ECE365: Introduction to NLP

Spring 2021

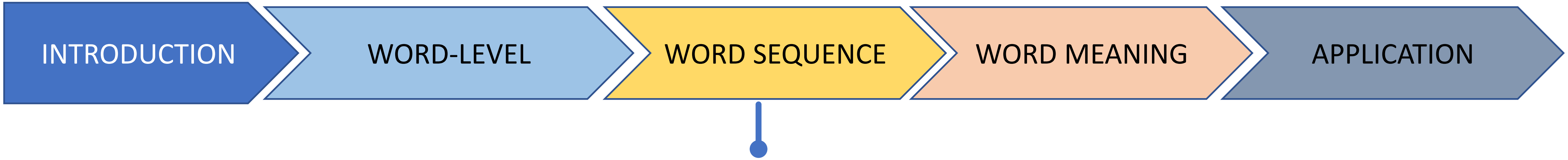
Lecture 4: Words in a sequence – Language modeling

[Reading J&M Chapter 3 (sections 3.1, 3.2)]

Logistics

- Quiz 1 today, material lectures 1, 2 & 3
- Quiz 1 solution will appear on course page later in the week
- Lab 3 will be up 04/20 due 4/28

Course Progress



WEEK 3

What is the nature of understanding we can get considering words as sequences?

ECE365_lecture4_postclass.p
Google

https://www.google.com/?gws_rd=ssl

We're updating our Terms of Service. Get to know our new Terms before they take effect on March 31, 2020. Review Got it

About Store

Gmail Images Sign in

Google

where is libr

where is library on mac
where is library
where is libra
where is library of congress
where is libreville
where is library on iphone
where is libra in the sky
where is library on ipad
where is library on ps4
where is library of congress located

Google Search I'm Feeling Lucky

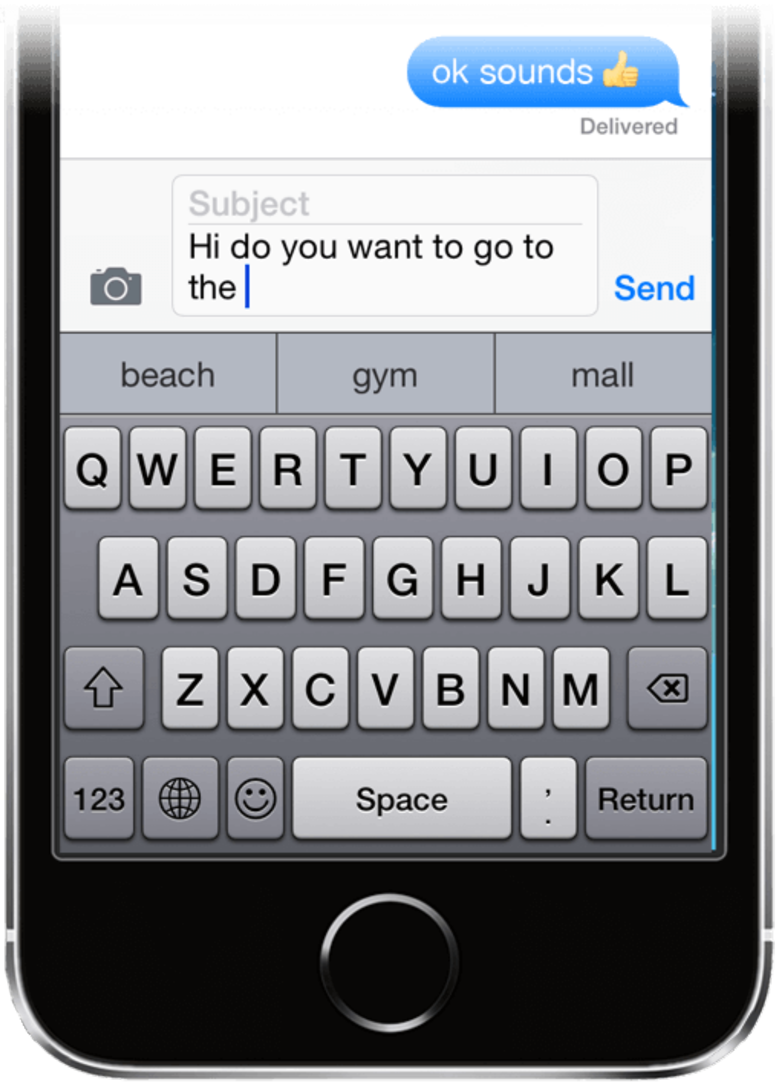
Report inappropriate predictions

Advertising Business How Search works

Privacy Terms Settings

Type here to search

12:31 PM
3/10/2020


ok sounds
Delivered
Subject
Hi do you want to go to the
Send
beach
gym
mall
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
Space
Return

What do we see?

