Lecture 6

tf-idf: combine two factors

tf: term frequency. frequency count (usually log-transformed):

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10}\mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Idf: inverse document frequency: tf-

$$\mathrm{idf}_i = \log\left(\frac{N}{\mathrm{df}_i}\right)$$

Total # of docs in collection

# of docs that have word i

Words like "the" or "good" have very low idf

tf-idf value for word t in document d:

$$w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

Summary: tf-idf

Compare two words using tf-idf cosine to see if they are similar

Compare two documents

- Take the centroid of vectors of all the words in the document
- Centroid document vector is:

$$d = \frac{w_1 + w_2 + \ldots + w_k}{k}$$

Pointwise Mutual Information

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Positive Pointwise Mutual Information

- PMI ranges from $-\infty$ to $+\infty$
- But the negative values are problematic
  - Things are co-occurring **less than** we expect by chance
  - Unreliable without enormous corpora
    - Imagine w1 and w2 whose probability is each 10^{-6}
    - Hard to be sure p(w1,w2) is significantly different than 10^{-12}
  - Plus it's not clear people are good at "unrelatedness"
- So we just replace negative PMI values by 0
- Positive PMI (PPMI) between word1 and word2:

$$\mathrm{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

Example:

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w,context) / **p(w)**

	computer	data	pinch	result	sugar	p(w)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

p(w,context) / **p(w)**

	computer	data	pinch	result	sugar	p(w)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

pmi(information,data) = $\log_2$ (.32 / (.37*.58)) = .58

(.57 using full precision)

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

1. Give higher prob

Weighting PMI: Giving rare context words slightly higher probability

Raise the context probabilities to $\alpha = 0.75$:

$$\mathrm{PPMI}_\alpha(w,c) = \max\left(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0\right)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

This helps because $P_\alpha(c) > P(c)$ for rare c

Consider two events, P(a) = .99 and P(b)=.01

$$P_\alpha(a) = \frac{.99^{.75}}{.99^{.75} + .01^{.75}} = .97 \quad P_\alpha(b) = \frac{.01^{.75}}{.01^{.75} + .01^{.75}} = .03$$

Lecture 7

Language modeling

- x is a "history" $w_1, w_2, \ldots w_{i-1}$
  - *Third, the notion "grammatical in English" cannot be identified in any way with the notion "high order of statistical approximation to English". It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical*
- Each possible *y* gets a score:

  v·f(x,model)=5.6 v·f(x,the)=-3.2
  v·f(x,is)=1.5 v·f(x,of)=1.3
  v·f(x,models)=4.5 …

$$p(\text{model} \mid x; v) = \frac{e^{5.6}}{e^{5.6} + e^{1.5} + e^{4.5} + e^{-3.2} + \ldots}$$

29

Some questions:

(1) Fig. 7.6 in Chapter 7 shows a simple neural network for the XOR problem with two variables. Show a similar neural network with a different set of parameters that also correctly computes XOR. Show that your neural network correctly classifies the 4 input pairs by working through the computations.
(2) Use the softmax function to convert the following set of scores to probabilities: z=[0.6 1.1 -1.5 1.2 3.2 -1.1]

(3) Computation graph. Consider computing the function L(a,b, c) = a(2c+3b). Assume the inputs a=2, b=1, c=4. Draw the computation graph and show the result of the forward pass for computing the value of L.
(4) Now perform the backward pass on the graph and show the computations for the backward pass on the graph.

2. $f_i(\vec{a}) = \frac{e^{a_i}}{\sum_k e^{a_k}}$ $f' = f(1-f)$

$e^a = [1.82, 3.00, 0.22, 3.32, 24.53, 0.33]$

$\sum_k e^{a_k} = 33.22$

$f_1(\vec{a}) = \frac{1.82}{33.22} = 0.05$

$f_2(\vec{a}) = \frac{3}{33.22} = 0.09$

$f_3(\vec{a}) = \frac{0.22}{33.22} = 0.007$

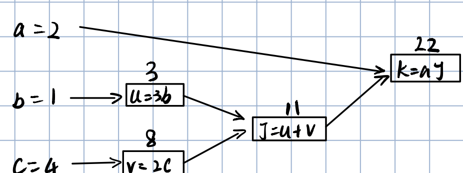
$f_4(\vec{a}) = \frac{3.32}{33.22} = 0.10$

$f_5(\vec{a}) = \frac{24.53}{33.22} = 0.74$

$f_6(\vec{a}) = \frac{0.33}{33.22} = 0.01$

3. $L(a,b,c) = a(2c+3b)$, forward pass

$u = 3b$
$v = 2c$
$J = u + v$
$k = aJ$



4. backward.

$k = aJ$

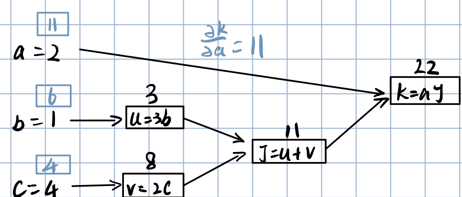
$\frac{\partial k}{\partial a} = J = 11$

$\frac{\partial k}{\partial b} = \frac{\partial k}{\partial J} \cdot \frac{\partial J}{\partial u} \cdot \frac{\partial u}{\partial b}$

$= 2 \cdot 1 \cdot 3$

$= 6$

$\frac{\partial k}{\partial c} = \frac{\partial k}{\partial J} \cdot \frac{\partial J}{\partial v} \cdot \frac{\partial v}{\partial c}$



$= 2 \cdot 1 \cdot 2$

$= 4$