S&DS 365 / 565

Data Mining and Machine Learning

Language Models

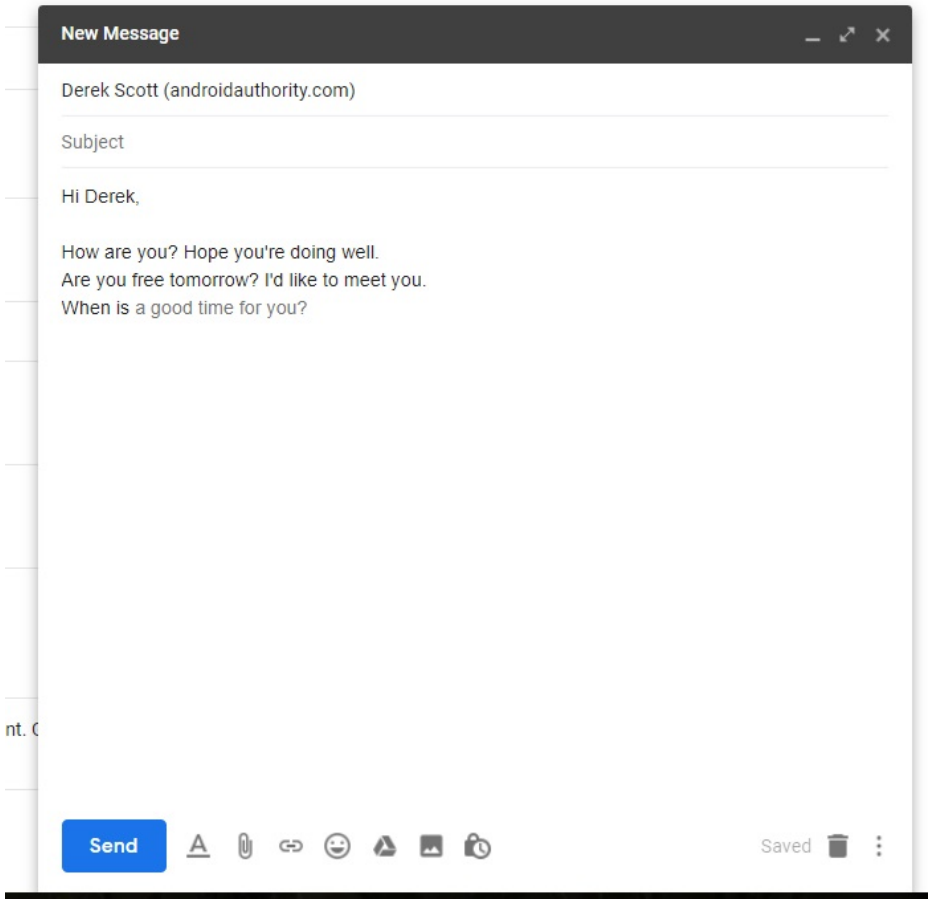
Yale

Outline

- Language models
- Class-based language models and clustering
- Some information theory

Hey Alexa

Smart Gmail


New Message
Derek Scott (androidauthority.com)
Subject
Hi Derek,
How are you? Hope you're doing well.
Are you free tomorrow? I'd like to meet you.
When is a good time for you?
Send
Saved

