Language Models

CS288
UC Berkeley

Language Models

Language Models

Acoustic Confusions

the station signs are in deep in english	-14732
the stations signs are in deep in english	-14735
the station signs are in deep into english	-14739
the station 's signs are in deep in english	-14740
the station signs are in deep in the english	-14741
the station signs are indeed in english	-14757
the station 's signs are indeed in english	-14760
the station signs are indians in english	-14790

Noisy Channel Model: ASR

- We want to predict a sentence given acoustics:

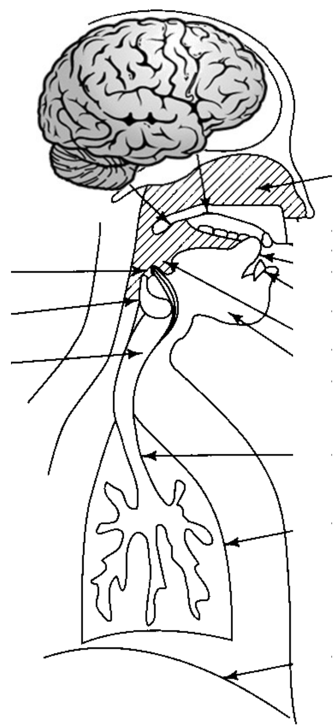
$$w^* = \arg\max_w P(w|a)$$

- The noisy-channel approach:

$$\begin{aligned} w^* &= \arg\max_w P(w|a) \\ &= \arg\max_w {\color{blue}P(a|w)}{\color{red}P(w)}/P(a) \\ &\propto \arg\max_w {\color{blue}P(a|w)}{\color{red}P(w)} \end{aligned}$$

