Lecture 2: Language Models

Roger Levy

多謝 to Dan Klein, Jason Eisner, Joshua Goodman, Stan Chen
Language and probability

- Birth of computational linguistics: 1950s & machine translation
- This was also the birth-time of cognitive science, and an extremely exciting time for mathematics
  - Birth of information theory (Shannon 1948)
  - Birth of formal language theory (Chomsky 1956)
- However, the role of probability and information theory was rapidly dismissed from formal linguistics
  
  ...the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term. (Chomsky 1967)

- (Also see Miller 1957, or Miller 2003 for a retrospective)
Why was the role of probability in formal accounts of language knowledge dismissed?

1. Colorless green ideas sleep furiously.
2. Furiously sleep ideas green colorless.

Chomsky (1957):

- Neither of these sentences has ever appeared before in human discourse
- None of the substrings has either
- Hence the probabilities $P(W_1)$ and $P(W_2)$ cannot be relevant to a native English speaker’s knowledge of the difference between these sentences
But don’t be so sure…
There’s more than one way to color an idea
Language users must be able to generalize beyond their input
Statistical inference is the study of such generalization
Pereira (2000; see also Saul & Pereira, 1997) use a class-based bigram model to model (1) and (2) before:

\[
P(\text{Colorless green ideas sleep furiously}) \approx 2 \times 10^5
\]

Maybe probabilities aren’t such a useless model of language knowledge after all
A class-based model looks something like this:

The $c_i$ are unseen variables that have to be introduced into the model, either through:
- Annotation
- Unsupervised learning

We’ll cover these later in the course

For now, we’ll focus on the basic problem of the language model: $P(W)$
The uses of language models

- Once we have a language model—$P(W)$—what can we do with it???
- Answer: Language models are core component of many applications where task is to recover an utterance from an input
- This is done through Bayes’ rule

$$P(\text{utterance}|\text{input}) = \frac{P(\text{input}|\text{utterance})P(\text{utterance})}{P(\text{input})}$$

$\propto P(\text{input}|\text{utterance})P(\text{utterance})$
Some practical uses of language models...
The Speech Recognition Problem

- We want to predict a sentence given an acoustic sequence:

\[ s^* = \arg \max_s P(s | a) \]

- The noisy channel approach:
  - Build a generative model of production (encoding)

\[ P(a, s) = P(s)P(a | s) \]

- To decode, we use Bayes’ rule to write

\[ s^* = \arg \max_s P(s | a) = \arg \max_s P(s)P(a | s) / P(a) = \arg \max_s P(s)P(a | s) \]

- Now, we have to find a sentence maximizing this product

- Why is this progress?
MT System Components

Language Model

source
P(e)

best
e

e

decoder

argmax P(e|f) = argmax P(f|e)P(e)
Computational Psycholinguistics

- Language models (structured & otherwise) can help us theorize about the way probabilistic expectations influence sentence comprehension & production (later in the course)
Other Noisy-Channel Processes

- Handwriting recognition
  \[ P(\text{text} | \text{strokes}) \propto P(\text{text})P(\text{strokes} | \text{text}) \]

- OCR
  \[ P(\text{text} | \text{pixels}) \propto P(\text{text})P(\text{pixels} | \text{text}) \]

- Spelling Correction

- Translation
  \[ P(\text{text} | \text{typos}) \propto P(\text{text})P(\text{typos} | \text{text}) \]

  \[ P(\text{english} | \text{french}) \propto P(\text{english})P(\text{french} | \text{english}) \]
Uses of language models, summary

- So language models are both practically and theoretically useful!
- Now, how do we estimate them and use them for inference?
- We’ll kick off this course by looking at the simplest type of model
Probabilistic Language Models

• Want to build models which assign scores to sentences.
  • $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$
  • Not really grammaticality: $P(\text{artichokes intimidate zippers}) \approx 0$

• One option: empirical distribution over sentences?
  • Problem: doesn’t generalize (at all)

• Two major components of generalization
  • **Backoff**: sentences generated in small steps which can be recombined in other ways
  • **Discounting**: allow for the possibility of unseen events
N-Gram Language Models

- No loss of generality to break sentence probability down with the chain rule

\[ P(w_1w_2 \ldots w_n) = \prod_i P(w_i \mid w_1w_2 \ldots w_{i-1}) \]

- Too many histories!
  - \( P(??? \mid \text{No loss of generality to break sentence}) \)
  - \( P(??? \mid \text{the water is so transparent that}) \)

- N-gram solution: assume each word depends only on a short linear history

\[ P(w_1w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1}) \]
Unigram Models

- Simplest case: unigrams

\[ P(w_1 w_2 \ldots w_n) = \prod P(w_i) \]

- Generative process: pick a word, pick a word, …

- As a graphical model:

To make this a proper distribution over sentences, we have to generate a special STOP symbol last. (Why?)

