

# Automatic morphological analysis



John Goldsmith  
Wednesday talk  
October 15, 1997



# Automatic analysis of corpora: why?

- ① Traditional view of what linguistic theory is; that is, finding the justification of a particular analysis of a particular language in the way that the theory works cross-linguistically.



# Automatic analysis of corpora: why?

- ② Resolve skepticism and concern for the actual relation between observation and theory construction in current theory -  
Building a theory which is specifically a theory of a (large) corpus, seeking those properties which tend to be consistent across corpora of the same language.



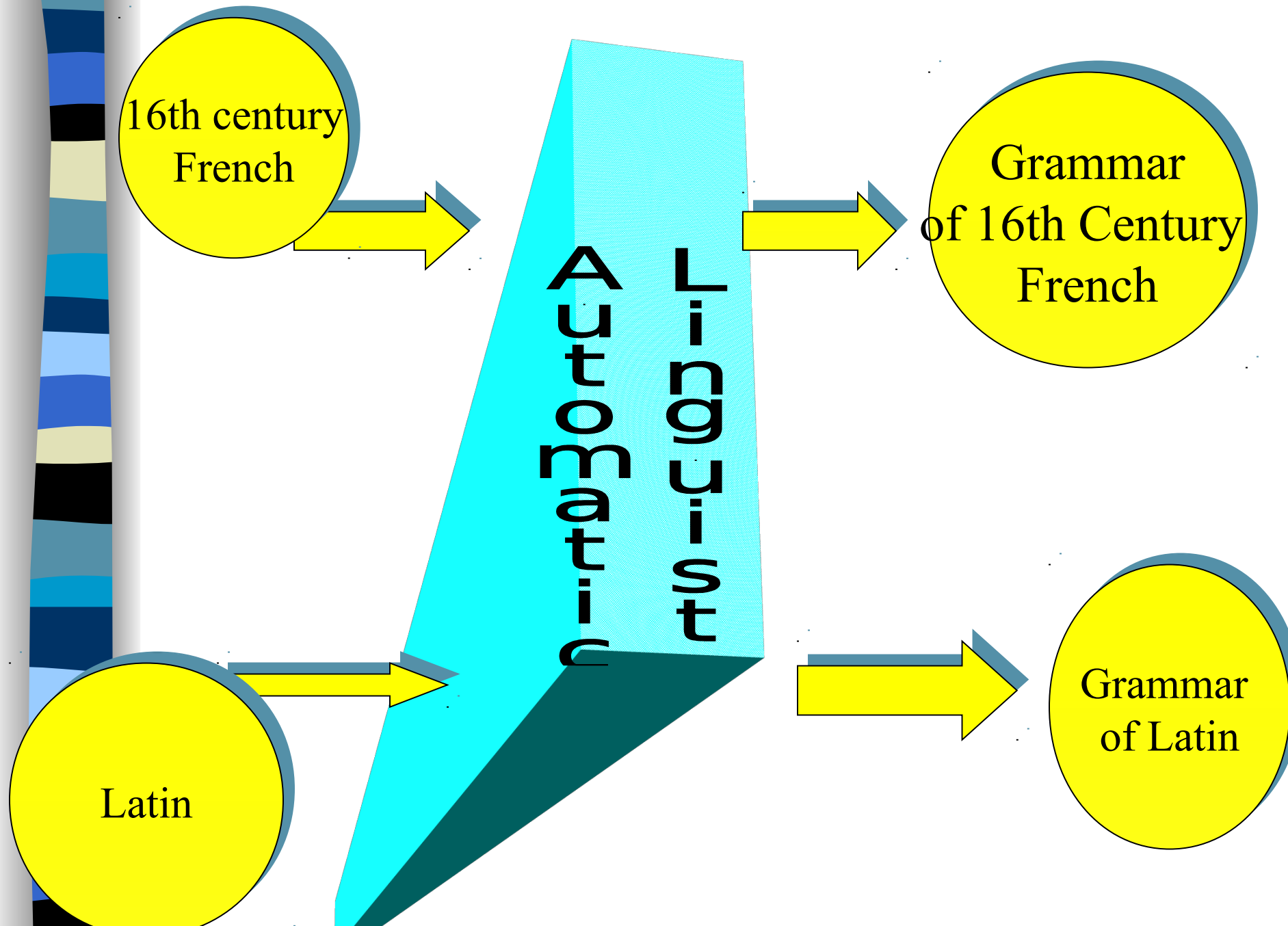
# Automatic analysis of corpora: why?

- ③ Practical applications and concerns:
  - a. rapid and accurate development for new languages
  - b. probabilistic grammars take steps towards overcoming the problem of rampant ambiguity in natural language.