The next word is being predicted

Consider the following instances

- Iryna went to the museum.
- Iryna the museum to went.
- Iryna went museum.

Consider System Outputs

I ate a cherry

Eye eight uh Jerry

Machine translation (Zh-> En)

他向记者介绍了发言的主要内容

- He briefed to reporters on the chief contents of the statement
- He briefed reporters on the chief contents of the statement
- He briefed to reporters on the main contents of the statement
- He briefed reporters on the main contents of the statement

Applications

- Deciding which word/phrase to choose

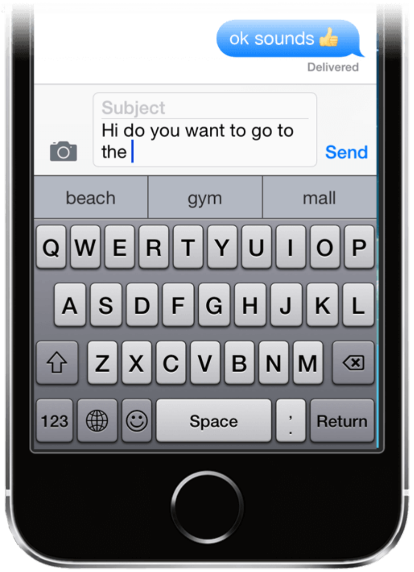
  Spelling checking/correction or Speech recognition:

  P(high school **principal**) > P(high school **principle**)

  Machine translation:

  P(How to make <u>strong</u> tea) > P(How to make <u>powerful</u> tea)

Problem



I ate a cherry

Eye eight uh Jerry

Sentence is not a bag of words. Use order of words permitted by language (grammar) to solve the problem.

Language model

What is a language model?

- A model that permit computing the probability of a sequence of words
  - How likely is a given word/phrase/sentence?

- P(X = *the*) be the probability that the random variable X takes the value “the”

- Let the joint probability of each word in a sequence (i.e., a sentence) having a particular value is

  $P(X_1 = w_1, X_2 = w_2, \ldots, X_n = w_n)$ *denoted by* $P(w_1, w_2, \ldots, w_n)$

Learning the language model

- *Chain rule*

$$P(w_1, w_2, \ldots, w_n) =$$

$$P(w_1\ w_2 \ldots\ w_n) = P(w_1)\ x\ P(w_2|w_1)\ x\ P(w_3|w_1\ w_2)\ x\ \ldots\ x\ P(w_n|w_1\ w_2\ \ldots w_{n-1})$$

P(the cow jumped over the moon) =

Learning the language model

- *Use a large corpus of the language*
  - Estimate the probability of word/phrase/sentence

  - By counting, i.e., using MLE

Estimating the Probabilities

$P(w_1\ w_2, \ldots\ w_n) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1\ w_2) \times \ldots \times P(w_n|w_1\ w_2 \ldots w_{n-1})$

Use a large corpus of English (e.g. Wikipedia) to get MLE

$$P(\mathrm{the}) = \frac{count(the)}{total\ words}$$

$$P(\mathrm{cow}|\mathrm{the}) = \frac{count(the\ cow)}{count(the)}$$

Markov Assumption

- Use only the recent past to predict the next word
- Definition: A sequence of n words is an n-gram
  - The cow jumped over the moon
  - unigram
  - bigram

Markov Assumption

- An n-gram language model considers only the most recent n-1 words
  - bigram model:

$P(w_n|w_1\, w_2 \ldots w_{n-1}) \approx$

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

K+1 gram model

Markov Assumption

- Use only the recent past to predict the next word

  - unigram LM

  $P(\texttt{moon}|\texttt{the cow jumped over the}) \approx$

  - Bigram LM

  $P(\texttt{moon}|\texttt{the cow jumped over the}) \approx$

N-gram models

Larger the N, more accurate and better the language model (but also higher costs)

