Logistics

- Reminder: you have 5 free late days. Once you've used them all up, you must turn in all subsequent assignments on time.
- You can see your late days by logging into Gradescope and checking your assignment submission times. Late ones are marked "LATE".
- If you cannot find your info on Piazza email Jie Gao.
- On Thursday: Guest lecture by Prof. Mitch Marcus!

Introduction to Markov Models

Estimating the probability of phrases of words, sentences, etc.…

But first:
A few preliminaries on text preprocessing

What counts as a word? A tricky question….

How to tokenize N'T?

- It makes sense to tokenize **didn't** as **did n't**, **hasn't** as **has n't**.
- BUT **can't** becomes **ca n't**.

How to tokenize NEEDLE-LIKE, SEVEN-DAY, MID-OCTOBER, CRAY-3?

- It seems sensible to leave hyphenated items as single tokens.
- **But:**
  - New **York-based**
  - the **New York-New Haven** Railroad
  - an **ad hoc** solution

How to find Sentences??

The Obvious Heuristic For Sentence Boundaries:

```
/[.?!]['’’]*/
```

But: I saw **Mr.** Jones visiting **St.** Peter's basilica.

Patch: Delete the break after Mr. | Mrs. | Dr. | St. | Prof | ... **But:**

- He left at 3 **a.m.** in the morning.
- He left at 3 **a.m.**
- In LISP, **2.0** and **2.** stand for the same number.

Patch: Don't break if the next word isn't capitalized.

But: We saw Peter on Jones **St.** Peter's brother was with him.

Q1: How to estimate the probability of a given sentence *W*?

- **A crucial step in speech recognition (and lots of other applications)**
- **First guess: bag of words :** $\hat{P}(W) = \prod_{w \in W} P(w)$

Given word lattice:

form	subsidy	for
farm	subsidies	far

Unigram **counts (in 1.7 * 10^6 words of AP text):**

form 183	*subsidy 15*	***for 18185***
farm 74	***subsidies 55***	*far 570*

Most likely word string given $\hat{P}(W)$ ***isn't quite right…***

Penn University of Pennsylvania

Predicting a word sequence II

- **Next guess: products of *bigrams***
  - For $W = w_1 w_2 w_3 \ldots w_n,$ $\hat{P}(W) = \prod_{i=1}^{n-1} P(w_i w_{i+1})$

Given word lattice:

form	*subsidy*	*for*
farm	*subsidies*	*far*

Bigram counts (in 1.7 * 10^6 words of AP text):

form subsidy 0	subsidy for 2
form subsidies 0	subsidy far 0
farm subsidy 0	**subsidies for 6**
farm subsidies 4	subsidies far 0

Much Better (if not quite right) …

(Q: the counts are tiny! Why?)

Language models

- A **language model** assigns a **probability** to a **sequence of words**
- Applications include:
  - Autocomplete for texting
  - Spelling correction
  - Speech recognition
  - Machine translation
  - Other natural language generation tasks

Language models

- Goal: compute the probability of a sentence or sequence of words $\mathbf{W = w_1, w_2, w_3, w_4, w_5, \ldots, w_N.}$

  $P(W) = p(w_1, w_2, w_3, w_4, w_5, \ldots, w_N)$

  p(the underdog Philadelphia Eagles won the Superbowl

- We also want to calculate the probability of an upcoming word:
  $p(w_5 \mid w_1, w_2, w_3, w_4)$

- **What kinds of probabilities are these?**

Manipulating probabilities

- Relationship between joint and conditional probabilities
  $p(B \mid A) = p(A, B) / p(A)$
  $p(A, B) = p(B \mid A) * p(A)$

- We can apply the chain rule:
  $p(A,B,C,D) = p(A) * p(B|A) * p(C|A,B) * p(D|A,B,C)$

- Or generally:

$$p(w_1, w_2, w_3, \ldots, w_N) = p(w_1) * p(w_2|w_1) * p(w_3|w_1, w_2) * \ldots p(w_N|w_1, \ldots, w_{N-1})$$

$$= \prod_i p(w_i | w_1 w_2 w_3 \ldots w_{i-1})$$

Estimating probabilities

- **How do we estimate these probabilities for a sentence?**
  p(the underdog team won)
- **Maximum likelihood estimation (MLE):**
  - Count a divide
  - p(the) = count(the) / length of whole training data
  - p(underdog | the) = count(the underdog) / count(the)
  - p(team | the underdog) = count(the underdog team) / count(the underdog)
  - p(won | the underdog team) = count(the underdog team won) / count(the underdog team)

The Web *is* HUGE!