# Work in progress

- Part of a larger project leading towards automatic grammatical analysis from large texts.
- Developing probabilistic grammars.



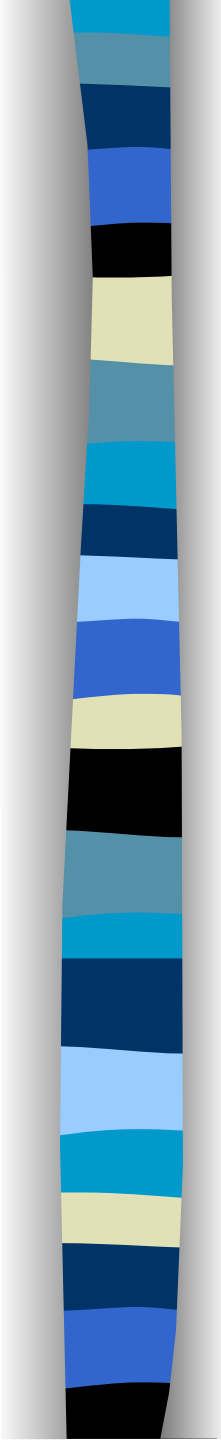
16th century  
French

Grammar  
of 16th Century  
French

Latin

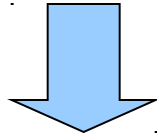
Grammar  
of Latin

A  
N  
T  
I  
Q  
U  
I  
T  
Y

- 
- No tagged corpora; no prepared corpus.
  - All learning from raw natural corpora.
  - All of these programs have been designed to work on large quantities of text.

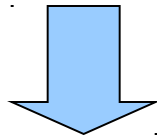
# Morphology

Start here



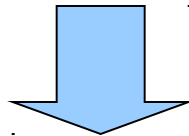
1st Morphological  
analysis

Maximize function:  
 $|\text{Stem}| + \log\text{freq}(\text{Stem}) +$   
 $|\text{Suffix}| + \log\text{freq}(\text{Suffix})$



2nd Morphological  
analysis

Collapse morphological paradigms  
that lead to optimal shortening of  
complexity of morphological grammar  
+ compressed size of corpus.

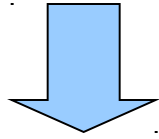


1st Morphophonological  
analysis

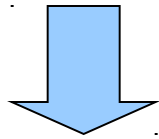
Find phonological near-neighbors  
among stems to collapse categories  
further.



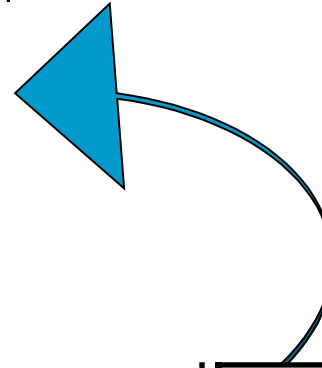
Start here



Morphology



Syntax



1. Lexical categories
2. Binary non-phrasal categories (*the dog; he is; etc.*)
3. Intervening elements (*the old dog; it really is*);
4. Constituents



# Using statistics

Basic idea:

- use probabilistic ideas to make explicit the concepts that linguists use intuitively.
- Combined with automatic search techniques, we can use computers to seek optimal analyses.



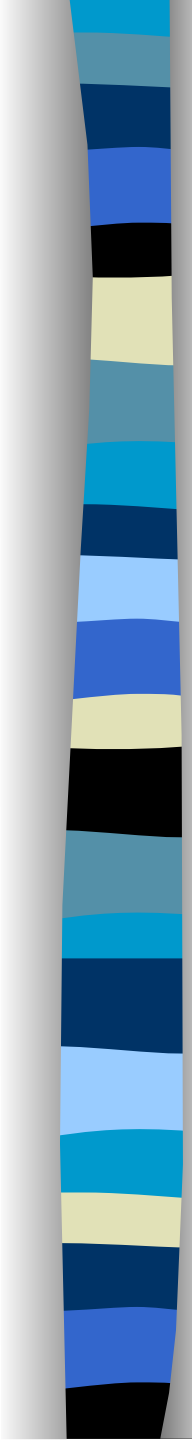
# Frequency

- Everything that follows is based on using information about word-frequencies in large corpora: information that linguistics usually ignore completely.



# Stages of morphology:

1. Determine optimal splitting of words:  
iterative procedure, using a  
measurement of how good a given split  
is, based on neighbors' current splits.
2. Determine corpus' "paradigms":
  - ed,-s (*cough, sleep*)
  - ed,-ing (*laugh, jump*)

- 
3. Collapse paradigms by trading off accuracy of word-probabilities for complexity of Morphological Grammar
  4. Seeking stem-identification (*cry* as in *crying* identified with *cri* in *cried*) based on linguistic features and decrease in complexity of Morphological Grammar.



