Computational Methods in Linguistics

Bender and Wassink 2012
University of Washington
Week 4: 4/16-4/18

Today

- practicum group 1
- types of frequency data
- (Jurafsky, 2003) linguistic research into comprehension and production of more frequent and less frequent forms
- implications of sampling over some corpus
- reading questions
- introduce lab 4/python tutorial

Three Uses of Frequency

Counts from corpora are just frequencies. It's the analyst's job to interpret what these frequencies *mean.* The most basic type of corpus-linguistic tool is the frequency list.

- frequency lists
- lexical co-occurrence lists (collocations)
- concordances

Type 1: Frequency Lists

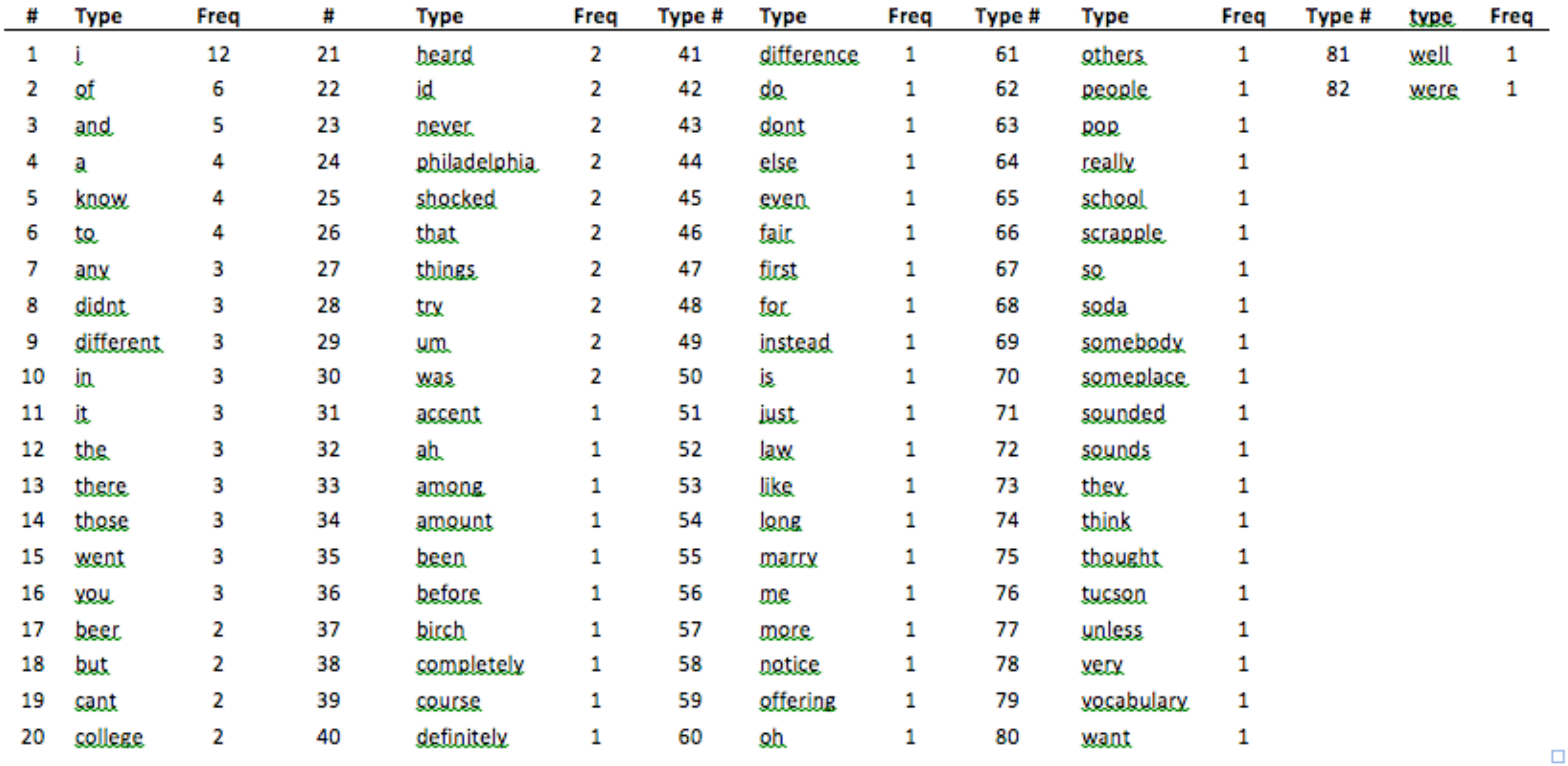
What: Tells you how often (word) “types” occur in a corpus

Form: Typically a two-column table with all “types” that occur in the corpus

How: Sorted in a variety of methods:

- alphabetical order of types
- decreasing/increasing order of frequency
- *n*-grams: sequences of *n* words with their frequencies (bigrams or word pairs, n=2; trigrams or word triples, n=3)

Scrapple text frequencies



#	Type	Freq	#	Type	Freq	Type #	Type	Freq	Type #	Type	Freq	Type #	type	Freq
1	i	12	21	heard	2	41	difference	1	61	others	1	81	well	1
2	of	6	22	id	2	42	do	1	62	people	1	82	were	1
3	and	5	23	never	2	43	dont	1	63	pop	1			
4	a	4	24	philadelphia	2	44	else	1	64	really	1			
5	know	4	25	shocked	2	45	even	1	65	school	1			
6	to	4	26	that	2	46	fair	1	66	scrapple	1			
7	any	3	27	things	2	47	first	1	67	so	1			
8	didnt	3	28	try	2	48	for	1	68	soda	1			
9	different	3	29	um	2	49	instead	1	69	somebody	1			
10	in	3	30	was	2	50	is	1	70	someplace	1			
11	it	3	31	accent	1	51	just	1	71	sounded	1			
12	the	3	32	ah	1	52	law	1	72	sounds	1			
13	there	3	33	among	1	53	like	1	73	they	1			
14	those	3	34	amount	1	54	long	1	74	think	1			
15	went	3	35	been	1	55	marry	1	75	thought	1			
16	you	3	36	before	1	56	me	1	76	tucson	1			
17	beer	2	37	birch	1	57	more	1	77	unless	1			
18	but	2	38	completely	1	58	notice	1	78	very	1			
19	cant	2	39	course	1	59	offering	1	79	vocabulary	1			
20	college	2	40	definitely	1	60	oh	1	80	want	1			

We can include a "stop list" to omit certain words (e.g., *the, and, of)*

What’s a “type”?

A unique string of characters in a corpus. We don’t use “word” because “word” has two senses in corpus linguistics:

- type: unique character string
- token: a word in the most common sense (a character string that may or may not be unique); an instantiation of a type

e.g.,

the phrase "the word and the phrase" contains:

5 tokens 1 2 3 4 5

4 types (since "the" is repeated)

Lemma

def: (plural *lemmas* or *lemmata*) is the **canonical form**, **dictionary form**, or **citation form** from which a set of words is derived (headword).

Different word forms may belong to the same *lemma*:

e.g.,

go, goes, going, a-going, went, gone belong to lemma: *go*

Type 2: Co-occurrence Relations

Three types (Also called “Association Relations” in Biber et al. 1998):

I. <u>Collocation</u>: the probabilistic co-occurrence of word forms such as *different from, different to, different than;* or frozenness of expressions such as *kith and kin, by and large*

II. <u>Colligation</u>: the co-occurrence of word forms with grammatical phenomena such as part-of-speech categories, grammatical relations or definiteness

e.g.
“consequence” tends to occur as a complement w/ an indef. article

III. <u>Collostructions</u>: the co-occurrence of words/lemmas with morpho-syntactic patterns such as the ditransitive construction or the cleft construction

e.g.,
“to hem” tends to occur in the passive

Type 3: Grammatical Co-occurrence: Concordances

- probably most widespread application of frequency in corpus linguistics today
- Tells you in which (larger) contexts a particular word is used.
- KWIC (key word in context) display

Limitations

- cannot usually extract concordance, colligation and collostruction information as easily as collocation information (R1, R2)
- easier if your corpus is syntactically annotated

Frequency in Linguistic Theory and Modeling

- Inputs to language comprehension are noisy, ambiguity is ubiquitous, the speech signal is unsegmented
- Want to understand how language is processed (identified, construed and comprehended)
- Researchers have pursued modeling of decision-making under uncertainty
    - do humans use probabilistic reasoning in cognitive tasks?

3 Roles for probability:

- access
- disambiguation
- processing

...in comprehension

- access: higher-frequency forms are retrieved from the lexicon more rapidly, with less sensory input (evidence), with less effort
- disambiguation: highly-probable forms and interpretations enable rapid segmentation, support semantic and syntactic disambiguation
- processing: facilitate understanding of what sentences types might be more difficult to process difficult tasks

...in production

- access: higher-frequency productions may be more quickly accessed from the lexicon with greater confidence and ease
- disambiguation: given multiple structures, we may choose the most probable one (phonetic strategy)
- inductive learning about cues and admissible variation

Evidence that “frequencies exist”

- Evidence that different linguistic structures have associated frequencies (words, word pairs, lexical categories, subcategorization relations)
- But what are they, and where do we find them?

http://www.auburn.edu/~nunnath/engl6240/kucera67.html

Brown Corpus

- 1,014,312 words
- 500 written texts (various genres: newspapers, novels, nonfiction, academic prose)
- Kučera and Francis 1967, type frequencies
- F &K 1982, lemmatized and POS tagged version for lemma freqs