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

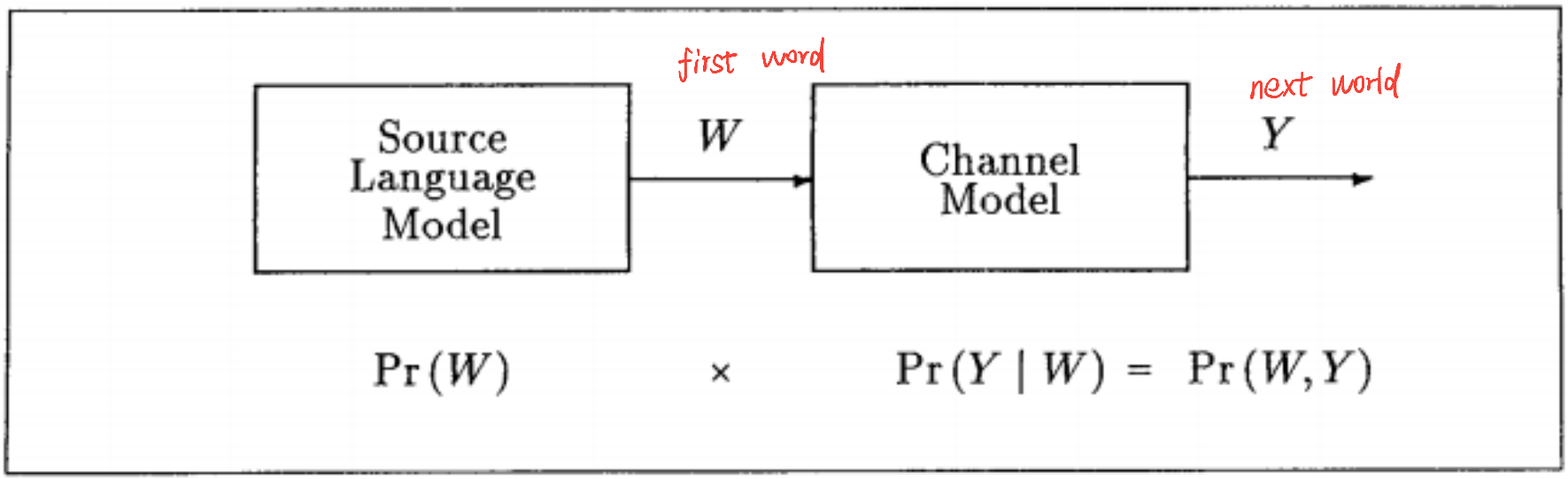
$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

low rank matrix

The source-channel framework



The source-channel framework

- Speech recognition
- Machine translation
- Texting
- Image captioning
- Mind reading from fMRI

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

- The number of *histories* grows as $|V|^{n-1}$. Number of free parameters in the last conditional probability is $(|V| - 1)|V|^{n-1}$. Why?
- What are some ways of reducing the number of parameters?

Grouping histories

- We need to group histories.
- Let $\pi_n : V^n \to \mathcal{C}$ be a mapping from word sequences of length n to some finite set.
- $\mathcal{C}$ a context
- Our model becomes

$$p(w_{n+1} \mid w_1, \ldots, w_n) = p(w_{n+1} \mid \pi_n(w_1, \ldots, w_n))$$

- Number of parameters: $O(|V| \cdot |\mathcal{C}|)$
- What are some example groupings?

Grouping histories

- Unigrams: $\pi(w_1, \ldots, w_n) = \emptyset$.

Grouping histories

- Unigrams: $\pi(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $\pi(w_1, \ldots, w_n) = w_n$.

Grouping histories

- Unigrams: $\pi(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $\pi(w_1, \ldots, w_n) = w_n$.
- Trigrams: $\pi(w_1, \ldots, w_n) = (w_{n-1}, w_n)$.

Grouping histories

- Unigrams: $\pi(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $\pi(w_1, \ldots, w_n) = w_n$.
- Trigrams: $\pi(w_1, \ldots, w_n) = (w_{n-1}, w_n)$.
- Number of parameters grows as $O(|V|)$, $O(|V|^2)$, and $O(|V|^3)$, respectively.

Grouping histories

- Unigrams: $\pi(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $\pi(w_1, \ldots, w_n) = w_n$.
- Trigrams: $\pi(w_1, \ldots, w_n) = (w_{n-1}, w_n)$.
- Number of parameters grows as $O(|V|)$, $O(|V|^2)$, and $O(|V|^3)$, respectively.
- Number of parameters in topics for LDA:

$$O(K \cdot V)$$

words are independent with each other
I don't pay attention to any history
only listen to the last word

listen to the last 2 words

number of topics

size of vocabulary

Estimating parameters

MLE Model

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- What are some problems with this model?

in training set w_1, w_2, w_3 may not appear together: $\text{count}(w_1, w_2, w_3) = 0$

Estimating parameters

Often used

- Bayesian approach: Dirichlet prior

$$\widehat{p}(w_3 \mid w_1, w_2) \propto \text{count}(w_1, w_2, w_3) + \eta$$

↑

add weight (-3 ~ 10)

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- Some kind of "shrinkage" or smoothing needs to be done.
- How else can the model be strengthened?

Interpolation

Linear interpolation:

$$p(w_3 \mid w_1, w_2) = \lambda_3 \widehat{p}(w_3 \mid w_1, w_2) + \lambda_2 \widehat{p}(w_3 \mid w_2) + \lambda_1 \widehat{p}(w_3)$$

Interpolation

Linear interpolation:

λ_i: weight of model i

$$p(w_3 \mid w_1, w_2) = \lambda_3 \widehat{p}(w_3 \mid w_1, w_2) + \lambda_2 \widehat{p}(w_3 \mid w_2) + \lambda_1 \widehat{p}(w_3)$$

trigram model | bigram model | unigram model

This is a type of mixture model.

