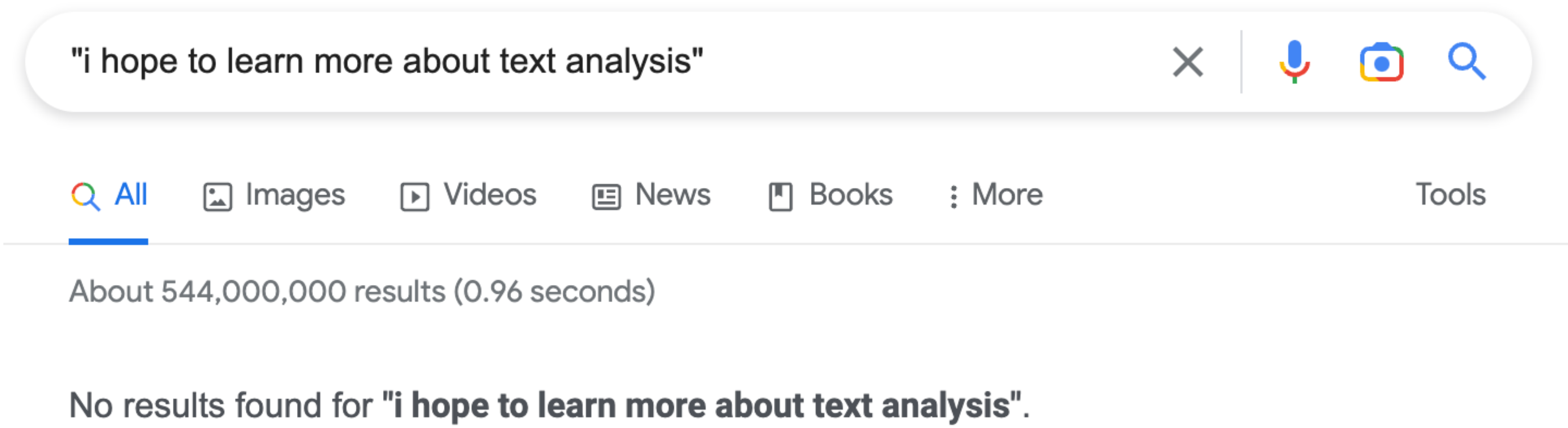
Compute P(“tools” | “I hope to learn more about text analysis”)

$$P(\text{”}tools \mid \text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”}) = \frac{count(\text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\ tools\text{”})}{count\ (\text{“}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{”})}$$



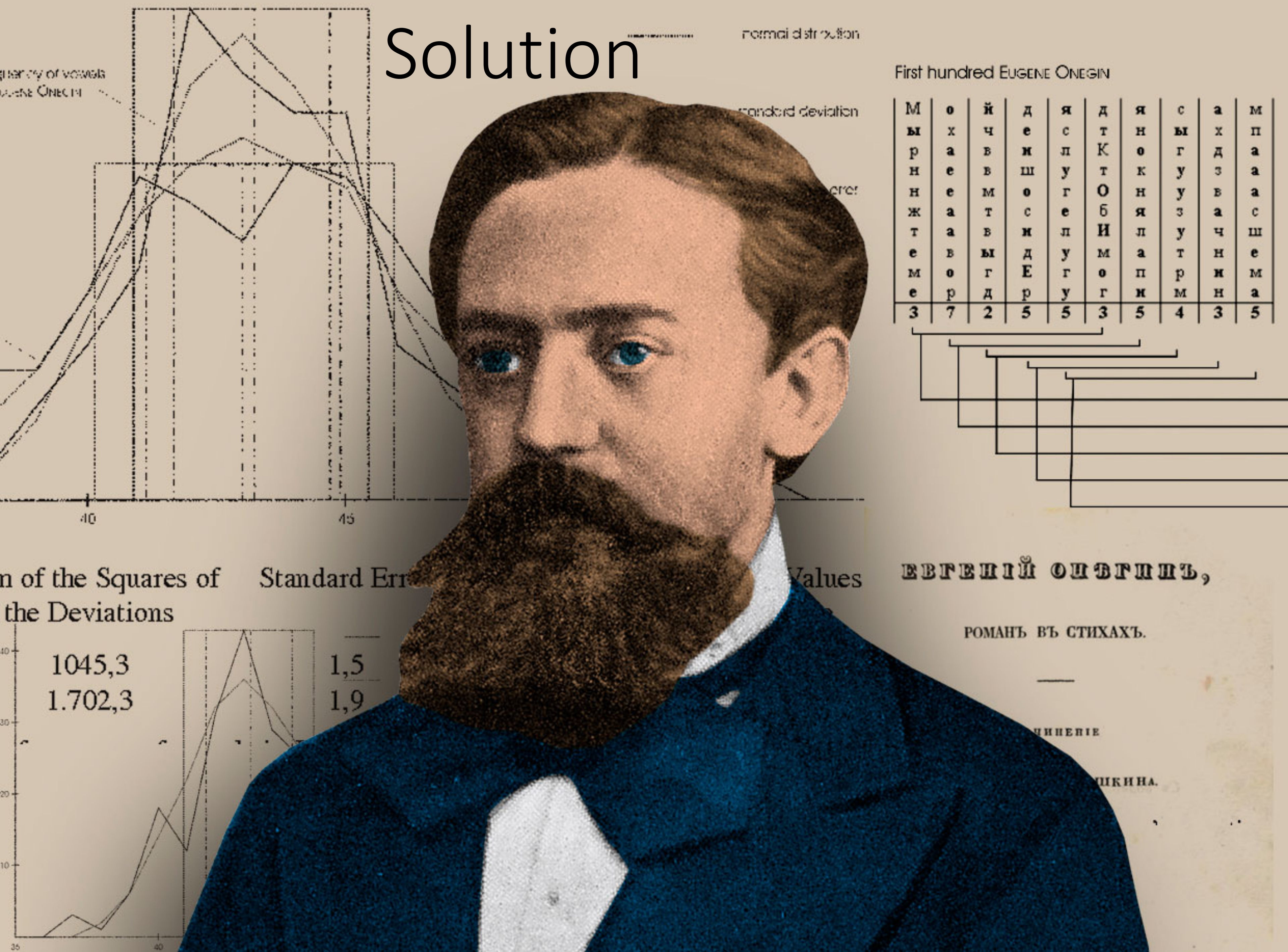
Compute P("tools" | "I hope to learn more about text analysis")