Acoustic model: score fit between sounds and words

Language model: score plausibility of word sequences

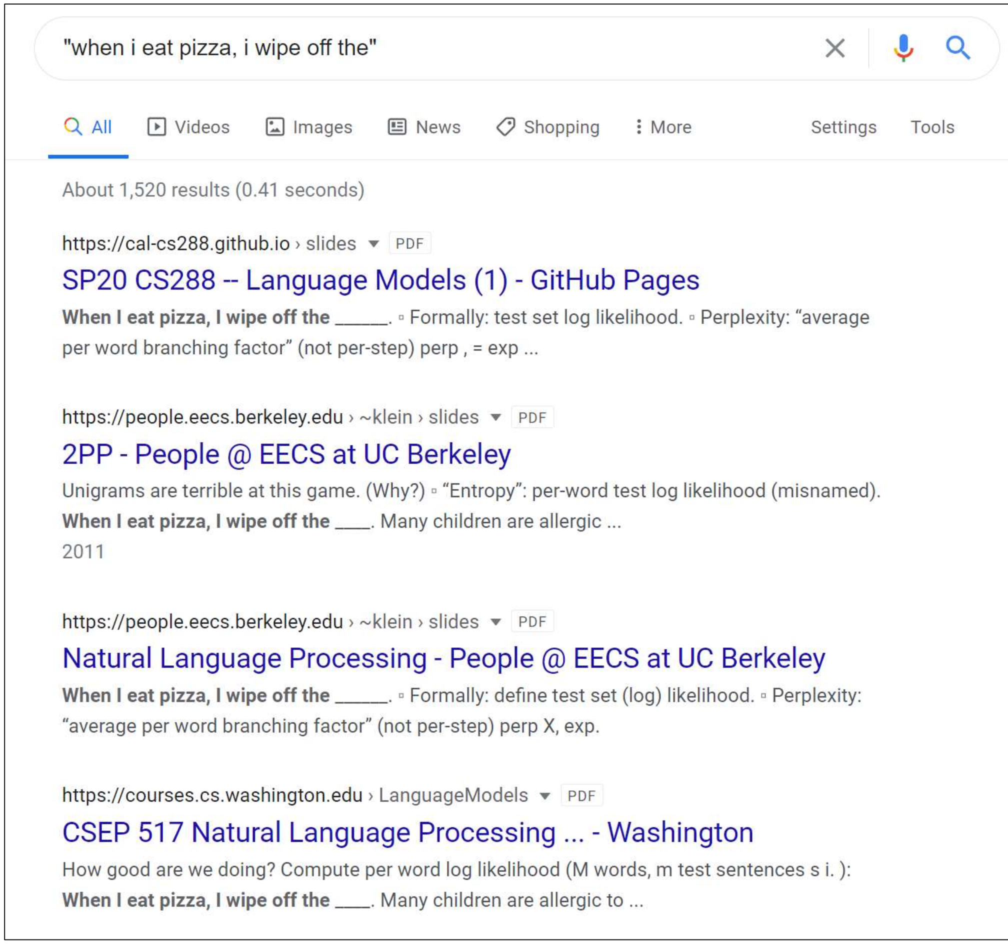


Noisy Channel Model: Translation

"Also knowing nothing official about, but having guessed and inferred considerable about, the powerful new mechanized methods in cryptography—methods which I believe succeed even when one does not know what language has been coded—one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.' "

Warren Weaver (1947)

Perplexity



"when i eat pizza, i wipe off the"

All | Videos | Images | News | Shopping | More | Settings | Tools

About 1,520 results (0.41 seconds)

https://cal-cs288.github.io › slides PDF
SP20 CS288 -- Language Models (1) - GitHub Pages
When I eat pizza, I wipe off the _____. ▫ Formally: test set log likelihood. ▫ Perplexity: “average per word branching factor” (not per-step) perp , = exp ...

https://people.eecs.berkeley.edu › ~klein › slides PDF
2PP - People @ EECS at UC Berkeley
Unigrams are terrible at this game. (Why?) ▫ “Entropy”: per-word test log likelihood (misnamed). **When I eat pizza, I wipe off the** ____. Many children are allergic ...
2011

https://people.eecs.berkeley.edu › ~klein › slides PDF
Natural Language Processing - People @ EECS at UC Berkeley
When I eat pizza, I wipe off the _____. ▫ Formally: define test set (log) likelihood. ▫ Perplexity: “average per word branching factor” (not per-step) perp X, exp.

https://courses.cs.washington.edu › LanguageModels PDF
CSEP 517 Natural Language Processing ... - Washington
How good are we doing? Compute per word log likelihood (M words, m test sentences s i.):
When I eat pizza, I wipe off the ____. Many children are allergic to ...

____

))

ctor” (not per-step)

$\theta)$

grease 0.5
sauce 0.4
dust 0.05
….
mice 0.0001
….
the 1e-100

3516 wipe off the excess
1034 wipe off the dust
547 wipe off the sweat
518 wipe off the mouthpiece
…
120 wipe off the grease
0 wipe off the sauce
0 wipe off the mice

28048 wipe off the *

N-Gram Models

Generative Models

- Generative models describe a probability distribution over some structure, here a sequence of words.

- Commonly of the form: build sequence one by one, left to right

$$P(w_1 \ldots w_n) = \prod_i P(w_i | w_1 \ldots w_{i-1})$$

- You will also hear "autoregressive": this term refers to example sequences being self-supervising examples for the function P(w|context)

- When trained to predict next words, models may capture many kinds of correlations

N-Gram Models

- Use chain rule to generate words left-to-right

$$P(w_1 \dots w_n) = \prod_i P(w_i | w_1 \dots w_{i-1})$$

- Can't condition atomically on the entire left context

P(??? | The computer I had put into the machine room on the fifth floor just)

- N-gram models make a Markov assumption

$$P(w_1 \dots w_n) = \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

$$P(\text{please close the door}) = \; P(\text{please}|\text{START}) P(\text{close}|\text{please}) \dots P(\text{STOP}|door)$$

Empirical N-Grams

- Use statistics from data (examples here from Google N-Grams)

Training Counts

```
198015222 the first
194623024 the same
168504105 the following
158562063 the world
…
14112454 the door
-----------------
23135851162 the *
```

$$\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162}$$

$$= 0.0006$$

- This is the maximum likelihood estimate, which needs modification
- N-gram models use such counts to compute probabilities on demand