Implications of sampling over a corpus

- Each corpus is a partial picture of language production
- Limited in representation of comprehension
- Temporally-confined
- Genre-restricted
- Therefore, no single corpus can be taken as an absolute, comprehensive representation of a form's frequency in a language (or speaker's input)
- plus side: studies have shown that frequencies from different corpora may be highly correlated, also...
- it turns out that broad-brush characterizations may be more useful than %s (binning into high, low, other)

Lexical Frequency Research: Production

- Latency effects (time between presentation and start of elicited response)
- Word duration effects (word onset to offset)
- Phonological reduction/deletion
- Phonological variation: HF words ending in /t,d/ show high levels of deletion in Chicano English (Bybee 2000)
- Speech "processing": ratio of the frequency of a derived word to that of its base is a predictor of processing time (Hay 2000)
- Picture naming: Oldfield and Wingfield (1965): found on-line effect of word frequency on latency of picture-naming (HF names > LF names)

Lexical Frequency Research: Production

- Word-to-word neighborhood effects of frequency or probability on phonetic form of a word
- Cliticization more likely in frequent word-pairs (Krug 1998)
- Sandhi: coronals at word-boundaries more likely to be palatalized b/w word sequences with high conditional probabilities (Bybee & Scheibman 1999)

Lexical Frequency Research: Production

- But,
- Few and far between
- Existing research does not provide conclusive evidence that lexical frequency effects are on-line and productive
- HF words could be stored with multiple phonological shapes, particular to individual words, resulting from diachronic change
- We may store detailed phonetic information about specific phones

Lexical Frequency Research: Comprehension

- Focus has been on ambiguity: word sense, lexical or syntactic category, morphological category
- Lexical decision tasks: Simpson and Burgess (1985)'s subjects performed lexical decision tasks on ambiguous primes. Homographs pairs in which one grapheme had HF sense, the other LF sense, HF sense retrieved more rapidly.
- Eye-fixation and gaze duration: Similar findings
- Sentence comprehension in garden-path constructions

e.g.,

- Lexical ambiguity
- Grammatical category preferences

The old man the boats.

The complex houses married and single students and their families.

Lexical Frequency Research: Comprehension

- Neighborhood effects: words prime expectations about their neighbors (McDonald 1993)
- Frequencies of semantic, syntactic, or morphological categories associated with ambiguous words play an important role in comprehension.
- HF categories preferred in disambiguation

Lexical Frequency Research: Comprehension

- But, it appears these effects may be confined to word-forms (phonology) rather than lemmas (semantics)

Reading Questions

- No necessary connection between non-modularity and probabilistic models: What are some examples of overlap?
- Does the question of modularity have implications for computational methods in linguistics?

Reading Questions

- Why would a low frequency word (“wee”) pattern with a high frequency one (“we”)?
- Why is the Brown Corpus so commonly used?
- What should we control for/count exactly when calculating word frequencies?

Reading Questions

- To what extent can we expect speakers to model different sets of frequencies for different contexts?
- Can comprehension ("processing") be probabilistic while production isn't?
- If frequencies "cause" certain choices, is there room still for other factors?

Reading Questions

- What's the difference between connectionist and probabilistic models?
- How do probabilistic models handle multimodal effects?
- What about formulaic language?
- Where does the field stand on frequency effects?

Lab 4 intro

- Tasks
- Useful python

Slide Suppement

- for the R sandbox

R primer

- Interpreted language
- Commands typically consist of:
  - functions (instruction to do something)
  - arguments (target of the operation, how to apply itself)
  - assignment operator <-
- Datastructures
  - Vector (most common): one-dimensional, sequentially-ordered sequences of elements (numbers or character strings)

- file.choose() #I can include comments this way
- my.variable <- c() #assignment occurs this way
- str() #examine the file structure
- length()
- nchar()
- getwd() #get working directory
- quit() #sometimes arg can be null

Demo

- Open conversational transcript in R
- (need to install package gsubfn)
- create a datastructure to hold the text of the transcript
- count stuff (vectors, lists)
- trim lists into words
- replace character strings, and capitals
- trim words into strings using subsetting

Split text into words

- **Example 1: cleanscrappletranscript.txt**
- Commands used in demo (first > is the prompt. Don’t type it):

```
> scrappletext<-scan(file=file.choose(), what="char", sep="\n")              #load file into memory
> (scrappletext<-gsub("'","",scrappletext, ignore.case=T))                   #replace apostrophes, print to screen
> (scrappletext<-gsub("other s","others",scrappletext, ignore.case=T)        #and replace problematic word
> words.list<-strsplit(scrappletext, "\\W")                                  #split into strings
> words.vector<-unlist(words.list)                                           #lists are chunks of strings containing
                                                                              vectors called components

> freq.list<-table(words.vector)
> sorted.freq.list<-sort(freq.list, decreasing=T)
> sorted.table<-paste(names(sorted.freq.list), sorted.freq.list, sep="\t")
> write.table(sorted.table, file=file.choose(new=T),quote=F)
```

