Lecture 1: Introduction & Language Modeling

Ankur Parikh – Google

Many thanks to Slav Petrov (Google) and Dan Klein (UC Berkeley) for slides and class materials
Course Details

- **Reference Books:**
  - Jurafsky and Martin, Speech and Language Processing, 2 Edition
  - 3rd Edition draft chapters available online: http://web.stanford.edu/~jurafsky/slp3/
  - Manning and Schuetze, Foundations of Statistical NLP

- **Additional readings from recent papers**

- **Prerequisites:**
  - Solid math background
  - Solid programming skills (Most assignments will be in Java)
  - Deep interest in language
Announcements

- Website:
  - http://cs.nyu.edu/courses/fall17/CSCI-GA.3033-008/

- Communication:
  - Announcements and public discussion: Piazza!
    https://piazza.com/nyu/fall2017/csciga3033008/home
  - My email is aparikh@cs.nyu.edu, but use Piazza whenever possible.

- Computing Resources
  - You might want more compute power than your laptop
  - Experiments will take minutes to hours, with efficient code
  - Recommendation: start assignments early
Work and Grading

- Class participation (5%)

- Scribe Notes (10%)
  - You will be create notes for one lecture. Sign up on the website.

- Five homework (programming+written) assignments (50%)

- Final project (35%)
  - Can be done in groups
AI: Where do we stand?

1980
1990
2000
2010
2020

Hollywood

R2D2

KITT

Wall-E

Rule based approaches

Early statistical approaches

Modern statistical approaches

Self-Driving Cars

Phone Assistants

Reality
What is NLP?

- Fundamental goal: *deep* understand of *broad* language
  - Not just string processing or keyword matching!

- End systems that we want to build:
  - Simple: spelling correction, text categorization…
  - Complex: speech recognition, machine translation, information extraction, dialog interfaces, question answering…
  - Unknown: human-level comprehension (is this just NLP?)
Speech Systems

- Automatic Speech Recognition (ASR)
  - Audio in, text out

- Text to Speech (TTS)
  - Text in, audio out

“Speech Lab”
Machine Translation

- Translate text from one language to another
- Recombines fragments of example translations
- Challenges:
  - What fragments?  [learning to translate]
  - How to make efficient?  [fast translation search]
  - Fluency (this class) vs fidelity (later)
New York Times Co. named Russell T. Lewis, 45, president and general manager of its flagship New York Times newspaper, responsible for all business-side activities. He was executive vice president and deputy general manager. He succeeds Lance R. Primis, who in September was named president and chief operating officer of the parent.

<table>
<thead>
<tr>
<th>Person</th>
<th>Company</th>
<th>Post</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russell T. Lewis</td>
<td>New York Times newspaper</td>
<td>president and general manager</td>
<td>start</td>
</tr>
<tr>
<td>Russell T. Lewis</td>
<td>New York Times newspaper</td>
<td>executive vice president</td>
<td>end</td>
</tr>
<tr>
<td>Lance R. Primis</td>
<td>New York Times Co.</td>
<td>president and CEO</td>
<td>start</td>
</tr>
</tbody>
</table>
Question Answering

- Question Answering:
  - More than search
  - Ask general comprehension questions of a document collection
  - Can be really easy: “What’s the capital of Wyoming?”
  - Can be harder: “How many US states’ capitals are also their largest cities?”
  - Can be open ended: “What are the main issues in the global warming debate?”
"The rock was still wet. The animal was glistening, like it was still swimming," recalls Hou Xianguang. Hou discovered the unusual fossil while surveying rocks as a paleontology graduate student in 1984, near the Chinese town of Chengjiang. "My teachers always talked about the Burgess Shale animals. It looked like one of them. My hands began to shake." Hou had indeed found a Naraoia like those from Canada. However, Hou's animal was 15 million years older than its Canadian relatives.

