Language Model Smoothing, Google N-Gram Corpus, Mutual Information

Using n-gram probabilities in a language model: Why do we need smoothing?

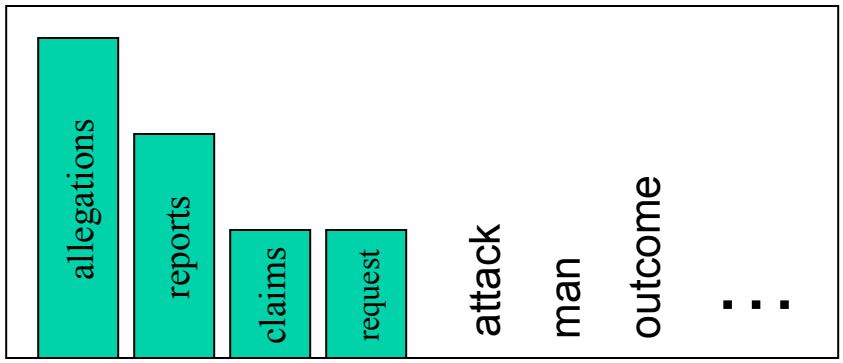
- Every N-gram training matrix is sparse, even for very large corpora (remember Zipf' s law)
  - There are words that don't occur in the training corpus that may occur in future text
  - These are known as the unseen words
- Whenever a probability is 0, it will multiply the entire sequence to be 0

- Solution: estimate the likelihood of unseen N-grams and include a small probability for unseen words

Intuition of smoothing (from Dan Klein)

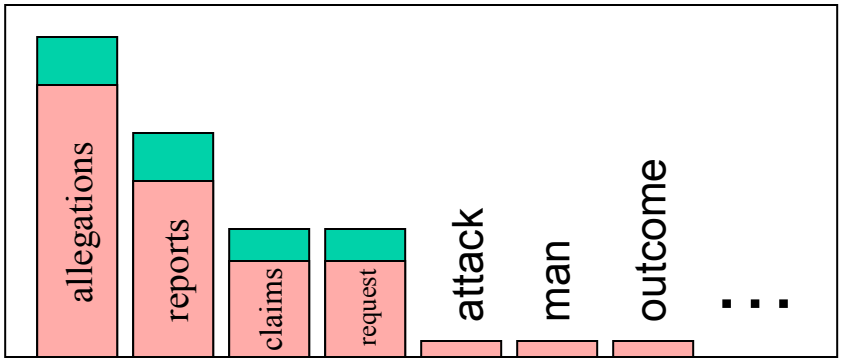
- When we have sparse statistics:

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



Smoothing

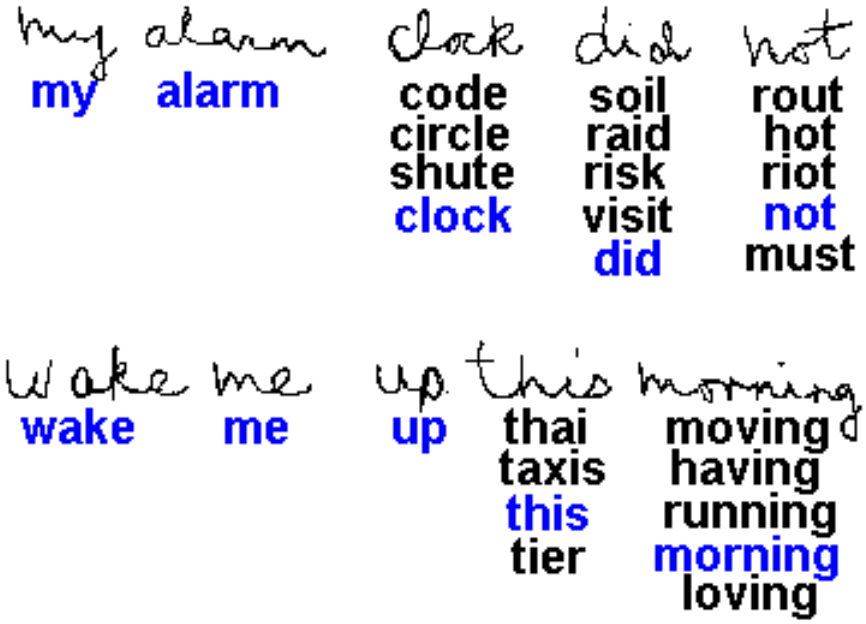
- Add-one smoothing
  - Given: $P(w_n|w_{n-1}) = C(w_{n-1}w_n)/C(w_{n-1})$
  - Add 1 to each count: $P(w_n|w_{n-1}) = [C(w_{n-1}w_n) + 1] / [C(w_{n-1}) + V]$
- Backoff Smoothing for higher-order N-grams
  - Notice that:
    - N-grams are more precise than (N-1)grams
    - But also, N-grams are more sparse than (N-1) grams
  - How to combine things?
    - Attempt N-grams and back-off to (N-1) if counts are not available
    - E.g. attempt prediction using 4-grams, and back-off to trigrams (or bigrams, or unigrams) if counts are not available
- More complicated techniques exist: in practice, NLP LM use Knesser-Ney smoothing