"the underdog team"

All News Images Videos Maps More Settings Tools

About 30,600 results (0.33 seconds)

10 Simple points on how to coach an Underdog Team | Layups.com
www.layups.com/10-simple-points-on-how-to-coach-an-underdog-team/
This should make them aware of what their opponents are capable of doing, as well as their advantages against them as **the underdog team**. Their awareness ...

ogy ...

am, my belief is

"the underdog team won"

All News Videos Images Maps More Settings Tools

About 1,410 results (0.43 seconds)

Why We Root for the Underdog - Undark
https://undark.org/article/why-we-root-for-the-underdog/
Feb 29, 2016 - Then, the same group was asked to imagine their feelings if **the underdog team won** the first three games of a series. Half of the participants ...

College Bowl team reigns at Penn - University of Chicago Chronicle
chronicle.uchicago.edu/980205/bowl.shtml
Feb 5, 1998 - Though **the underdog team won** the Super Bowl on Jan. 25, a favorite claimed victory that day in a bowl of another sort -- the 7th annual Penn ...

or

p winning, this

$$\boxed{1410/30600=0.046}$$

MLE is Problematic

- Sentences may never appear
- Need a large data set
- Many ways of combining words into sentences
- Never going to have enough data to estimate the probability of a whole sentence

How can we estimate *P(W) correctly*?

Let's do this right….

- **Let $W=w_1w_2w_3\ldots w_n$. Then, by the chain rule,**

$$P(W) = P(w_1) * P(w_2 \mid w_1) * P(w_3 \mid w_1 w_2) * \ldots * P(w_n \mid w_1 \ldots w_{n-1})$$