# Two Important Ideas

1. Compressed Length of Corpus
  2. Entropy of the Morphological Lexicon  
(alternative measures of Lexicon's complexity are conceivable)
- ❖ There's a tradeoff between the two, however.



A trade-off is a good thing:

- it allows us to let the search for optimization proceed automatically;
- We can also weigh the trade-off differently and see the linguistic results of the change in optimization.

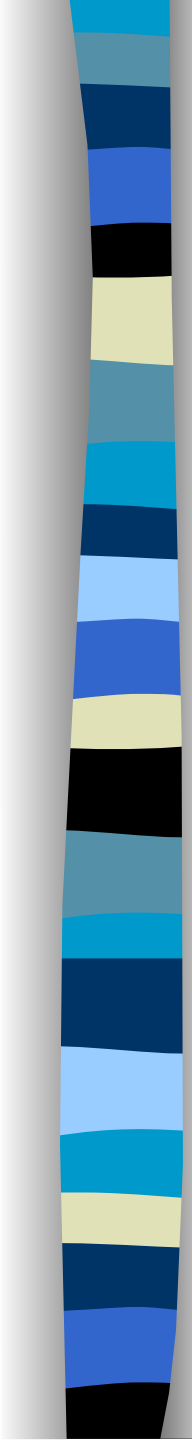
(We have traditionally *said* that in linguistics, but not done it.)



# Stages of morphology:

1. Determine optimal splitting of words: iterative procedure, using a measurement of how good a given split is, based on neighbors' current splits.
2. Determine corpus' "paradigms":
  - ed,-s (*cough, sleep*)
  - ed,-ing (*laugh, jump*)





Suppose a language learner comes to the task of learning a language with the hypothesis that words are likely to be composed of two parts. Let's call that the stem and the suffix.

(Occasionally will look like prefix and stem to us.)

That is our *first analytical step*.



# How do words decompose?

- The usual problem: the best analysis of one word depends on the analysis of all the other words in the language!
- *Audacity* is *audac* + *ity* because it forms part of a larger system with *audac* + *ious* and *mendac* + *ity*, *tenac* + *ity*, *san* + *ity*, etc.



# We can implement an iterative procedure...

... in which each word starts out entirely uncertain as to how it should be divided:

How should we break up audacity?

a udacity

au dacity

aud acity

auda city

audac ity

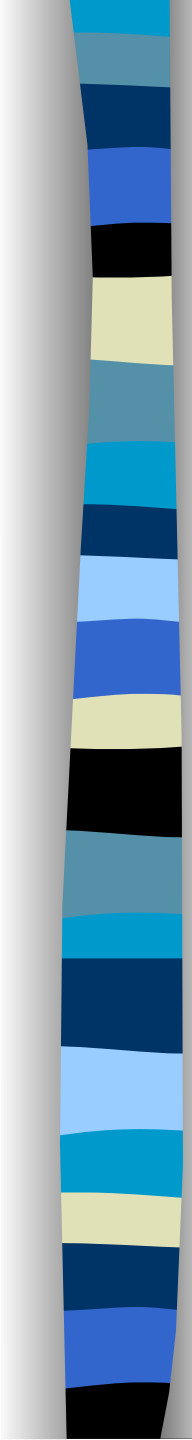
audaci ty

audacit y

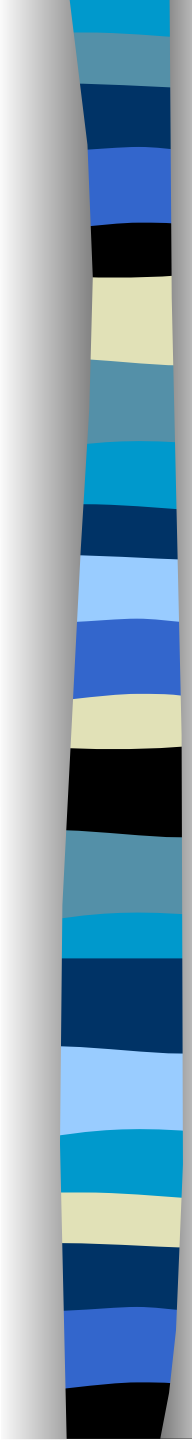


# Bootstrap:

- Each factorization is (on the first iteration) given equal “credit.” For an 8 letter word that appears 13 times in a corpus, that means  $13/(8-1)$  points to each stem form (a, au, aud, auda, audac, audici, audicit).
- Likewise for the conceivable suffixes.