Scripting

- combine commands into .txt file
- remove prompt character >
- invoke .txt file, or paste its contents directly at > prompt

IPA replacement

- **Example 2: IPA character replacement**
- Commands used in demo (first > is the prompt. Don’t type it):

```
> IPA.test.string<-c("iæn æɹbɹ")            #c () is the function for concatenation, or assigning value to a
datastructure
> chartr("æ","a",IPA.test.string)            #chartr() is a character replacement function
```

Trim strings

- **Example 3: using regular expressions to trim factor information from an ID code:**
- Commands used in demo (first > is the prompt. Don’t type it):

```
> PNWE.names[grep("F",PNWE.names)]
[1] "SP19CF2A" "SR02CF2A" "SN17CF1D"
> PNWE.names[grep("D",PNWE.names)]
[1] "SN17CF1D"
> grep("S?F", PNWE.names, ignore.case=T, perl=T, value=T)
[1] "SP19CF2A" "SR02CF2A" "SN17CF1D"
```

R exercise

Instructions

1. Download the file GastonTranscript.txt from the course website.
2. Launch R.
You will find a helpful tutorial on R here: Rtutorialhandout.rtf
You will find the sociolab's list of R functions here: Rtutorialfunctions.rtf

3. Using the command below, select the file, and create a datastructure called "gastontext" that will store the contents of GastonTranscript.txt in working memory. (Note: GastonTranscript.txt remains unchanged while you work with it, which is nice.)

MAC:

```
> gastontext<-scan(file=file.choose(), what="char", sep="\n", quote="",comment.char="")
```

PC:

```
> gastontext<-scan(file=choose.file(), what="char", sep="\n", quote="",comment.char="")
```

4. Take a look at the first few lines of the file to ensure it loaded properly:

```
> head(gastontext)
```

5. Create a new datastructure called "words.list". We can split the contents of "gastontext" up into words by using non-word characters "\\W" to specify word-boundaries:

```
> words.list<-strsplit(gastontext, "\\W")   #this MUST be a capital W!!!
```

Don't forget that you can check the output of each datastructure assignment to see what R is doing while you work. For example, at the next prompt, try typing ...

```
> words.list
```

to see the outcome of your last command.

A. Does the output represent tokens or types?
B. What is the word associated with string 135 in the main paragraph (the one beginning, "Alice Gaston: We was talking"?)
C. Look at the R output. You will see double-brackets[[4]] around some values, and single brackets [4] around others. What is the significance of the two types of brackets?

Instructions, cont.

6. Save the contents of the list to a vector, and generate a frequency list, ordering items from less frequent to most frequent in the corpus:

```
> words.vector<-unlist(words.list)
> freq.list<-table(words.vector)
> sorted.freq.list<-sort(freq.list, decreasing=T)
```

A. In 1-2 sentences, describe the difference between the datastructures words.list and words.vector?
B. What is the first item in the datastructure, the one with 111 occurrences?
C. Lexical forms spelled the same, but in which one begins with a capital and the other with lower-case are not pooled together in the output (the output is case-sensitive). Provide an example of one such pair, together with their frequency counts from the table.
D. Let's use the function tolower to replace all the upper-case wordforms with lower case ones and recalculate our frequencies. We will enclose the entire expression in () to print output immediately to screen:

```
> (words.vector<-tolower(words.vector))
> (freq.list<-table(words.vector))
```

7. Now, let's use "paste" to associate words with their frequencies, and create a new textfile to contain the output of this operation. The words (or "types", more properly, since they're not individual tokens any longer) all have "names" as a property. So, the datastructure sorted.freq.list is a vector of numbers, and "names" represents the words (types).

```
> sorted.table<-paste(names(sorted.freq.list), sorted.freq.list, sep="\t")
```

8. We can save the output of this operation to a tab-delimited textfile which we name as part of the operation, and which will omit the "" in the vector (don't forget the extension .txt!):

PC:

```
> write.table(sorted.table, file=choose.files(new=T),quote=F)
```

MAC:

```
> write.table(sorted.table, file=file.choose(new=T),quote=F)
```

10. Open GastonFrequencies in a software package you typically use for producing tables for class or journal article (e.g., LaTEX, Excel or Word, etc.). Format your table, using column headings (Word and Frequency as column headings) and rule lines. Paste a copy of this completed table into your assignment write-up and submit with the rest of this lab exercise.