1 trillion words from public web pages

Google AI Blog

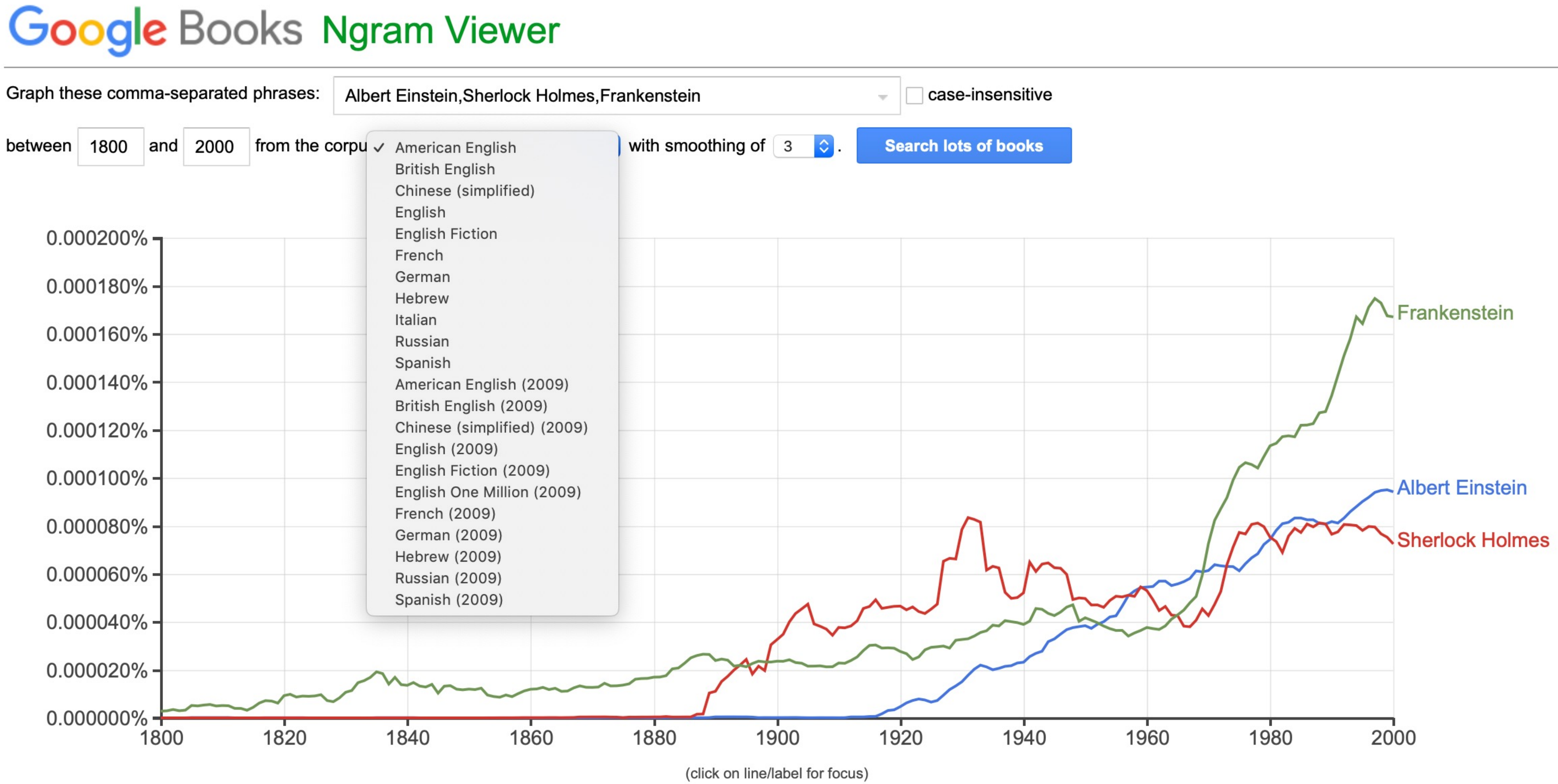
The latest news from Google AI

All Our N-gram are Belong to You

Thursday, August 3, 2006

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

N-grams


Google Books Ngram Viewer
Graph these comma-separated phrases:
Albert Einstein,Sherlock Holmes,Frankenstein
case-insensitive
between
1800
and
2000
from the corpu
American English
British English
Chinese (simplified)
English
English Fiction
French
German
Hebrew
Italian
Russian
Spanish
American English (2009)
British English (2009)
Chinese (simplified) (2009)
English (2009)
English Fiction (2009)
English One Million (2009)
French (2009)
German (2009)
Hebrew (2009)
Russian (2009)
Spanish (2009)
with smoothing of
3
Search lots of books
0.000200%
0.000180%
0.000160%
0.000140%
0.000120%
0.000100%
0.000080%
0.000060%
0.000040%
0.000020%
0.000000%
1800
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
Frankenstein
Albert Einstein
Sherlock Holmes
(click on line/label for focus)

Generalization of n-grams

- Not all n-grams will be observed in training data

- Long tail of infrequent words in finite-sized corpora
  - Zipf's law

- Test might have unseen words in training corpus
  - Training data: Google news
  - Test set: Shakespeare