Increasing N-Gram Order

- Higher orders capture more correlations

Bigram Model

198015222	the first
194623024	the same
168504105	the following
158562063	the world
…	
14112454	the door

23135851162	the *

Trigram Model

197302	close the window
191125	close the door
152500	close the gap
116451	close the thread
87298	close the deal

3785230	close the *

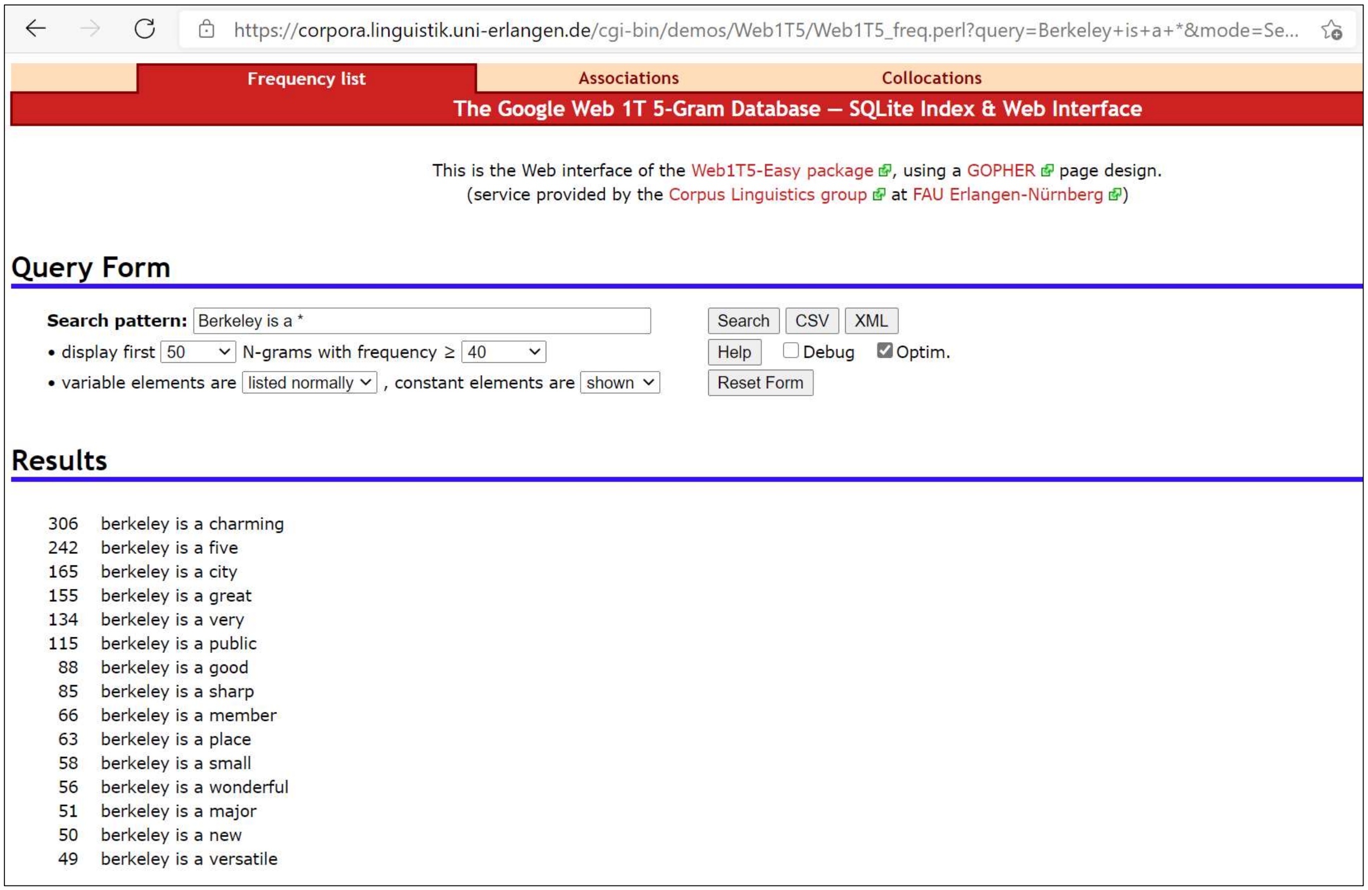
P(door | the) = 0.0006

P(door | close the) = 0.05

Increasing N-Gram Order

Unigram	• 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

N-Grams on the Web



What’s in an N-Gram?