Assign a weighting, and iterate -  
We want to split each word in a  
way that puts it in tune with how  
all the other words are split.



By the way -- simple counting won't do the trick:

- it will split words after their first letter or before their last letter (a -udacity and audacit- y)
- Use the formula: compute
  - Length of stem \* log frequency (stem) + length of suffix \* log frequency (suffix),
- and distribute “frequency credit” among all the factors based on that measure.



# Stages of morphology:

1. Determine optimal splitting of words: iterative procedure, using a measurement of how good a given split is, based on neighbors' current splits.

2. Determine corpus' "paradigms":

-ed,-s (*cough, sleep*)

-ed,-ing (*laugh, jump*)



Let's look at some real examples  
from Spanish





# Stages of morphology:

1. Determine optimal splitting of words: iterative procedure, using a measurement of how good a given split is, based on neighbors' current splits.
2. Determine corpus' "paradigms":
  - ed,-s (*cough, sleep*)
  - ed,-ing (*laugh, jump*)
3. **Collapse paradigms by trading off accuracy of word-probabilities for complexity of Morphological Grammar**

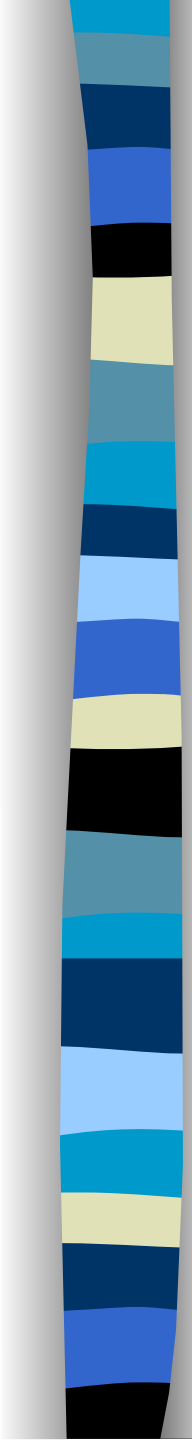


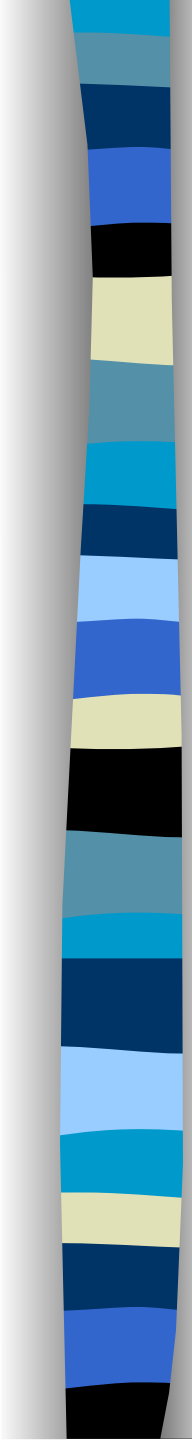
# 1. Compressed Length of Corpus

We use the frequency of words to compute the:

“Compressed Length of the Corpus”

This is a measurement of the best probabilistic measurement of the corpus, given *only* our knowledge of frequency, and none of syntax.

- 
- A probabilistic grammar assigns a probability to any and all strings of words -- including the entire corpus at hand.
  - To compute the probability of a sentence, we must assume *a set of probabilities for every word* in the lexicon: and these probabilities must add up to 1.0 (= *distribution*).



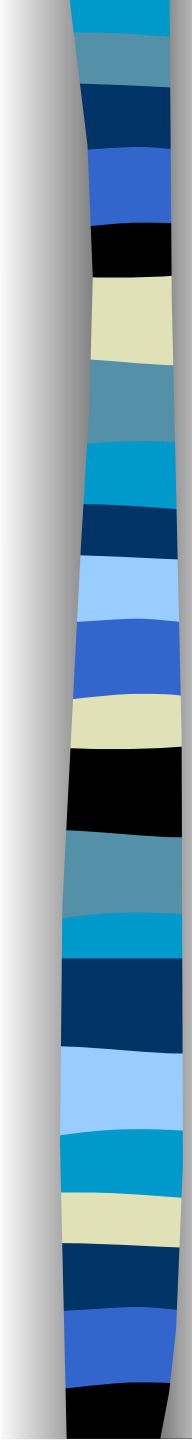
The probability of a sentence =  
the product of the probabilities that you  
have assigned to each word in the  
sentence; that is,