  Since arm from arm that voice doth us affray,

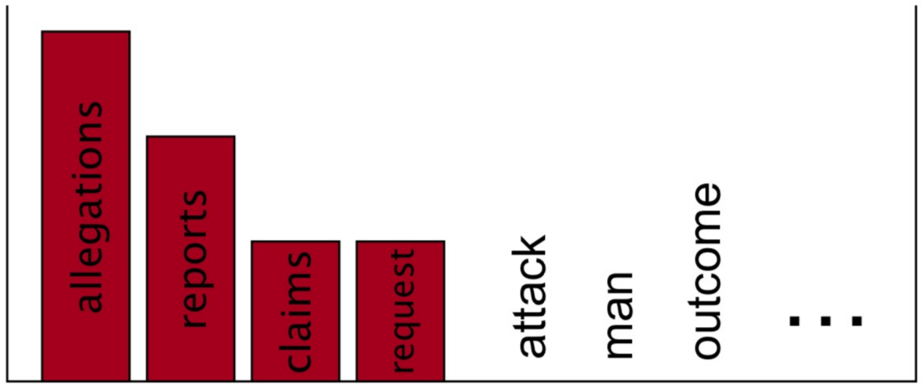
    - P (affray | voice doth us) = ?

Smoothing intuition

When we have sparse statistics:

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

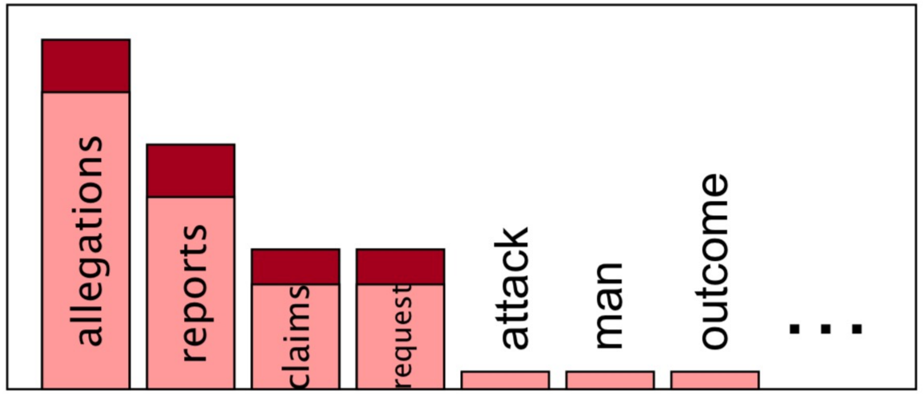
7 total



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

- Handle sparsity by making sure all probabilities are non-zero in our model
  - Additive: Add a small amount to all probabilities
  - Discounting: Redistribute probability mass from observed n- grams to unobserved ones
  - Back-off: Use lower order n-grams if higher ones are too sparse
  - Interpolation: Use a combination of different granularities of n-grams

Evaluating Language Models

- A good language model should assign higher probability to typical, grammatically correct sentences

Process

- on a suitable training corpus
  - Assumption: observed sentences ~ good sentences
- on *different, unseen* corpus
  Training on any part of test set not acceptable!

Evaluating Language Models

- Best evaluation for comparing LMs A and B
  - Put each model in a task
    - spelling corrector, speech recognizer, MT system
  - Run the task, get performance for A and for B
    - How many misspelled words corrected properly
    - How many words translated correctly
  - Compare A and B

  - Extrinsic evaluation, expensive

Evaluating language models

- Intrinsic evaluation
- Need a number that says how good my model is for a test set
  - Given by
  - If a model assigns higher   to a test set, it more   predicts the test set
- For a test sentence W given by $w_1w_2...w_N$

Practical Issues

- Computation using log
  - Probabilities can be very small

- <s> I am Sam </s>
- <s> Sam I am </s>
- <s> I do not like green eggs and ham </s>

Limitations

- In general this is an insufficient model of language
  - Because language has **long-distance dependencies**
  - "The computers, which I had just put into the machine room on the fifth floor, are crashing."
- But we can often get away with N-gram models
- State-of-the-art (for the curious): https://openai.com/blog/better-language-models/

Summary

- Text has an intrinsic structure and language model is a way to mathematically represent this structure
- Useful in a variety of tasks (decide the next word, or to judge the quality of a sentence)
- N-gram language models offer one way of doing this
- Smoothing as a way to prevent unseen words getting zero probability
- Evaluated using Perplexity (instrinsic), or in a specific task (extrinsic)