- **We can estimate $P(w_2|w_1)$ by the *Maximum Likelihood Estimator***

$$\frac{Count(w_1 w_2)}{Count(w_1)}$$

and $P(w_3|w_1w_2)$ by

$$\frac{Count(w_1 w_2 w_3)}{Count(w_1 w_2)}$$

and so on…

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

Again, we can estimate $P(w_n \mid w_1 w_2 \ldots w_{n-1})$ with the MLE

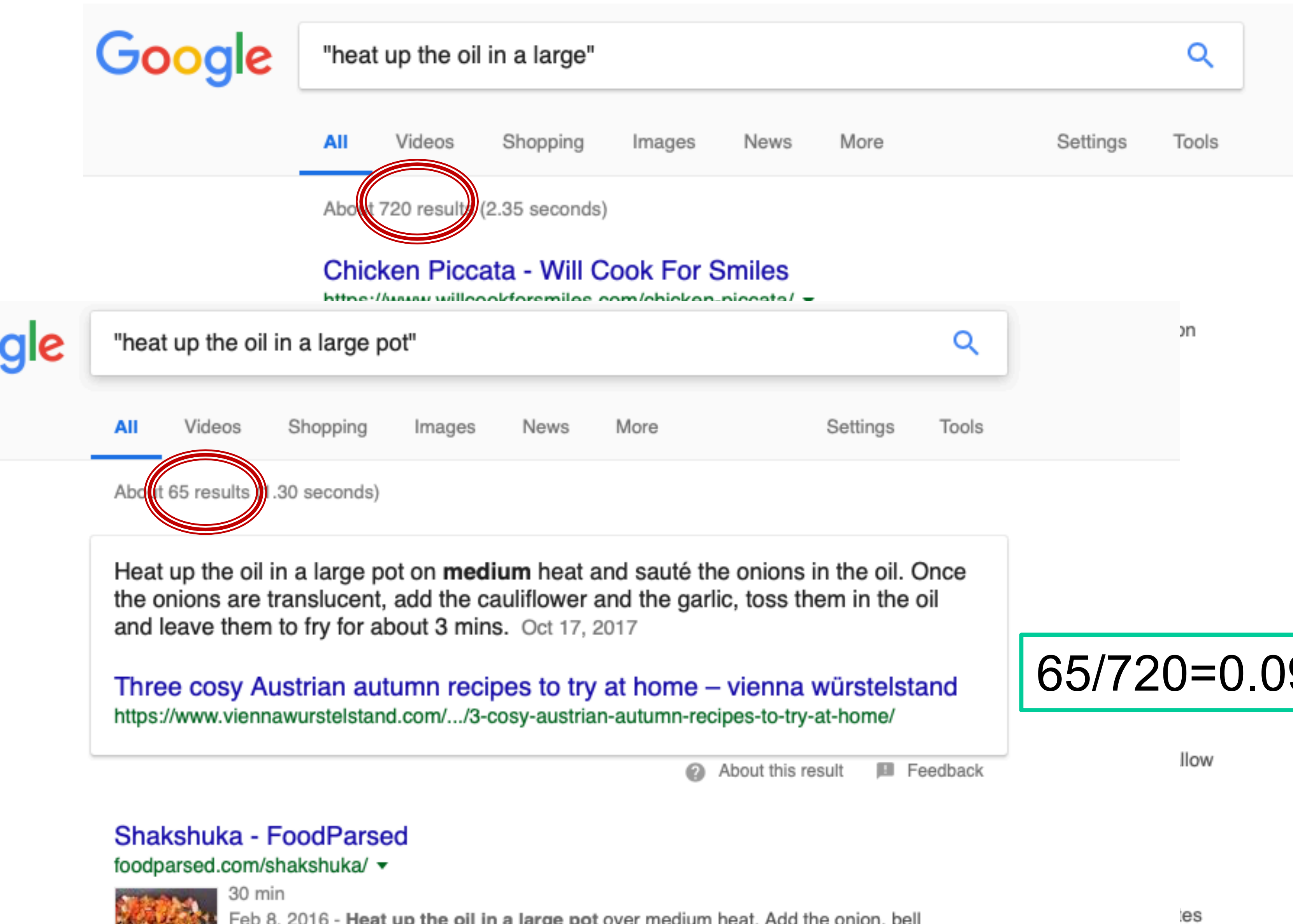
$$\frac{Count(w_1 w_2 \ldots w_n)}{Count(w_1\, w_2 \ldots w_{n-1})}$$

So to decide *pat* vs. *pot in Heat up the oil in a large p?t,*
compute for *pot*

$$\frac{Count(\text{"Heat up the oil in a large pot"})}{Count(\text{"Heat up the oil in a large"})}$$

UNLESS OUR CORPUS IS REALLY HUGE
BOTH COUNTS WILL BE 0, yielding 0/0

The Web *is* HUGE!



65/720=0.09

But what if we have "only" 100 million words for our estimates?

A BOTEC Estimate of What We Can Estimate

What parameters can we estimate with 100 million words of training data??

Assuming (for now) uniform distribution over only 5000 words

Event	**Count**	**Estimate Quality?**
words	5000	Excellent
word bigrams	25 million	OK
word trigrams	12.5 billion	Terrible!

So even with 10^8 words of data, for even trigrams we encounter the *sparse data problem*..

Review: How can we estimate *P(W) correctly*?

- **Problem: Naïve Bayes model for bigrams violates independence assumptions.**

Let's do this right….

- **Let $W=w_1w_2w_3\ldots w_n$. Then, by the chain rule,**

$$P(W) = P(w_1) * P(w_2 \mid w_1) * P(w_3 \mid w_1 w_2) * \ldots * P(w_n \mid w_1 \ldots w_{n-1})$$

- **We can estimate $P(w_2|w_1)$ by the *Maximum Likelihood Estimator***

$$\frac{Count(w_1 w_2)}{Count(w_1)}$$ ✓

and $P(w_3|w_1w_2)$ by

$$\frac{Count(w_1 w_2 w_3)}{Count(w_1 w_2)}$$ ✗

and so on… ✗

The Markov Assumption: Only the Immediate Past Matters

The (First Order) Markov Assumption:

$$\mathrm{P(w_i|w_1 \ldots w_{i-1}) = P(w_i|w_{i-1})}$$

Under this assumption, instead of

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

we estimate the probability of a string $\boldsymbol{W}$ by

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

The Markov Assumption: Estimation

We estimate the probability of each w_i given previous context by

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) = P(w_i \mid w_{i-1})$$

which can be estimated by

$$\frac{Count(w_{i-1} w_i)}{Count(w_{i-1})}$$

So we're back to counting only unigrams and bigrams!!