It can be inferred that Hou Xianguang's "hands began to shake", because he was:

(A) afraid that he might lose the fossil
(B) worried about the implications of his finding
(C) concerned that he might not get credit for his work
(D) uncertain about the authenticity of the fossil
(E) excited about the magnitude of his discovery
Conversational Search

who is the chef at The French Laundry

The French Laundry is a French restaurant located in Yountville, California, in the Napa Valley. The chef and owner of the French Laundry is Thomas Keller. The restaurant building dates from 1900, and is in the National Register of Historic Places.

what are its hours

5:30 – 9:15 pm
The French Laundry, Thursday hours

what books has he written

Thomas Keller
Books
The French Laundry Cookbook
1999
An Odflm And Mc-cdma Primer
2006
Single and Multicarrier Modulation
2000
Angels: The Lifting of the Veil
1994
Computers for Managing Information Stud... 1988

[live on your phone]
Turkey!

dcorrado
to me

Hi all,
We wanted to invite you to join us for an early Thanksgiving on November 22nd, beginning around 2PM. Please bring your favorite dish! RSVP by next week.

Dave

Server issues

Dan Mané
to me

Hi team,
The server appears to be dropping about 10% of requests (see attached dashboards). There hasn't been a new release since last night, so I'm not sure what's going on. Is anyone looking into this?

***

Reply

Bad suggestions?

Count us in! We'll be there! Sorry, we won't be able to make it.

I'll check on it. I'll see if I can find out. I'm on it.
Summarization

- Condensing documents
  - Single or multiple
  - Extractive or synthetic
  - Aggregative or representative
  - Even just shortening sentences
- Very context-dependent!
- An example of analysis with generation
The Good News

- Language isn’t adversarial:
  - It’s produced with the intent of being understood
  - With some understanding of language, you can often tell what knowledge sources are relevant

- In many tasks, can often achieve relatively high accuracy without a deep understanding of the meaning of the text
  - Text classification
  - Sentiment analysis
But NLP is Hard

- Some core problems:
  - Sparsity
  - Long range dependencies
  - Multilinguality
  - Difficulty of Annotation
  - Ambiguity / common sense reasoning
Sparsity

- Many linguistic phenomena follow power law curve

- Example (Word frequency):
  - Common words occur a lot.
  - But many many words appear much fewer times
Hurricane Emily howled toward Mexico's Caribbean coast on Sunday packing 135 mph winds and torrential rain and causing panic in Cancun, where frightened tourists squeezed into musty shelters.

What is causing panic?: Hurricane Emily

Syntax tree: A representation to transform long range dependencies into local dependencies.
Multilinguality

- Much more data in English than in any other language.
- English is the most common known language among NLP researchers.
- But so many other languages are spoken!
Difficulty of Annotation

- Obtaining detailed annotation for linguistic phenomena (e.g. syntax trees) is difficult and cannot easily be gathered at scale.

- On the other hand, weakly labelled data is often more readily available.

- Unsupervised data is abundant
Ambiguity / Common Sense

- **Headlines:**
  - Hospitals Are Sued by 7 Foot Doctors
  - Iraqi Head Seeks Arms
  - Enraged Cow Injures Farmer with Ax
  - Ban on Nude Dancing on Governor’s Desk
  - Stolen Painting Found by Tree
  - Teacher Strikes Idle Kids
  - Kids Make Nutritious Snacks
  - Local HS Dropouts Cut in Half

- **Why are these funny?**
Corpora

- A corpus is a collection of text
  - Annotated in some way: supervised learning
  - Sometimes just lots of text without any annotations: unsupervised learning

- Examples
  - Newswire collections: 500M+ words
  - Brown corpus: 1M words of tagged “balanced” text
  - Penn Treebank: 1M words of parsed WSJ
  - Canadian Hansards: 10M+ words of aligned French / English sentences
  - The Web: billions of words of who knows what
Experimental Setup

Hold Out Set:

- Train
- Train
- Train
- Train

Cross Validation:

- Train
- Train
- Train
- Train
What is this Class?

Three aspects to the course:

- **Linguistic Issues**
  - What are the range of language phenomena?
  - What are the knowledge sources that let us disambiguate?
  - What representations are appropriate?
  - How do you know what to model and what not to model?