(But the latent structure isn't interesting.)

Class-based bigram model

- Model takes form

$$p(w_2 \mid w_1) = p(\text{class}(w_2) \mid \text{class}(w_1))\, p(w_2 \mid \text{class}(w_2))$$
$$= p(c_2 \mid c_1)\, p(w_2 \mid c_2)$$

- Use bottom-up agglomerative clustering to group the words.
- Bigram model gives highest likelihood (no grouping)
- Each step: merge the pair of classes that gives the smallest reduction in likelihood of the data.

group words based on which one hurt perplexity is least

Brown et al., "Class-based *n*-gram models of natural language" (1992)

Perplexity 困惑度

$$\text{Perplexity}(\theta) = \left(\underbrace{\prod_{i=1}^{N} p_\theta(w_i \mid w_{1:i-1})}_{\text{likelihood}} \right)^{-1/N}$$

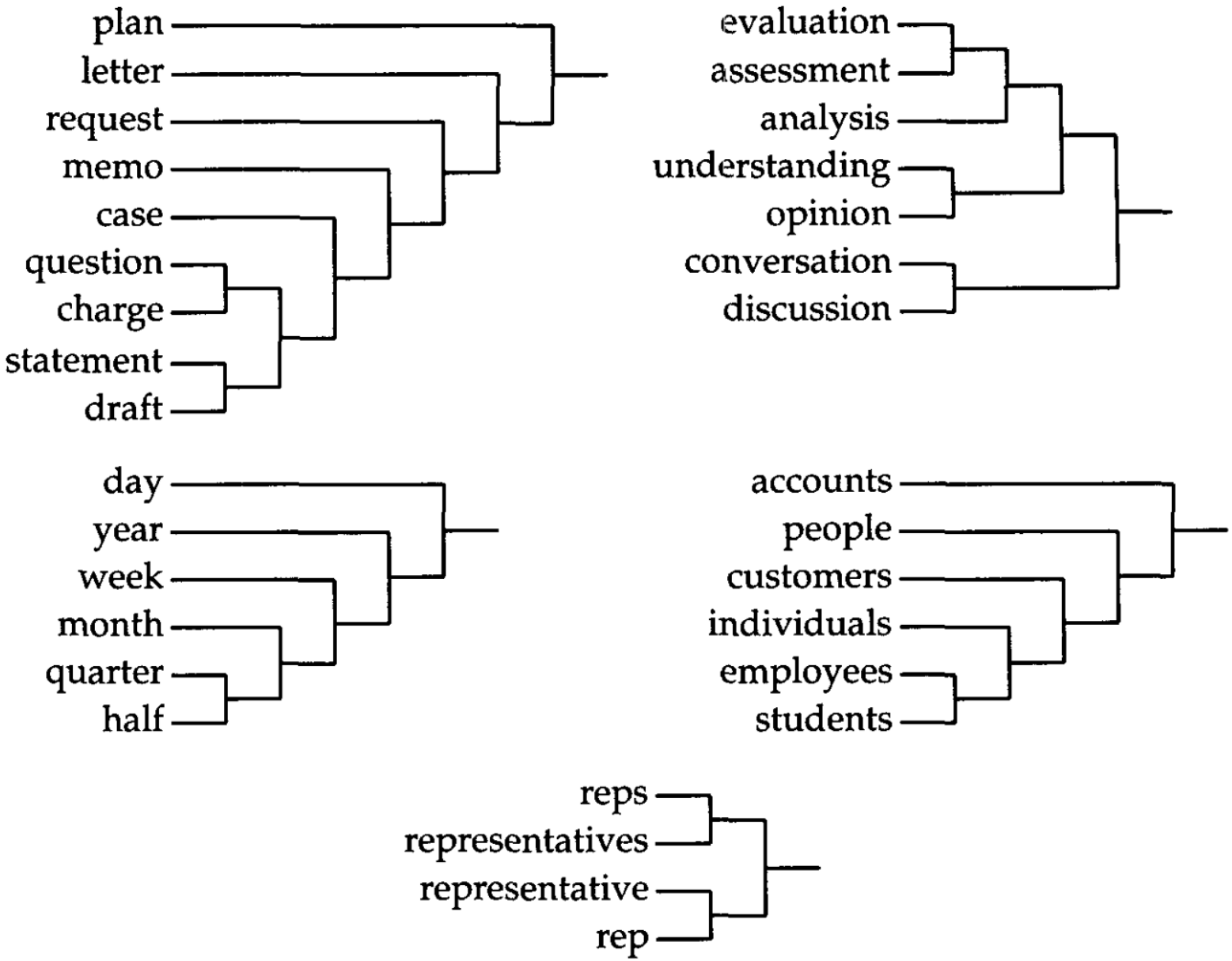
Just likelihood raised to $-1/N$. Smaller better.

log perplexity = $\frac{1}{N}$ negative log likelihood

If perplexity is 100, then the model predicts, on average, as if there were 100 equally likely words to follow.