AND we have a *correct practical* estimation method for $P(W)$ *given the Markov assumption*!

Markov Models

A bigram model can be viewed as a **Markov model**, a probabilistic FSA with

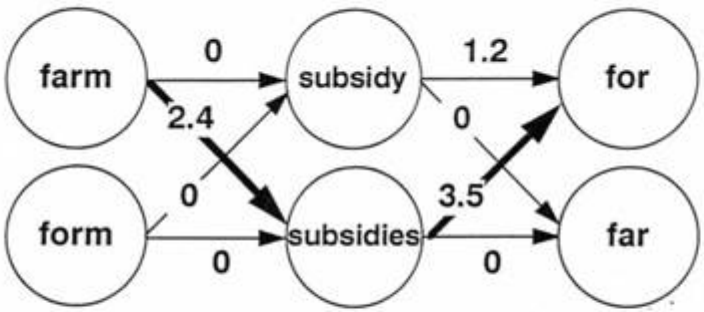
- **S, a set of states,** one for each word w_i in the vocabulary.
- **A, a transition matrix** where $a(i,j)$ is the probability of going from state w_i to state w_j.

  The probability $a(i,j)$ can be estimated by

$$a(i,j) = \frac{Count(w_i w_j)}{Count(w_i)}$$

.

- **Π, a vector of initial state probabilities,** where $\pi(i)$ is the probability of the first word being w_i.



Transition probabilities x 10^{-6}

Review (and crucial for upcoming homework): Cumulative distribution Functions (CDFs)

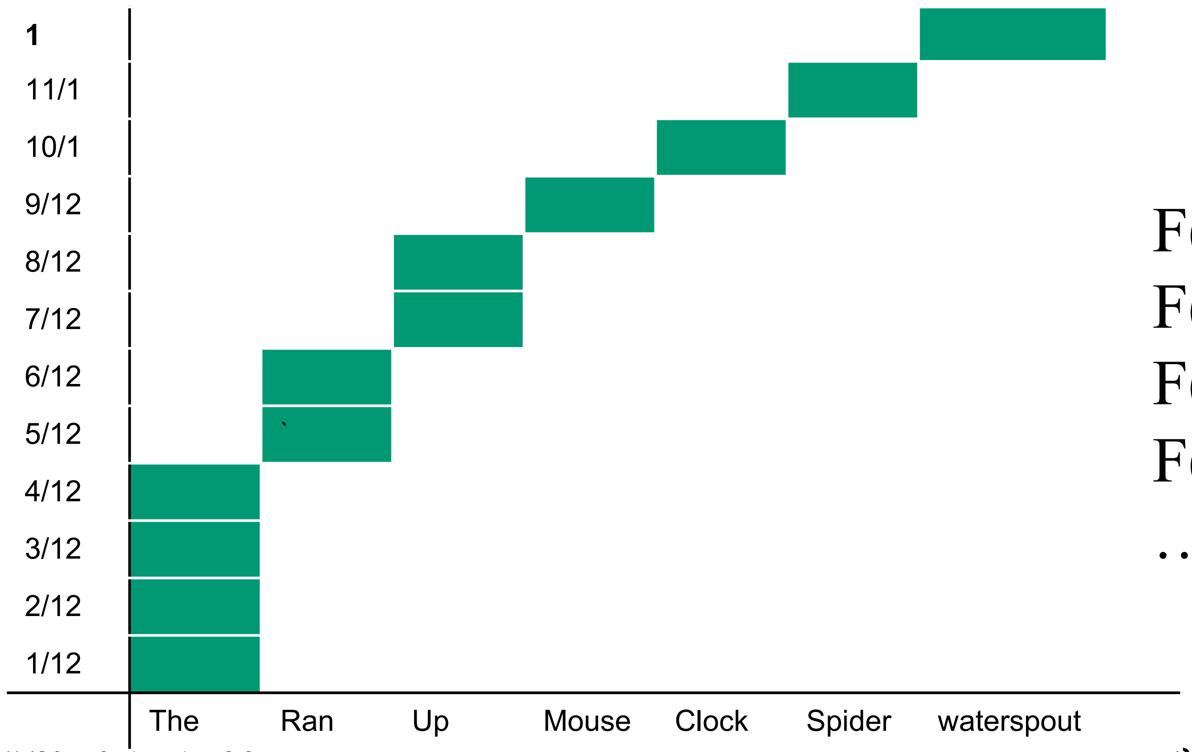
- **The CDF of a random variable** X **is denoted by** $F_X(x)$ **and is defined by** $F_X(x)=Pr(X\leq x)$
  - F is monotonic nondecreasing: $\forall x \leq y, F(x) \leq F(y)$