- **Statistical Modeling Methods**
  - Increasingly complex model structures
  - Learning and parameter estimation
  - Efficient inference: dynamic programming, search, sampling

- **Engineering Methods**
  - Issues of scale
  - Where the theory breaks down (and what to do about it)
Class Requirements and Goals

- **Class requirements**
  - Uses a variety of skills / knowledge:
    - Probability and statistics
    - Decent coding skills, knowledge of data structures
  - Most people are probably missing one of the above
  - You will often have to work on your own to fill the gaps

- **Class goals**
  - Learn the issues and techniques of statistical NLP
  - Build realistic tools used in NLP (language models, taggers, parsers, translation systems)
  - Be able to read current research papers in the field
  - See where the holes in the field still are!
The (Effective) NLP Cycle

- Pick a problem (usually some disambiguation)
- Get a lot of data (hopefully with some labels)
- Build the simplest thing that could possibly work
- Repeat:
  - Examine the most common errors are
  - Figure out what information a human might use to avoid them
  - Modify the system to exploit that information
    - Feature engineering
    - Representation redesign
    - Different machine learning methods
  - Revisit problem/data as needed
- We’re going to do this over and over again
Some Disclaimers

- The purpose of this class is to train NLP researchers
  - Some people will put in a lot of time
  - There will be a lot of reading, some required, some not – you will have to be strategic about what reading enables your goals
  - There will be a lot of coding and running systems on substantial amounts of real data
  - There will be a lot of statistical modeling (though we do use a few basic techniques very heavily)
  - There will be discussion and questions in class that will push past what I present in lecture, and I’ll answer them
  - Not everything will be spelled out for you in the assignments

- Don’t say I didn’t warn you!
Outline of Topics

- **Words**
  - N-gram models and smoothing
  - Classification and clustering

- **Sequences**
  - Part-of-speech tagging
  - Word-alignments

- **Trees**
  - Syntax
  - Semantics
  - Machine translation

- **Sentiment Analysis**

- **Summarization**

- Generative Models
- Discriminative Models
- Graphical Models
- Neural Networks
Speech in a Slide

- Frequency gives pitch; amplitude gives volume

- Frequencies at each time slice processed into observation vectors
The Noisy-Channel Model

- We want to predict a sentence given acoustics:
  \[ w^* = \arg\max_w P(w|a) \]

- The noisy channel approach (Bayes rule):
  \[
  w^* = \arg\max_w P(w|a) \\
  = \arg\max_w P(w, a)/P(a) \\
  = \arg\max_w P(a|w)P(w)/P(a) \\
  = \arg\max_w P(a|w)P(w) \\
\]

acoustic model
language model
**Machine Translation**

- Given a French sentence $f$, find an English sentence $e$.

- The noisy channel approach (Bayes rule):

\[
e^* = \arg\max_e P(e | f)
= \arg\max_e \frac{P(e, f)}{P(f)}
= \arg\max_e \frac{P(f | e) P(e)}{P(f)}
= \arg\max_e P(f | e) P(e)
\]
Other Noisy-Channel Processes

- **Spelling Correction**
  \[ P(\text{words} \mid \text{characters}) \propto P(\text{words})P(\text{characters} \mid \text{words}) \]

- **Handwriting recognition**
  \[ P(\text{words} \mid \text{strokes}) \propto P(\text{words})P(\text{strokes} \mid \text{words}) \]

- **OCR**
  \[ P(\text{words} \mid \text{pixels}) \propto P(\text{words})P(\text{pixels} \mid \text{words}) \]

- **More...**
Probabilistic Language Models

- **Goal:** Assign useful probabilities \( P(x) \) to sentences \( x \)
  - Input: many observations of training sentences \( x \)
  - Output: system capable of computing \( P(x) \)

- Probabilities should broadly indicate plausibility of sentences
  - \( P(\text{I saw a van}) >> P(\text{eyes awe of an}) \)
  - In principle, “plausible” depends on the domain, context, speaker…

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

- Two aspects of generalization
  - Decomposition: break sentences into small pieces which can be recombined in new ways (conditional independence)
  - Smoothing: allow for the possibility of unseen pieces
N-Gram Model Decomposition

- Chain rule: break sentence probability down
  \[ P(w_1 \ldots w_n) = \prod_i P(w_i|w_1 \ldots w_{i-1}) \]

- Impractical to condition on everything before
  - \( P(??? \mid \text{Turn to page 134 and look at the picture of the}) \)

- N-gram models: assume each word depends only on a short linear history
  \[ P(w_1 \ldots w_n) = \prod_i P(w_i|w_{i-k} \ldots w_{i-1}) \]

- Example:
  \[ P(\text{please close the door}) = P(\text{please}|\text{START})P(\text{close}|\text{please}) \ldots P(\text{STOP}|\text{door}) \]
N-Gram Model Parameters