- Just about every local correlation!
  - Word class restrictions: “will have been ___”
  - Morphology: “she ___”, “they ___”
  - Semantic class restrictions: “danced a ___”
  - Idioms: “add insult to ___”
  - World knowledge: “ice caps have ___”
  - Pop culture: “the empire strikes ___”
- But not the long-distance ones
  - “The computer which I had put into the machine room on the fifth floor just ___.”

N-Gram Models: Challenges

Sparsity

Please close the first door on the left.

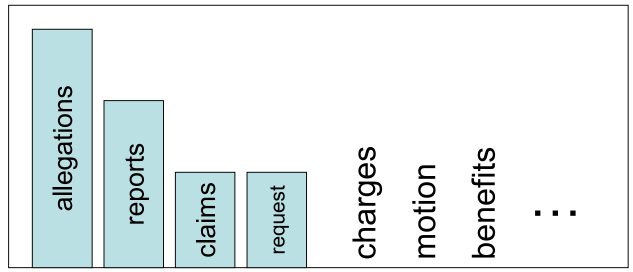
3380 please close the door
1601 please close the window
1164 please close the new
1159 please close the gate
…
0 please close the first

13951 please close the *

Smoothing

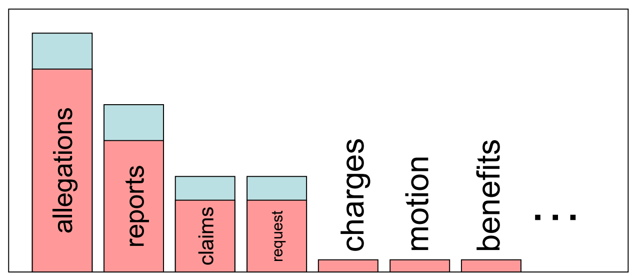
- We often want to make estimates from sparse statistics:

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



- Smoothing flattens spiky distributions so they generalize better:

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



- Very important all over NLP, but easy to do badly

Back-off

Please close the first door on the left.

4-Gram	3-Gram	2-Gram
3380 please close the door	197302 close the window	198015222 the first
1601 please close the window	191125 close the door	194623024 the same
1164 please close the new	152500 close the gap	168504105 the following
1159 please close the gate	116451 close the thread	158562063 the world
…	…	…
0 please close the first	8662 close the first	…
-----------------	-----------------	-----------------
13951 please close the *	3785230 close the *	23135851162 the *
0.0	0.002	0.009

Specific but Sparse ⟺ Dense but General

$$\lambda \hat{P}(w|w_{-1}, w_{-2}) + \lambda' \hat{P}(w|w_{-1}) + \lambda'' \hat{P}(w)$$

Discounting

- Observation: N-grams occur more in training data than they will later

Empirical Bigram Counts (Church and Gale, 91)

Count in 22M Words	Future c* (Next 22M)
1	
2	
3	
4	
5	

- Absolute discounting: reduce counts by a small constant, redistribute "shaved" mass to a model of new events

$$P_{\text{ad}}(w|w') = \frac{c(w', w) - d}{c(w')} + \alpha(w')\hat{P}(w)$$

Fertility

- Shannon game: "There was an unexpected _____"

delay? Francisco?

