Natural Language Processing

Berkeley NLP

Language Modeling I

Dan Klein – UC Berkeley

A Speech Example

The Noisy-Channel Model

- We want to predict a sentence given acoustics:

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

- The noisy-channel approach:

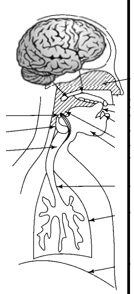
$$w^* = \arg\max_w P(w|a)$$

$$= \arg\max_w P(a|w)P(w)/P(a)$$

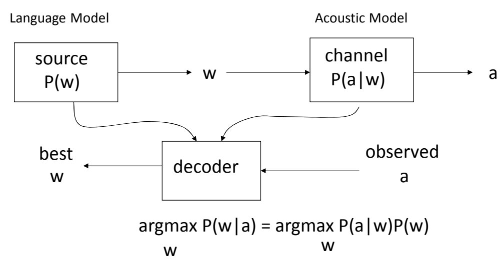
$$\propto \arg\max_w P(a|w)P(w)$$

Acoustic model: HMMs over word positions with mixtures of Gaussians as emissions

Language model: Distributions over sequences of words (sentences)



ASR Components



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
the station signs are indian in english	-14799
the stations signs are indians in english	-14807
the stations signs are indians and english	-14815

Language Models

- A language model is a distribution over sequences of words (sentences)

$$P(w) = P(w_1 \dots w_n)$$

  - What's w? (closed vs open vocabulary)
  - What's n? (must sum to one over all lengths)
  - Can have rich structure or be linguistically naive

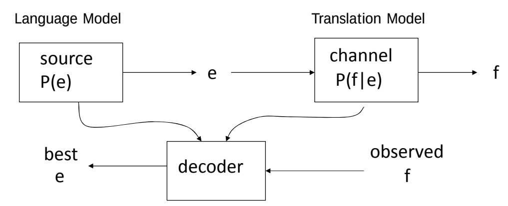
- Why language models?
  - Usually the point is to assign high weights to plausible sentences (cf acoustic confusions)
  - This is not the same as modeling grammaticality

Translation: Codebreaking?

"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)

MT System Components



$$\arg\max_e P(e|f) = \arg\max_e P(f|e)P(e)$$

Other Noisy Channel Models?

- We're not doing this only for ASR (and MT)
  - Grammar / spelling correction
  - Handwriting recognition, OCR
  - Document summarization
  - Dialog generation
  - Linguistic decipherment
  - …

N-Gram Models

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 on the entire left context

P(??? | Turn to page 134 and look at the picture of the)

- 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

- How do we know P(w | history)?
  - Use statistics from data (examples using Google N-Grams)
  - E.g. what is P(door | the)?

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

Increasing N-Gram Order

- Higher orders capture more dependencies

Bigram Model	Trigram Model
198015222 the first	197302 close the window
194623024 the same	191125 close the door
168504105 the following	152500 close the gap
158562063 the world	116451 close the thread
…	87298 close the deal
14112454 the door	
-----------------	-----------------
23135851162 the *	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 execunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Sparsity

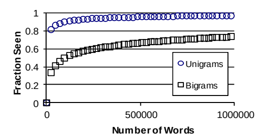
Please close the first door on the left.

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 *

Sparsity

- Problems with n-gram models:
  - New words (open vocabulary)
    - Synaptitute
    - 132,701.03
    - multidisciplinarization
  - Old words in new contexts



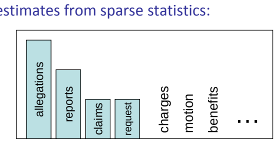
- Aside: Zipf's Law
  - Types (words) vs. tokens (word occurences)
  - Broadly: most word types are rare ones
  - Specifically:
    - Rank word types by token frequency
    - Frequency inversely proportional to rank
  - Not special to language: randomly generated character strings have this property (try it!)
  - This law qualitatively (but rarely quantitatively) informs NLP

N-Gram Estimation

Smoothing

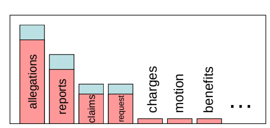
- We often want to make estimates from sparse statistics:

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

Likelihood and Perplexity

- How do we measure LM "goodness"?
  - Shannon's game: predict the next word

When I eat pizza, I wipe off the _________

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

- Formally: define test set (log) likelihood

$$\log \mathrm{P}(X|\theta) = \sum_{w \in X} \log(P(w|\theta))$$

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 *

- Perplexity: "average per word branching factor" (not per-step)

$$\mathrm{perp}(X, \theta) = \exp\left(-\frac{\log P(X|\theta)}{|X|}\right)$$

Train, Held-Out, Test

- Want to maximize likelihood on test, not training data
  - Empirical n-grams won't generalize well
  - Models derived from counts / sufficient statistics require generalization parameters to be tuned on held-out data to simulate test generalization

Training Data	Held-Out Data	Test Data
Counts / parameters from here	Hyperparameters from here	Evaluate here

- Set hyperparameters to maximize the likelihood of the held-out data (usually with grid search or EM)

Measuring Model Quality (Speech)

- We really want better ASR (or whatever), not better perplexities
- For speech, we care about word error rate (WER)

Correct answer: Andy saw a part of the movie

Recognizer output: And he saw apart of the movie

$$WER: \frac{\text{insertions} + \text{deletions} + \text{substitutions}}{\text{true sentence size}} = 4/7 = 57\%$$

- Common issue: intrinsic measures like perplexity are easier to use, but extrinsic ones are more credible

Idea 1: Interpolation

Please close the first door on the left.