- The parameters of an n-gram model:
  - The actual conditional probability estimates, we’ll call them $\theta$
  - Obvious estimate: relative frequency (maximum likelihood) estimate

\[
\hat{P}(w|w_{-1}) = \frac{c(w_{-1}, w)}{\sum_{w'} c(w_{-1}, w')}
\]

- General approach
  - Take a training set $X$ and a test set $X'$
  - Compute an estimate $\theta$ from $X$
  - Use it to assign probabilities to other sentences, such as those in $X'$

Training Counts

- 198015222 the first
- 194623024 the same
- 168504105 the following
- 158562063 the world
- ...
- 14112454 the door

\[
\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162} = 0.0006
\]
Higher Order N-grams?

Please close the door

Please close the first window on the left

---

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

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

-----------------

3380 please close the door
1601 please close the window
1164 please close the new
1159 please close the gate
900 please close the browser

-----------------

3785230 close the *

13951 please close the *
Unigram Models

- Simplest case: unigrams

\[ P(w_1 \ldots w_n) = \prod_i P(w_i) \]

- 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 single word:

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

- Obvious that this should help – in probabilistic terms, we’re using weaker conditional independence assumptions (what’s the cost?)
- 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]
The game isn’t to pound out fake sentences!
- Obviously, generated sentences get “better” as we increase the model order
- More precisely: using ML estimators, higher order is always better likelihood on train, but not test

What we really want to know is:
- Will our model prefer good sentences to bad ones?
- Bad ≠ ungrammatical!
- Bad ≈ unlikely
- Bad = sentences that our acoustic model really likes but aren’t the correct answer
Measuring Model Quality

- **The Shannon Game:**
  - How well can we predict the next word?
    
    When I eat pizza, I wipe off the ____
    Many children are allergic to ____
    I saw a ____

  - Unigrams are terrible at this game. (Why?)

- “Entropy”: per-word test

\[
H(X|\theta) = -\frac{1}{|X|} \sum_{x \in X} \log_2 P(x|\theta)
\]

\[
\sum_{x \in X} |x| \sum_{i} \log P(x_i|x_{i-1}, \theta)
\]

- 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 *
Measuring Model Quality

- More common measure is to exponentiate the entropy, called “perplexity”

\[
\text{perp}(X, \theta) = 2^{H(X|\theta)}
\]

- Important notes:
  - It’s easy to get bogus perplexities by having bogus probabilities that sum to more than one over their event spaces.
Measuring Model Quality

- **Word Error Rate (WER)**

  \[
  \text{WER} = \frac{\text{insertions} + \text{deletions} + \text{substitutions}}{\text{true sentence size}}
  \]

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

  \( \text{WER: 4/7} \)

  \( = 57\% \)

- **The “right” measure:**
  - Task error driven
  - For speech recognition
  - For a specific recognizer!

- **Common issue:** Intrinsic measures like perplexity are easier to use, but extrinsic ones are more credible.
Problems with n-gram models:
- New words appear all the time:
  - Synaptitute
  - 132,701.03
  - multidisciplinarization
- 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!)
Parameter Estimation

- Maximum likelihood estimates won’t get us very far

\[ \hat{P}(w|w_{-1}) = \frac{c(w_{-1}, w)}{\sum_{w'} c(w_{-1}, w')} \]

- Need to *smooth* these estimates

- General method (procedurally)
  - Take your empirical counts
  - Modify them in various ways to improve estimates

- General method (mathematically)
  - Often can give estimators a formal statistical interpretation
  - … but not always
  - Approaches that are mathematically obvious aren’t always what works

---

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 *
Smoothing

- We often want to make estimates from sparse statistics:
  
  \[ P(w \mid \text{denied the}) \]
  
  3 allegations
  2 reports
  1 claims
  1 request
  7 total

- Smoothing flattens spiky distributions so they generalize better

  \[ P(w \mid \text{denied the}) \]
  
  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! We’ll illustrate with bigrams today (h = previous word, could be anything).
Smoothing: Add-One, Etc.