N-gram Model Application - Spell Correction

- Frequency of spelling errors in human typed text varies
  - 0.05% of the words in carefully edited journals
  - 38% in difficult applications like telephone directory lookup
- Word-based spell correction checks each word in a dictionary/lexicon
  - Detecting spelling errors that result in non-words
  - *mesage -> message* by looking only at the **word** in isolation
  - May fail to recognize an error (**real-word errors**)
    - Typographical errors e.g. *there* for *three*
    - Homonym or near-homonym e.g. *dessert* for *desert*, or *piece* for *peace*
- Use context of preceding word and language model to choose correct word
  - *P(Japanese Imperial Navy) > P(Japanese Empirical Navy)*

N-gram Model Analysis of Handwritten Sentence

- Optical character recognition has higher **error** rates than human typists
- Lists of up to top 5 choices of the handwritten word recognizer, with correct choice highlighted
- Using language models with bigram probabilities (*alarm clock*) & syntactic (POS) information, correct sentence is extracted:



Language Modeling Toolkit

- SRI Language Modeling:
  - http://www.speech.sri.com/projects/srilm/

More on Corpus Statistics

- Recall that so far, we have primarily looked at the measure involving bigrams:
  - Bigram probability – conditional probability that the second word follows the first word in the corpus
- The Google n-gram viewer shows n-gram frequencies over time in a collection of books
  - Bigram frequency – percentage occurrence of the bigram in the corpus
    - Count of a bigram / total number of bigrams in corpus

Google N-Gram Release

All Our N-gram are Belong to You

By Peter Norvig - 8/03/2006 11:26:00 AM

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects, such as statistical machine translation, speech recognition, spelling correction, entity detection, information extraction, and others. While such models have usually been estimated from training

to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Example Data

- Examples of 4-gram frequencies from the Google N-gram release
  - `serve as the incoming 92`
  - `serve as the incubator 99`
  - `serve as the independent 794`
  - `serve as the index 223`
  - `serve as the indication 72`
  - `serve as the indicator 120`
  - `serve as the indicators 45`
  - `serve as the indispensable 111`
  - `serve as the indispensible 40`
  - `serve as the individual 234`