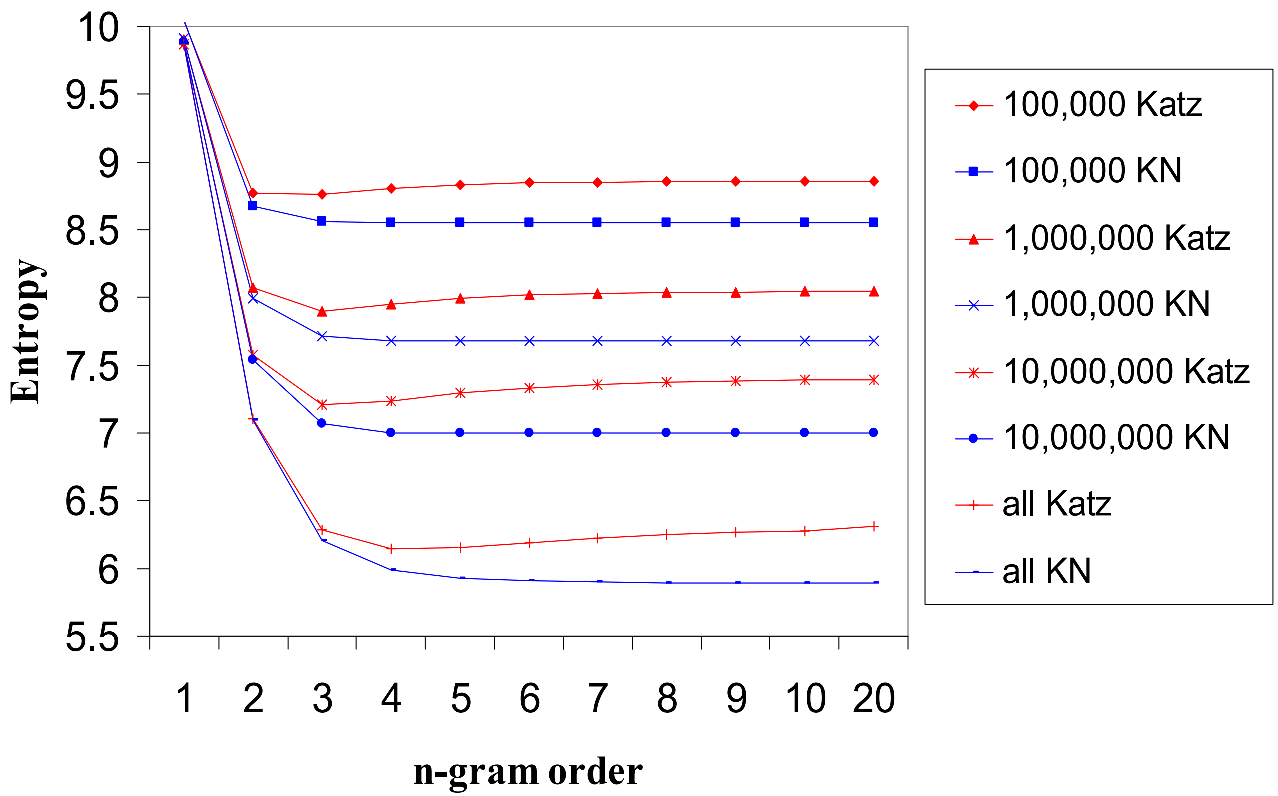
- Context fertility: number of distinct context types that a word occurs in
  - What is the fertility of "delay"?
  - What is the fertility of "Francisco"?
  - Which is more likely in an arbitrary new context?
- Kneser-Ney smoothing: new events proportional to context fertility, not frequency

[Kneser & Ney, 1995]