$$P(\text{"}tools \mid \text{``}I\ hope\ to\ learn\ more\ about\ text\ analysis\text{''}) =$$



Same issue as before

Solution


First hundred EUGENE ONEGIN
standard deviation
40
45
n of the Squares of the Deviations
Standard Err
Values
1045,3
1.702,3
1,5
1,9
ЕВГЕНІЙ ОНѢГИНЪ,
РОМАНЪ ВЪ СТИХАХЪ.

Markovian Assumption

Andrei Markov

- Simplifying assumption:

$$P(analysis\ |I\ hope\ to\ learn\ more\ about\ textual) \approx P(analysis\ |learn\ more\ about\ textual)$$

- Or maybe

$$P(analysis\ |I\ hope\ to\ learn\ more\ about\ textual) \approx P(analysis\ |\ textual)$$

Slide from textbook slides

Markov Assumption in plain language

Don't worry too much about the past

Markov Assumption

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

In other words, we approximate each component in the product by recent history

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

Slide from textbook slides

So how far back should we go?

One word

Two words

Three words

5 words

So how far back should we go?

One word

Unigram $P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$

Two words

Bigram $P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$

Three words

Trigram

5 words

Five-gram

n-grams

“a sequence of n words”

Alternatively:

“predictive model that assigns it a probability”

Outline

- NLP/HLT/CTA
- Define LMs
  - Motivate LMs, applications
- Probability review
  - Joint
  - Conditional
  - Chain rule
- N-grams
- Computing n-grams/LMs
  - MLE
  - Smoothing
- Evaluating LMs

Computing n-gram probabilities

MLE: Maximum Likelihood Estimate

Computing bi-grams:

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

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

MLE: Maximum Likelihood Estimate

Computing tri-grams:

$$P(w_i \,|w_{i-1}, w_{i-2}) = \frac{count(w_i, w_{i-1}, w_{i-2})}{count(w_{i-1}, w_{i-2})}$$

$$P(w_i \,|w_{i-1}, w_{i-2}) = \frac{c\ (w_i, w_{i-1}, w_{i-2})}{c\ (w_{i-1}, w_{i-2})}$$

Maximum Likelihood Estimates

- The maximum likelihood estimate
  - of some parameter of a model M from a training set T
  - maximizes the likelihood of the training set T given the model M
- Suppose the word “bagel” occurs 400 times in a corpus of a million words
- What is the probability that a random word from some other text will be “bagel”?
- MLE estimate is 400/1,000,000 = .0004
- This may be a bad estimate for some other corpus
  - But it is the **estimate** that makes it **most likely** that “bagel” will occur 400 times in a million word corpus.