Google n-gram viewer

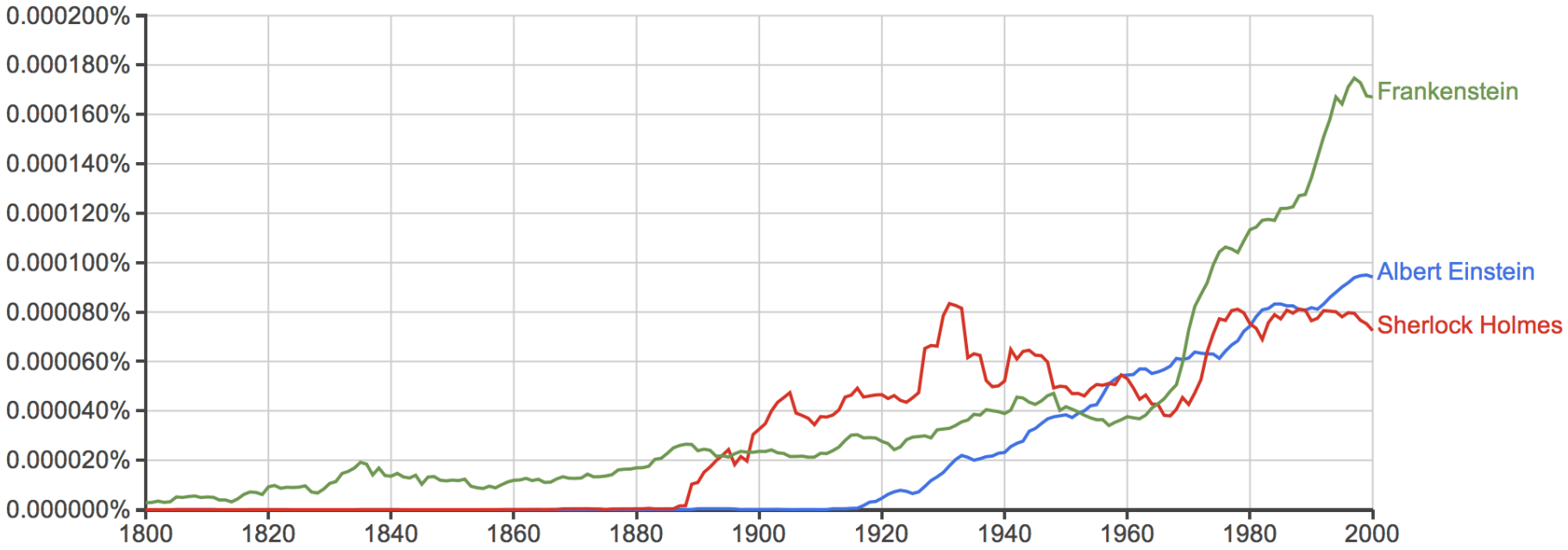
- In 2010, Google placed on on-line n-gram viewer that would display graphs of n-gram frequencies of one or more n-grams, based on a corpus defined from Google Books
  - https://books.google.com/ngrams
  - And see also the “About NGram Viewer” link

Google n-gram viewer

Google books Ngram Viewer

Graph these comma-separated phrases: Albert Einstein,Sherlock Holmes,Frankenstein case-insensitive

between 1800 and 2000 from the corpus English with smoothing of 3 . Search lots of books



Additional corpus measures

- Recall that so far, we have looked at two measures involving bigrams (and these definitions can be extended to n-grams):
  - Bigram frequency – percentage occurrence of the bigram in the corpus
    - Seen in the Google N-gram data
  - Bigram probability – conditional probability that the second word follows the first word in the corpus
    - Used to define Language Models
- Other measures can be defined about the occurrences of bigrams in a corpus
  - Mutual information, …
  - More of these can be found in the NLTK

Corpus Statistics: Mutual Information (MI)

- N-Gram probabilities predict the next word – Mutual Information computes probability of two words occurring in sequence
- Given a pair of words, compares probability that the two occur together as a joint event to the probability they occur individually & that their co-occurrences are simply the result of chance
  - The more strongly connected 2 items are, the higher will be their MI value

Mutual Information

- Based on work of Church & Hanks (1990), generalizing MI from information theory to apply to words in sequence
  - They used terminology *Association Ratio*
- P(x) and P(y) are estimated by the number of observations of x and y in a corpus and normalized by N, the size of the corpus
- P(x,y) is the number of times that x is followed by y in a window of w words
- Mutual Information score (also sometimes called PMI, Pointwise Mutual Information):

$$\text{PMI}(x,y) = \log_2 \left(P(x,y) / P(x)\,P(y) \right)$$

MI values based on 145 WSJ articles

x	freq (x)	y	freq (y)	freq (x,y)	MI
Gaza	3	Strip	3	3	14.42
joint	8	venture	4	4	13.00
Chapter	3	11	14	3	12.20
credit	15	card	11	7	11.44
average	22	yield	7	5	11.06
appeals	4	court	47	4	10.45
…..					
said	444	it	346	76	5.02

Uses of Mutual Information

- Used in similar NLP applications as Language Models
  - Idiomatic phrases for MT
  - Sense disambiguation (both statistical and symbolic approaches)
  - Error detection & correction in speech analysis and spell-checking
- Used for distributional semantics in “deep learning”
- Used in non-text applications such as comparing features in machine learning