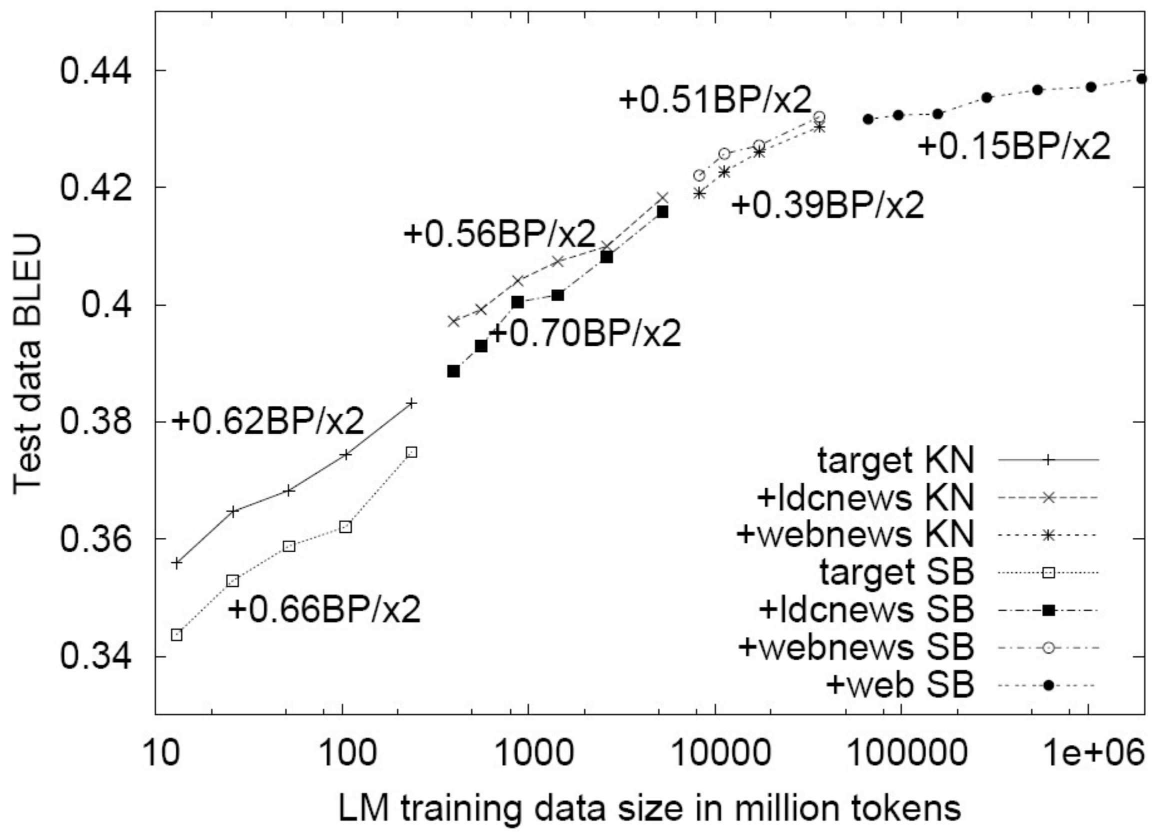
$$P(w) \propto |\{w' : c(w', w) > 0\}|$$

  - Can be derived as inference in a hierarchical Pitman-Yor process [Teh, 2006]

Better Methods?



More Data?



[Brants et al, 2007]

Storage

…	
searching for the best	192593
searching for the right	45805
searching for the cheapest	44965
searching for the perfect	43959
searching for the truth	23165
searching for the “	19086
searching for the most	15512
searching for the latest	12670
searching for the next	10120
searching for the lowest	10080
searching for the name	8402
searching for the finest	8171
…	

Google N-grams

- 14 million < 2^{24} words
- 2 billion < 2^{31} 5-grams
- 770 000 < 2^{20} unique counts
- 4 billion n-grams total

Graveyard of Correlations

- Skip-grams
- Cluster models
- Topic variables
- Cache models
- Structural zeros
- Dependency models
- Maximum entropy models
- Subword models
- …

Entirely Unseen Words

- What about totally unseen words?

- Classical real world option: systems are actually closed vocabulary
  - ASR systems will only propose words that are in their pronunciation dictionary
  - MT systems will only propose words that are in their phrase tables (modulo special models for numbers, etc)

- Classical theoretical option: build open vocabulary LMs
  - Models over character sequences rather than word sequences
  - N-Grams: back-off needs to go down into a “generate new word” model
  - Typically if you need this, a high-order character model will do

- Modern approach: syllable-sized subword units (more later)