- **If X is a discrete random variable that attains values x_1, x_2, …, x_n with probabilities $p(x_1)$, $p(x_2)$…, then**

$$\boldsymbol{F_x(x_i) = \sum_{j\leq i} p(x_j)}$$

CDF for a very small English corpus

- **Corpus: "the mouse ran up the clock. The spider ran up the waterspout."**
- ***P(the)*=4/12, *P(ran)*=P(up)=2/12**
- ***P(mouse)=P(clock)=P(spider)=P(waterspout)*=1/12**
- ***Arbitrarily* fix an order: *w1=the*, *w2=ran*, *w3=up*, *w4=mouse*, …**



F(*the*)=4/12
F(*ran*)=6/12
F(*up*)=8/12
F(*mouse*)=9/12
…

Visualizing an n-gram based language model: the Shannon/Miller/Selfridge method

- **To generate a sequence of n words given *unigram* estimates:**
  - Fix some ordering of the vocabulary $v_1\, v_2\, v_3 \ldots v_k$.
  - For each word ***position i***, $1 \le i \le n$
    - —Choose a random value r_i between 0 and 1
    - —***Choose*** w_i = the first v_j such that $F_V(v) \ge r_i$

      i.e the first v_j such that $\sum_{m=1}^{j} P(v_m) \ge r_i$

Visualizing an n-gram based language model: the Shannon/Miller/Selfridge method

- **To generate a sequence of n words given *a 1^{st} order* Markov model (i.e. conditioned on one previous word):**
  - Fix some ordering of the vocabulary $v_1\, v_2\, v_3 \ldots v_k$.
  - Use unigram method to generate an initial word w_1
  - For each remaining position i, $2 \le i \le n$
    —Choose a random value r_i between 0 and 1
    — Choose w_i = the first v_j such that $\sum_{m=1}^{j} P(v_m \mid w_{i-1}) \ge r_i$

The Shannon/Miller/Selfridge method trained on Shakespeare

Unigram

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
Every enter now severally so, let
Hill he late speaks; or! a more to leg less first you enter
Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Bigram

What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Trigram

Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Quadrigram

King Henry.What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
Will you not tell me who I am?
It cannot be but so.
Indeed the short and the long. Marry, 'tis a noble Lepidus.

(This and next two slides from Jurafsky)

Wall Street Journal just isn't Shakespeare

Unigram

Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

Bigram

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Trigram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Shakespeare as corpus

- **N=884,647 tokens, V=29,066**
- **Shakespeare produced 300,000 bigram types out of V^2= 844 million possible bigrams.**
  - So 99.96% of the possible bigrams were never seen (have zero entries in the table)
- **4-grams are worse: What's coming out looks like Shakespeare because it *is* Shakespeare**

The Sparse Data Problem Again

Under the Markov Assumption,

$$\boldsymbol{P(W) = P(w_1) * P(w_2|w_1) * \ldots * P(w_{i+1}|w_i) * \ldots * P(w_n|w_{n-1})}$$

But what if we've never before seen $\boldsymbol{w_i w_{i+1}}$ in string $\boldsymbol{W}$?

Then our estimate of $\boldsymbol{P(w_{i+1}|w_i)}$ is

$$\boldsymbol{\frac{Count(w_i w_{i+1})}{Count(w_i)} = \frac{0}{Count(w_i)} = 0}$$

So our estimate of $\boldsymbol{P(W)}$=0!