- Classic solution: add counts (Laplace smoothing / Dirichlet prior)

\[ P_{\text{add-}\delta}(x) = \frac{c(x) + \delta}{\sum_{x'}(c(x') + \delta)} \]

- Add-one smoothing especially often talked about

- For a bigram distribution, can add counts shaped like the unigram:

\[ P_{\text{dir}}(w|w_{-1}) = \frac{c(w_{-1}, w) + k\hat{P}(w)}{(\sum_{w'} c(w_{-1}, w')) + k} \]

- Can consider hierarchical formulations: trigram is recursively centered on smoothed bigram estimate, etc [MacKay and Peto, 94]

- Can be derived from Dirichlet / multinomial conjugacy: prior shape shows up as *pseudo-counts*

- Problem: works quite poorly!
Held-Out Reweighting

- What’s wrong with add-d smoothing?
- Let’s look at some real bigram counts [Church and Gale 91]:

<table>
<thead>
<tr>
<th>Count in 22M Words</th>
<th>Actual c* (Next 22M)</th>
<th>Add-one’s c*</th>
<th>Add-0.0000027’s c*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.448</td>
<td>2/7e-10</td>
<td>~1</td>
</tr>
<tr>
<td>2</td>
<td>1.25</td>
<td>3/7e-10</td>
<td>~2</td>
</tr>
<tr>
<td>3</td>
<td>2.24</td>
<td>4/7e-10</td>
<td>~3</td>
</tr>
<tr>
<td>4</td>
<td>3.23</td>
<td>5/7e-10</td>
<td>~4</td>
</tr>
<tr>
<td>5</td>
<td>4.21</td>
<td>6/7e-10</td>
<td>~5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mass on New</th>
<th>~100%</th>
<th>9.2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of 2/1</td>
<td>2.8</td>
<td>1.5</td>
</tr>
</tbody>
</table>

- Big things to notice:
  - Add-one vastly overestimates the fraction of new bigrams
  - Add-anything vastly underestimates the ratio 2*/1*
Linear Interpolation

- Problem: \( \hat{P}(w|w_{-1}, w_{-2}) \) is supported by few counts
- Classic solution: mixtures of related, denser histories, e.g.:
  \[
  \lambda \hat{P}(w|w_{-1}, w_{-2}) + \lambda' \hat{P}(w|w_{-1}) + \lambda'' \hat{P}(w)
  \]
- The mixture approach tends to work better than the add-one approach for several reasons
  - Can flexibly include multiple back-off contexts, not just a chain
  - Often multiple weights, depending on bucketed counts
  - Good ways of learning the mixture weights with EM (later)
  - Not entirely clear why it works so much better

- All the details you could ever want: [Chen and Goodman, 98]
Problem with Linear Interpolation

- Observed n-grams occur more in training than they will later:

<table>
<thead>
<tr>
<th>Count in 22M words</th>
<th>Avg in next 22M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.448</td>
</tr>
<tr>
<td>2</td>
<td>1.25</td>
</tr>
<tr>
<td>3</td>
<td>2.24</td>
</tr>
<tr>
<td>4</td>
<td>3.23</td>
</tr>
</tbody>
</table>

- But the discount is relatively “constant” i.e. not a function of n-gram frequency.

- But in linear interpolation, high counts are effectively penalized more than low counts since all the counts are scaled by the same factor.
Absolute Discounting

- Just subtract a constant from all the counts. Assign leftover probability to the lower order model.

\[ P_{ad}(w|w') = \frac{\max(c(w', w) - d, 0)}{\sum_w c(w', w)} + \alpha(w') P(w) \]

\[ \alpha(w') = \frac{d \times |\{w : c(w', w) > 0\}|}{\sum_w c(w', w)} \]
Consider the following property for the Maximum likelihood conditional distribution:

\[
\hat{P}(w) = \sum_{w'} \hat{P}(w|w') P(w')
\]

However, this property doesn’t hold for absolute discounting.

\[
\hat{P}(w) \neq \sum_{w'} P_{ad}(w|w') P(w')
\]

Absolute discounting biases the unigram distribution away from the maximum likelihood estimate.
Kneser-Ney Smoothing

- Intuition: Lower order distribution needs to be “altered” to preserve the marginal constraint.