4-Gram	3-Gram	2-Gram
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 *	197302 close the window 191125 close the door 152500 close the gap 116451 close the thread … 8662 close the first ----------------- 3785230 close the *	198015222 the first 194623024 the same 168504105 the following 158562063 the world … … ----------------- 23135851162 the *
0.0	0.002	0.009

Specific but Sparse ⟺ Dense but General

(Linear) Interpolation

- Simplest way to mix different orders: linear interpolation

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

  - How to choose lambdas?
  - Should lambda depend on the counts of the histories?
- Choosing weights: either grid search or EM using held-out data
- Better methods have interpolation weights connected to context counts, so you smooth more when you know less

Idea 2: 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	0.45
2	1.25
3	2.24
4	3.23
5	4.21

Absolute Discounting

- Absolute discounting
  - Reduce numerator counts by a constant d (e.g. 0.75)
  - Maybe have a special discount for small counts
  - Redistribute the "shaved" mass to a model of new events
- Example formulation

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

Idea 3: 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

- Kneser-Ney smoothing combines two ideas
  - Discount and reallocate like absolute discounting
  - In the backoff model, word probabilities are proportional to context fertility, not frequency

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

- Theory and practice
  - Practice: KN smoothing has been repeatedly proven both effective and efficient
  - Theory: KN smoothing as approximate inference in a hierarchical Pitman-Yor process [Teh, 2006]

Kneser-Ney Details*

- All orders recursively discount and back-off:

$$P_k(w|prev\ k-1) = \frac{c'(w, prev\ k-1) - d}{\sum_v c'(v, prev\ k-1)} + \alpha(prev\ k-1)P_{k-1}(w|prev\ k\ -2)$$

- Alpha is computed to make the probability normalize (see if you can figure out an expression).
- For the highest order, c' is the token count of the n-gram. For all others it is the context fertility of the n-gram:

$$c'(x) = |\{u : c(u, x) > 0\}|$$

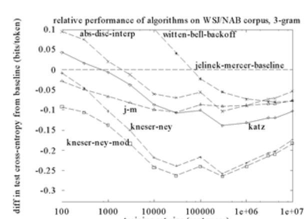
- The unigram base case does not need to discount.
- Variants are possible (e.g. different d for low counts)

Idea 4: Big Data

There's no data like more data.

What Actually Works?

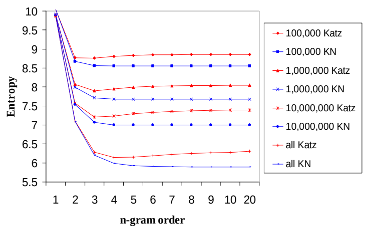
- Trigrams and beyond:
  - Unigrams, bigrams generally useless
  - Trigrams much better
  - 4-, 5-grams and more are really useful in MT, but gains are more limited for speech
- Discounting
  - Absolute discounting, Good-Turing, held-out estimation, Witten-Bell, etc…
- Context counting
  - Kneser-Ney construction of lower-order models
- See [Chen+Goodman] reading for tons of graphs…



[Graph from Joshua Goodman]

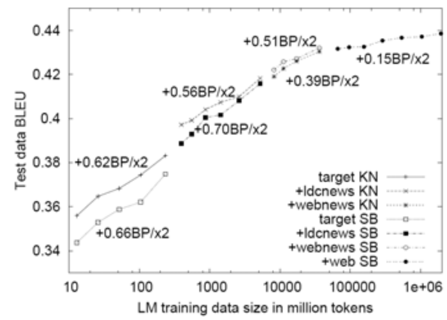
Data >> Method?

- Having more data is better…



- … but so is using a better estimator
- Another issue: N > 3 has huge costs in speech recognizers

Tons of Data?



[Brants et al, 2007]

What about…

Unknown Words?

- What about totally unseen words?
- Most LM applications are 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)
- In principle, one can build open vocabulary LMs
  - E.g. models over character strings rather than words
  - Back-off needs to go down into a "generate new word" model
  - Typically if you need this, a high-order character model is almost as good

Grammar?

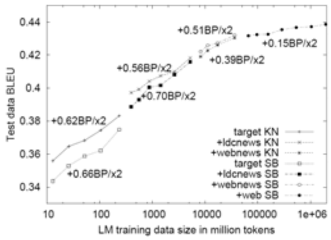
- The N-Gram assumption hurts one's inner linguist!
  - Many linguistic arguments that language isn't regular
    - Long-distance effects: "The computer which I had just put into the machine room on the fifth floor ___."
    - Recursive structure
- Answers
  - N-grams only model local correlations, but they get them all
  - As N increases, they catch even more correlations
  - N-gram models scale much more easily than structured LMs
- Not convinced?
  - Can build LMs out of our grammar models (later in the course)
  - Take any generative model with words at the bottom and marginalize out the other variables

What Gets Captured?

- Bigram model:
  - [texaco, rose, one, in, this, issue, is, pursuing, growth, in, a, boiler, house, said, mr., gurria, mexico, 's, motion, control, proposal, without, permission, from, five, hundred, fifty, five, yen]
  - [outside, new, car, parking, lot, of, the, agreement, reached]
  - [this, would, be, a, record, november]
- PCFG model:
  - [This, quarter, 's, surprisingly, independent, attack, paid, off, the, risk, involving, IRS, leaders, and, transportation, prices, .]
  - [It, could, be, announced, sometime, .]
  - [Mr., Toseland, believes, the, average, defense, economy, is, drafted, from, slightly, more, than, 12, stocks, .]

Scaling Up?

- There’s a lot of training data out there…



… next class we’ll talk about how to make it fit.

Other Techniques?

- Lots of other techniques
  - Maximum entropy LMs (soon)
  - Neural network LMs (soon)
  - Syntactic / grammar-structured LMs (much later)