Examples:
- [fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass.]
- [thrift, did, eighty, said, hard, 'm, july, bullish]
- [that, or, limited, the]
- [after, any, on, consistently, hospital, lake, of, of, other, and, factors, raised, analyst, too, allowed, mexico, never, consider, fall, bungled, davison, that, obtain, price, lines, the, to, sass, the, the, further, board, a, details, machinists, the, companies, which, rivals, an, because, longer, oakes, percent, a, they, three, edward, it, currier, an, within, in, three, wrote, is, you, s., longer, institute, dentistry, pay, however, said, possible, to, rooms, hiding, eggs, approximate, financial, canada, the, so, workers, advancers, half, between, nasdaq]
Bigram Models

- Big problem with unigrams: $P(\text{the the the the}) \gg P(\text{I like ice cream})!$
- Condition on previous word:

$$P(w_1w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-1})$$

- Any better?
  - [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]
  - [although, common, shares, rose, forty, six, point, four, hundred, dollars, from, thirty, seconds, at, the, greatest, play, disingenuous, to, be, reset, annually, the, buy, out, of, american, brands, vying, for, mr., womack, currently, sharedata, incorporated, believe, chemical, prices, undoubtedly, will, be, as, much, is, scheduled, to, conscientious, teaching]
  - [this, would, be, a, record, november]
More N-Gram Examples

- You can have trigrams, quadrigrams, etc.
Regular Languages?

- N-gram models are (weighted) regular processes
  - Why can’t we model language like this?
    - Linguists have many arguments why language can’t be merely regular.
    - Long-distance effects:
      “The computer which I had just put into the machine room on the fifth floor crashed.”
  - Why CAN we often get away with n-gram models?

- PCFG language models (hopefully later in the course):
  - [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, .]
Evaluation

• What we want to know is:
  • Will our model prefer good sentences to bad ones?
    • That is, does it assign higher probability to “real” or “frequently observed” sentences than “ungrammatical” or “rarely observed” sentences?
  • As a component of Bayesian inference, will it help us discriminate correct utterances from noisy inputs?
Measuring Model Quality

• For Speech: Word Error Rate (WER) \[ \frac{\text{insertions} + \text{deletions} + \text{substitutions}}{\text{true sentence size}} \]

Correct answer: Andy saw a part of the movie
Recognizer output: And he saw apart of the movie

• The “right” measure:
  • Task error driven
  • For speech recognition
  • For a specific recognizer!  \[ \text{WER: } \frac{4}{7} = 57\% \]

• For general evaluation, we want a measure which references only good text, not mistake text
Measuring Model Quality

- The Shannon Game:
  - How well can we predict the next word?
    - When I order pizza, I wipe off the ____
    - Many children are allergic to ____
    - I saw a ____
  - Unigrams are terrible at this game. (Why?)

- The “Entropy” Measure
  - Really: average cross-entropy of a text according to a model

\[
H(S \mid M) = \log_2 \frac{P_M(S)}{|S|} = \frac{\sum \log_2 P_M(s_i)}{\sum |s_i|} \sum \log_2 P_M(w_j \mid w_{j-1})
\]

- grease 0.5
- sauce 0.4
- dust 0.05
- mice 0.0001
- the 1e-100
Measuring Model Quality

- Problem with entropy:
  - 0.1 bits of improvement doesn’t sound so good
  - Solution: perplexity

\[ P(S \mid M) = 2^{H(S \mid M)} = \sqrt[n]{\prod_{i=1}^{n} P_M(w_i \mid h)} \]

- Minor technical note: even though our models require a stop step, people typically don’t count it as a symbol when taking these averages.
Sparsity

- Problems with n-gram models:
  - New words appear all the time:
    - Synaptitude
    - 132,701.03
    - fuzzificational
  - New bigrams: even more often
  - Trigrams or more – still worse!

- Zipf’s Law
  - Types (words) vs. tokens (word occurrences)
  - 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!)
Smoothing

• We often want to make estimates from sparse statistics:

\[
P(w \mid \text{denied the}) \\
3 \text{ allegations} \\
2 \text{ reports} \\
1 \text{ claims} \\
1 \text{ request} \\
7 \text{ total}
\]

• Smoothing flattens spiky distributions so they generalize better

\[
P(w \mid \text{denied the}) \\
2.5 \text{ allegations} \\
1.5 \text{ reports} \\
0.5 \text{ claims} \\
0.5 \text{ request} \\
2 \text{ other} \\
7 \text{ total}
\]

• Very important all over NLP, but easy to do badly!
• Illustration with bigrams (h = previous word, could be anything).
Smoothing

- Estimating multinomials
  - We want to know what words follow some history $h$
  - There’s some true distribution $P(w | h)$
  - We saw some small sample of $N$ words from $P(w | h)$
  - We want to reconstruct a useful approximation of $P(w | h)$
  - Counts of events we didn’t see are always too low ($0 < N P(w | h)$)
  - Counts of events we did see are in aggregate too high

- Example:

  $P(w \mid \text{denied the})$
  - 3 allegations
  - 2 reports
  - 1 claims
  - 1 speculation
  - ...
  - 1 request
  - 13 total

  $P(w \mid \text{affirmed the})$
  - 1 award

- Two issues:
  - Discounting: how to reserve mass what we haven’t seen
  - Interpolation: how to allocate that mass amongst unseen events