- Diversity of history:
  - *York* is a popular word, but only follows very few words i.e. *New York* (high frequency, but low diversity)
  - (assume that) *crayon* is a less popular word, but can follow many words i.e. red crayon, blue crayon

- Thus, typically

\[ \hat{P}(\text{York}) > \hat{P}(\text{crayon}) \]

- But really the unigram probability will only be used when the bigram counts were low. In this case Kneser New will alter the unigram distribution such that:

\[ \hat{P}_{kn}(\text{York}) < \hat{P}_{kn}(\text{crayon}) \]
Kneser-Ney Derivation

\[ P_{kn}(w|w') = \frac{\max(c(w', w) - d, 0)}{\sum_w c(w', w)} + \alpha(w')P_{kn}(w) \]

Design lower order distribution so that constraint holds

\[ \hat{P}(w) = \sum_{w'} P_{kn}(w|w')P(w') \]

Equivalent to:

\[ c(w) = \sum_{w'} P_{kn}(w|w')c(w') \]
Kneser-Ney Derivation

Notation:

\[ N_{1+}(w', \cdot) := |\{w : c(w', w) > 0\}| \]

\[ N_{1+}(\cdot, w) := |\{w' : c(w', w) > 0\}| \]

\[ N_{1+}(\cdot, \cdot) := \sum_{w'} N_{1+}(w', \cdot) \]

Claim:

\[ P_{kn}(w) = \frac{N_{1+}(\cdot, w)}{N_{1+}(\cdot, \cdot)} \]
Kneser-Ney Derivation

\[
c(w) = \sum_{w'} c(w') \left( \frac{\max(c(w', w) - d, 0)}{\sum_w c(w', w)} \right) + \frac{d}{\sum_w c(w', w)} N_{1+}(w', \cdot) \frac{\sum_w c(w', w)}{N_{1+}(w', \cdot)} P_{kn}(w)
\]

\[
= \sum_{w'} c(w') \frac{\max(c(w', w) - d, 0)}{c(w')} + \sum_{w'} c(w') \frac{d}{c(w')} N_{1+}(w', \cdot) P_{kn}(w)
\]

\[
= c(w) - N_{1+}(\cdot, w) d + d \times P_{kn}(w) \times \sum_{w'} N_{1+}(w', \cdot)
\]

\[
= c(w) - N_{1+}(\cdot, w) d + d P_{kn}(w) N_{1+}(\cdot, \cdot)
\]

Solving for \( P_{kn}(w) \) gives the desired result
Kneser-Ney

- Absolute discounting

\[ P_{kn}(w|w') = \frac{\max(c(w', w) - d, 0)}{\sum_w c(w', w)} + \alpha(w')P_{kn}(w) \]

- Lower order continuation probabilities

\[ P_{kn}(w) = \frac{N_{1+}(\cdot, w)}{N_{1+}(\cdot, \cdot)} \propto |\{w' : c(w', w) > 0\}| \]

measures “diversity” of histories

- KN smoothing repeatedly proven effective (ASR, MT, …)
What Actually Works?

- **Trigrams and beyond:**
  - Unigrams, bigrams generally useless
  - Trigrams much better (when there’s enough data)
  - 4-, 5-grams really useful in MT, but not so much for speech

- **Discounting**
  - Absolute discounting, Good-Turing, held-out estimation, Witten-Bell

- **Context counting**
  - Kneser-Ney construction of lower-order models

- See [Chen+Goodman] reading for tons of graphs!

[Graphs from Joshua Goodman]
Having more data is better…

… but so is using a better estimator

Another issue: N > 3 has huge costs in speech recognizers
Beyond N-Gram LMs

- Stay tune for neural language models [Bengio et al. 03, Mikolov et al. 11]

- Lots of ideas we won’t have time to discuss:
  - Caching models: recent words more likely to appear again
  - Topic models

- A few other (not so recent) ideas
  - Syntactic models: use tree models to capture long-distance syntactic effects [Chelba and Jelinek, 98]
  - Discriminative models: set n-gram weights to improve final task accuracy rather than fit training set density [Roark 05, for ASR; Liang et. al. 06, for MT]
  - Compressed LMs [Pauls & Klein 11, Heafield 11]
What’s Next?

- Next Topic: Classification
  - Naive Bayes vs. Maximum Entropy vs. Neural Networks
  - We introduce a single new global variable
  - Still a very simplistic model family
  - Lets us model hidden properties of text, but only very non-local ones…
  - In particular, we can only model properties which are largely invariant to word order (like topic)

- If you are not fully comfortable with conditional probabilities and maximum likelihood estimators are, read up!

- Reading on the web
- Assignment 1 is already out, due next week!