Log Probability (Sentence) =

$\Sigma$  Log prob (each word in the corpus);

or, if we scan through each word in the  
lexicon:

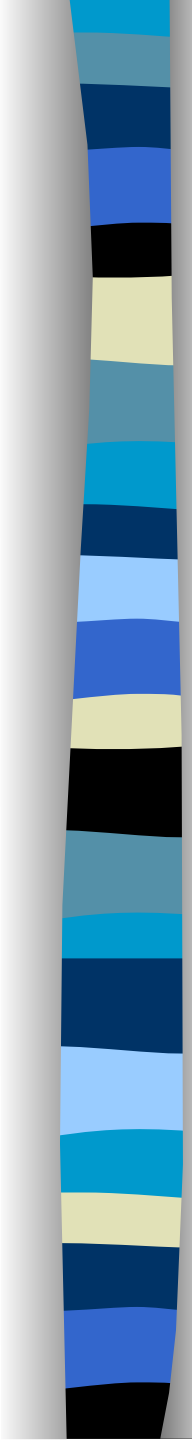
$\Sigma$  Log prob (each word<sub>i</sub> in the lexicon) x  
# occurrences (word<sub>i</sub> in the corpus) ;



$\Sigma \text{ Log prob (each word}_i \text{ in the lexicon) } \times$   
 $\# \text{ occurrences (word}_i \text{ in the corpus) ;}$

The *best* probability to assign to each word is its *actual* probability in the corpus, but nothing says we *have* to use that probability.

That is ...



Probability of a corpus =  
 $\Sigma$  probability (word<sub>i</sub>) (obtained *however*)  
x # occurrences (word<sub>i</sub> in the corpus) :  
summing over the words in the Lexicon.  
Writing it yet again in another way:



# Again...an inner product

Probability of a corpus =

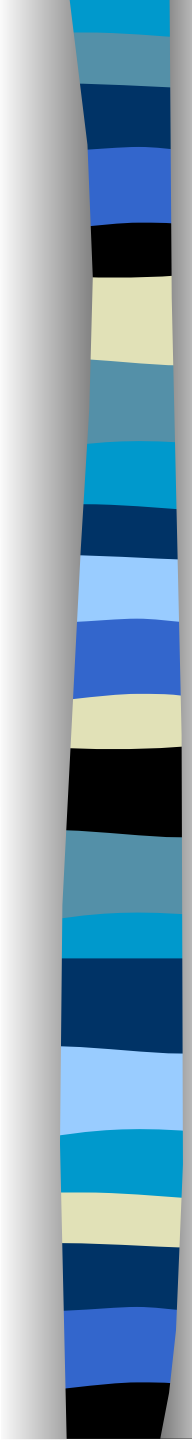
(Log-Probability Vector) \* (Distribution in this  
Corpus Vector)



Logs of a set of  
numbers  
that add up to 1.0  
(a log distribution)



A set of numbers  
that add up to 1.0  
(a distribution)



Probability of a corpus =  
(Log-Probability Vector) \* (Distribution in this  
Corpus Vector)

A basic theorem (Shannon): this number has a unique maximum when the two distributions used are the *same*.

*Significance for the linguist:* The best probabilistic grammar uses the raw observed frequencies.





But ...

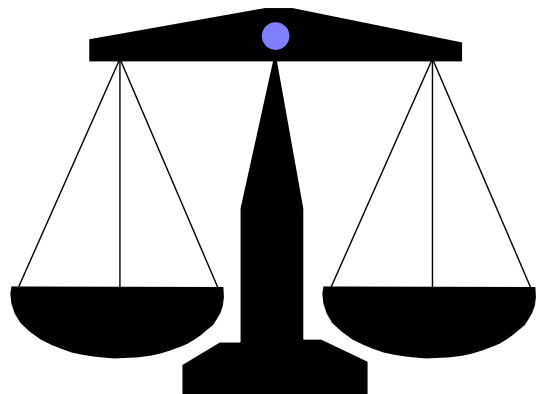
To say that a language has inflectional morphology *is* to say that the frequency

$\text{Freq}(\text{Stem}) * \text{Freq}(\text{Suffix})$ . (i)

Hence any morphological analysis will include assignment of Freq's to each stem and each suffix, and hence to each Word, by (i).

But that means that the Compression created by means of these “modeled” log Freq's will be worse than that created by raw frequencies: the trade-off.

# Basic idea of morphology search



Collapsing incomplete paradigms in the corpus will *improve* the entropy of the Morphological Lexicon (decrease it), but *worsen* the compression (by departing from observed frequencies).



That's enough for now.

- Probably more than enough.
- Results are coming in at this point: I'll keep you posted on them.