An example (bi-gram)

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

<s> I am Sam </s>

<s> Sam I am </s>

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

P(I | <s>)　　P(Sam | <s>)　　P(am | I)

P(<s> | Sam)　　P(Sam | am)　　P(do | I)

An example (bi-gram)

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

<s> I am Sam </s>

<s> Sam I am </s>

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

$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67$ $\quad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33$ $\quad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$

$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5$ $\quad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5$ $\quad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$

More examples: Berkeley Restaurant Project sentences

- 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

Raw bigram counts

- Out of 9222 sentences

	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

Raw bigram probabilities

- Normalize by unigrams:

- Result:

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

Bigram estimates of sentence probabilities

$P(\langle s\rangle \text{ I want english food } \langle /s\rangle) =$

$P(\text{I}|\langle s\rangle)$

$\times\ P(\text{want}|\text{I})$

$\times\ P(\text{english}|\text{want})$

$\times\ P(\text{food}|\text{english})$

$\times\ P(\langle /s\rangle|\text{food})$

$= .000031$

What kinds of knowledge?

- 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

How did we learn this knowledge?

How did we learn this knowledge?

Just by counting!

Bigram estimates of sentence probabilities

P(<s> I want english food because it is very very yummy </s>)

=

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

…

× P(</s>|yummy)

Bigram estimates of sentence probabilities

P(<s> I want english food because it is very very yummy </s>) =

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

…

× P(</s>|yummy)

= .00000000000000001

Practical Issues

- We do everything in log space
  - Avoid underflow
  - (also adding is faster than multiplying)

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Example (tri-gram)

Data from Europarl

word	c	P(word | “the red”)
		0.547
		0.138
		0.040
		0.031
		0.022

Example (tri-gram)

Data from Europarl

word	c	P(word | “the red”)
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

How many trigrams starting with “the red” appear in Europarl?

225

What’s probability of “Strengthen capacities of the Red Crescent Society of Kazakhstan”?

Assuming a trigram model

$$P(strengthen \mid <s><s>) * P(capacities \mid <s> strengthen) * \ldots P(crescent \mid the\ red) * \ldots$$

What's $P(crescent \mid the\ red)$?

Assuming a trigram model

$$P(strengthen \mid <s><s>) * P(capacities \mid <s>\ strengthen) * \ldots P(crescent \mid the\ red) * \ldots$$

word	c	P(word | "the red")
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

What's probability of "Strengthen capacities of the Red Crescent Society of Kazakhstan"?

Assuming a trigram model

$$P(strengthen \mid <s><s>) * P(capacities \mid <s> \; strengthen) * \ldots P(crescent \mid the \; red) * \ldots$$

word	c	P(word | "the red")
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

What's probability of "Strengthen capacities of the Red Crescent Society of Kazakhstan"?

Assuming a trigram model

$$P(strengthen \mid <s><s>) * P(capacities \mid <s>\ strengthen) * \ldots \quad 0 * \ \ldots$$

word	c	P(word | "the red")
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

Unknown n-grams

If we have an n-gram we haven't seen before, probability of the sequence is equal to _0_

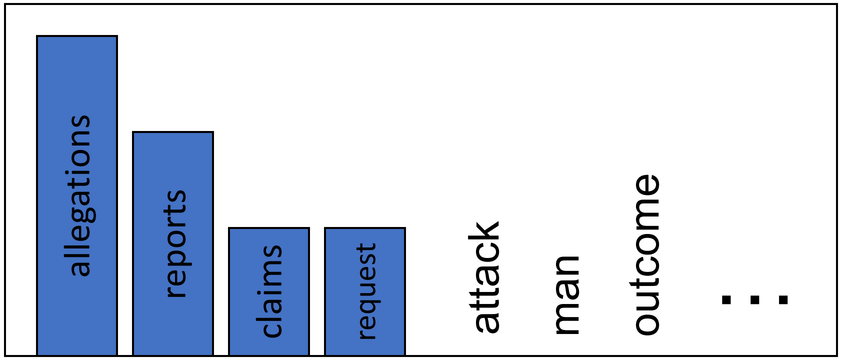
Smoothing

The intuition of smoothing (from Dan Klein)

- When we have sparse statistics:

  P(w | denied the)
  3 allegations
  2 reports
  1 claims
  1 request

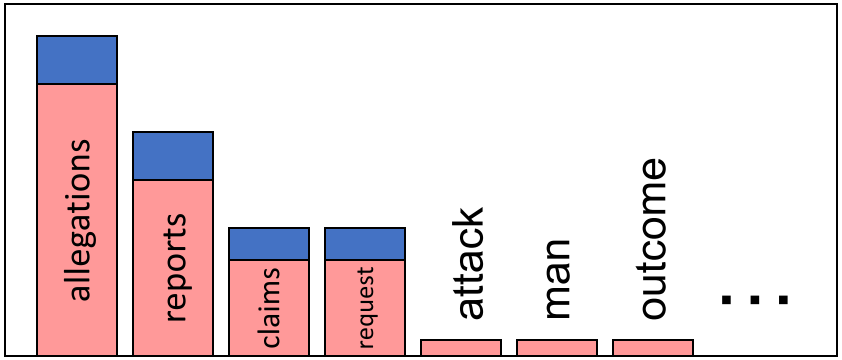
  7 total



- Steal probability mass to generalize better

  P(w | denied the)
  2.5 allegations
  1.5 reports
  0.5 claims
  0.5 request
  2 other