$\left(\prod_{i=1}^{100} \frac{1}{100}\right)^{-\frac{1}{100}} = \left(\frac{1}{100}\right)^{100 \cdot (-\frac{1}{100})} = 100$

Sample merges



Sample clusters

Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays

June March July April January December October November September August

people guys folks fellows CEOs chaps doubters commies unfortunates blokes

down backwards ashore sideways southward northward overboard aloft downwards adrift

water gas coal liquid acid sand carbon steam shale iron

great big vast sudden mere sheer gigantic lifelong scant colossal

man woman boy girl lawyer doctor guy farmer teacher citizen

American Indian European Japanese German African Catholic Israeli Italian Arab

pressure temperature permeability density porosity stress velocity viscosity gravity tension

mother wife father son husband brother daughter sister boss uncle

machine device controller processor CPU printer spindle subsystem compiler plotter

John George James Bob Robert Paul William Jim David Mike

anyone someone anybody somebody

feet miles pounds degrees inches barrels tons acres meters bytes

director chief professor commissioner commander treasurer founder superintendent dean custodian

liberal conservative parliamentary royal progressive Tory provisional separatist federalist PQ

had hadn't hath would've could've should've must've might've

asking telling wondering instructing informing kidding reminding bothering thanking deposing

that tha theat

head body hands eyes voice arm seat eye hair mouth

Group globally, compute locally

focus on context

- Clusters contain syntactic and semantic elements
- Surprising, since use local statistics only
- “A word is known by the company it keeps”

Class-based bigram model: Perplexity

- Reduces size of model with relatively small increase in perplexity, $244 \mapsto 271$.

Pointwise mutual information (PMI)

Average mutual information

$$I(W_1, W_2) = \sum_{w_1, w_2} p(w_1, w_2) \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}$$

word 1 word 2

$$= \sum_{w_1} p(w_2 \mid w_1) \log \frac{p(w_2 \mid w_1)}{p(w_2)}$$

$$= H(W_2) - H(W_2 \mid W_1)$$

$H(W_2) = \sum_{w_2} p(w_2) \log\left(\frac{1}{p(w_2)}\right)$

Measure uncertainty of word 2

entropy

Related statistic is "pointwise mutual information" (PMI)

Measure how connected 2 words are (micky mouse)

$\begin{cases} >0 & \text{very connected} \\ <0 & \text{very not} \sim \end{cases}$

$$\log\left(\frac{p_{\text{near}}(w_1, w_2)}{p(w_1)p(w_2)}\right)$$

joint probability distribution of near model

- How likely are specific words/clusters to co-occur together within some window, compared to if they were independent?

Example clusters from PMI

we our us ourselves ours
question questions asking answer answers answering
performance performed perform performs performing
tie jacket suit
write writes writing written wrote pen
morning noon evening night nights midnight bed
attorney counsel trial court judge
problems problem solution solve analyzed solved solving
letter addressed enclosed letters correspondence
large size small larger smaller
operations operations operating operate operated
school classroom teaching grade math
street block avenue corner blocks
table tables dining chairs plate
published publication author publish writer titled
wall ceiling walls enclosure roof
sell buy selling buying sold

Shortcomings of word clusters

- No centroid in $\mathbb{R}^d$, or Euclidean structure
- Can't use vector space operations
- "One hot" representation wasteful
- These are addressed with *distributed representations* (next)

total k words

one hot coding for words

word j $\in \mathbb{R}^k$ = $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ → jth

sharing info between features

eg: topic modeling : info shared is topic

if words have similar topic structure ⇒ 2 words are similar

Summary: Language models

- A language model is a conditional probability model for predicting/generating the next word (or character) of text
- Probabilities need to be “smoothed” to avoid zeros
- Grouping words into classes and estimating the class-based model gives meaningful clusters
- Surprising amount of information captured by purely local cooccurrence counts
- Looking at cooccurrence over larger windows gives more “semantic” information