- **How likely is a 0 count? Much more likely than I let on!!!**
- **So we use a technique called *smoothing***

Smoothing

- How do we avoid zero probabilities?
- Add one!

$$P_{add1}(w_i|w_{i-1}) = \text{count}(w_{i-1}, w_i) + 1 \; / \\ \text{count}(w_{i-1}) + V$$

Where v is the size of the vocabulary

English word frequencies well described by *Zipf's Law*

- **Zipf (1949) characterized the relation between word frequency and rank as:**

$$f \cdot r = C \quad (\text{for constant } C)$$

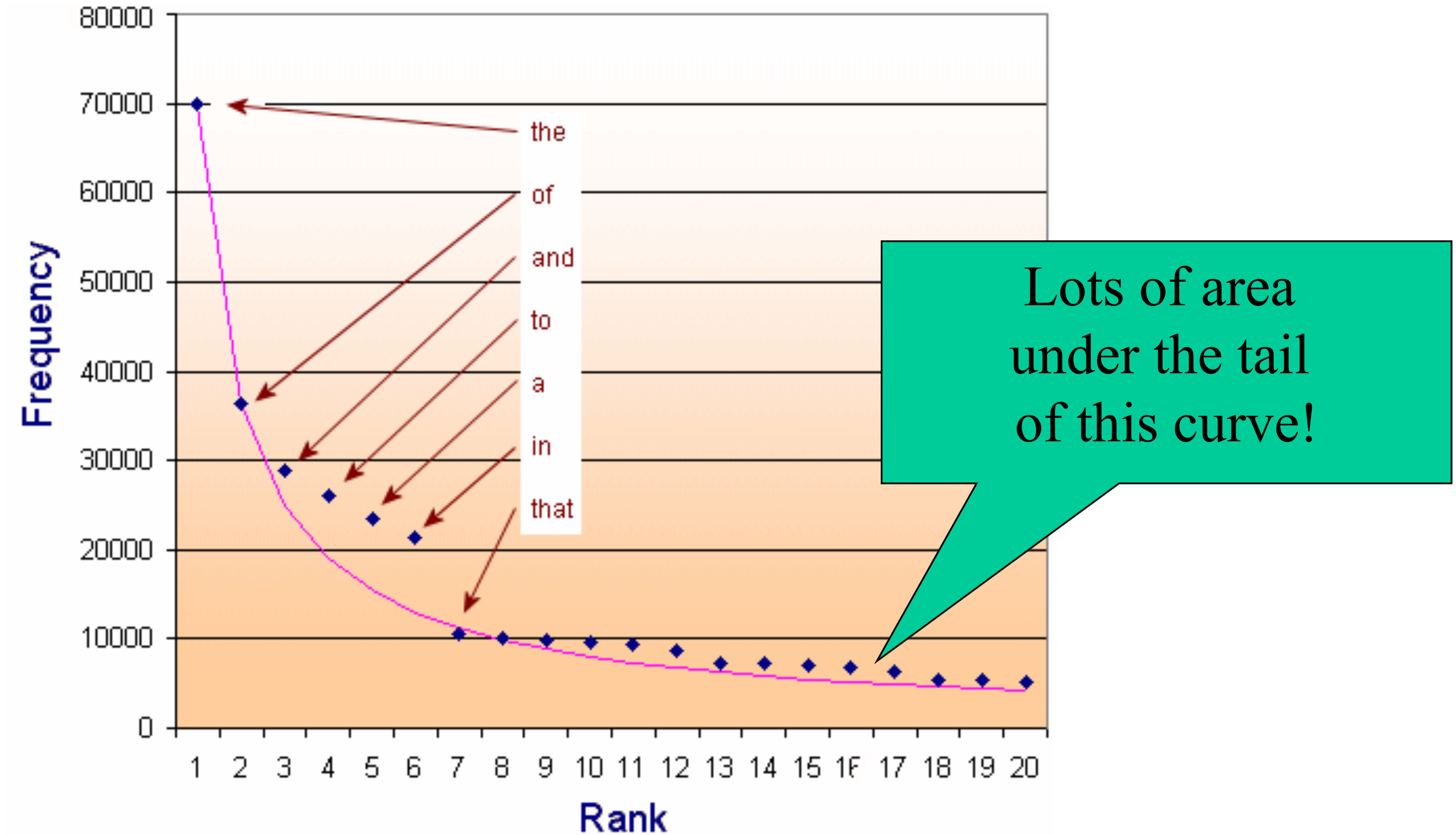
$$\mathrm{r} = \mathrm{C/f}$$

$$\log(\mathrm{r}) = \log(\mathrm{C}) - \log(\mathrm{f})$$

- **Purely Zipfian data plots as a straight line on a log-log scale**

Rank (r): The numerical position of a word in a list sorted by decreasing frequency (f).

Word frequency & rank in Brown Corpus vs Zipf



From: Interactive mathematics http://www.intmath.com

Zipf's law for the Brown corpus