  7 total



Add-1 smoothing (aka Laplace smoothing)

Just add one to every count

MLE Estimate: $P_{MLE}\ (w_i\ |w_{i-1}) = \frac{c(w_i,w_{i-1})}{c(w_{i-1})}$

Add-1 Estimate: $P_{Add-1}\ (w_i\ |w_{i-1}) = \frac{c(w_i,w_{i-1})+1}{c(w_{i-1})}$

Why is this Add-1 Estimate incorrect?

$\sum P_{Add-1}\ (w_i\ |w_{i-1}) \quad = 1$ isn't true anymore

Add-1 smoothing
(aka Laplace smoothing)

Just add one to every count

MLE Estimate: $P_{MLE}\,(w_i\,|w_{i-1}) = \frac{c(w_i,w_{i-1})}{c(w_{i-1})}$

Add-1 Estimate: $P_{Add-1}\,(w_i\,|w_{i-1}) = \frac{c(w_i,w_{i-1})+1}{c(w_{i-1})+V}$

Now

$$\sum P_{Add-1}\,(w_i\,|w_{i-1}) \;\; = 1$$

Raw bigram 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

Laplacian bigram 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

Add-α smoothing

MLE Estimate: $P_{MLE}\,(w_i\,|w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})}$

Add-1 Estimate: $P_{Add-1}\,(w_i\,|w_{i-1}) = \frac{c(w_i, w_{i-1})+1}{c(w_{i-1})+V}$

Add- α Estimate: $P_{Add-\alpha}\,(w_i\,|w_{i-1}) = \frac{c(w_i, w_{i-1})+\alpha_i}{c(w_{i-1})+\alpha_k}$

Assumes a sparse Dirichlet prior

Add-1 estimation in practice

- add-1 isn’t used for N-grams:
  - Not every word should get the same boost in every situation
  - We’ll see better methods

- But add-1 is used to smooth other NLP models
  - For text classification
  - In domains where the number of zeros isn’t so huge.

What other approaches might we try?

Longer vs shorter n-grams higher vs lower order n-grams

Big n:

- Sensitive to more context
- More sparse

Small n

- Consider short context
- Robust counts

Approach 1: Backoff

When we have good higher-order n-grams, use them. Otherwise, use lower-order n-grams

For example:

Start with 4-gram, if not good,

use tri-gram, if not good,

use bi-gram, if not good,

use unigram

Approach 2 – Combine the ‘grams

We call this interpolation

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) \;=\; & \lambda_1 P(w_n|w_{n-2}w_{n-1}) \\ & +\lambda_2 P(w_n|w_{n-1}) \\ & +\lambda_3 P(w_n)\end{aligned}$$

Weighted average of all the grams

Approach 2 – Combine the ‘grams

Context specific weights

Jelinek-Mercer smoothing (1980)

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) \;=\;& \lambda_1(w_{n-2}^{n-1})P(w_n|w_{n-2}w_{n-1}) \\ &+\lambda_2(w_{n-2}^{n-1})P(w_n|w_{n-1}) \\ &+\lambda_3(w_{n-2}^{n-1})P(w_n)\end{aligned}$$

Additional Approaches

- Discounted backoff (Katz backoff)
- Stupid backoff
- Kneser-Ney smoothing
  - Extra credit

Evaluating Language Models

Perplexity

$$Perplexity\ (w_1, w_2, w_3, \dots, w_n) =$$

$$= P(w_1, w_2, w_3, \dots, w_n)^{\frac{1}{n}}$$

$$= \sqrt[n]{\frac{1}{P(w_1, w_2, w_3, \dots, w_n)}}$$

$P(w_1, w_2, w_3, \dots, w_n)$ depends on the LM we use

The lower the perplexity, the better the model

Perplexity

$$Perplexity\ (w_1, w_2, w_3, \ldots, w_n) =$$

$$= \sqrt[n]{\frac{1}{P(w_1,w_2,w_3,\ldots,w_n)}}$$

The lower the perplexity, =>

the higher the probability =>

the model is less surprised by the sentence

Summary

- Motivate LMs, applications
- Reviewed
  - Joint
  - Conditional
  - Chain rule
- N-grams
- Training LMs
- Evaluating LMs

Bonus of LMs

We can generate text!