Day 2 Basic Outline

- Building blocks/foundations of quantitative text analysis
- Justifying a term/feature frequency approach
- Selecting texts
- Selecting features
- Practical issues working with texts
- Demonstrations
- Examples
THE ELEMENTS OF TEXTUAL DATA
Some key basic concepts

(text) corpus a large and structured set of texts for analysis

types for our purposes, a unique word

tokens any word – so token count is total words
  ▶ hapax legomena (or just hapax) are types that occur just once

stems words with suffixes removed

lemmas canonical word form (the base form of a word that has the same meaning even when different suffixes (or prefixes) are attached)

keys such as dictionary entries, where the user defines a set of equivalence classes that group different word types
Some more key basic concepts

“key” words  Words selected because of special attributes, meanings, or rates of occurrence
stop words  Words that are designated for exclusion from any analysis of a text
readability  provides estimates of the readability of a text based on word length, syllable length, etc.
complexity  A word is considered “complex” if it contains three syllables or more
diversity   (lexical diversity) A measure of how many types occur per fixed word rate (a normalized vocabulary measure)
DEFINING “DOCUMENTS”
Strategies for selecting units of textual analysis

- Words
- $n$-word sequences
- Pages
- Paragraphs
- Themes
- Natural units (a speech, a poem, a manifesto)
- Key: depends on the research design
Sample v. “population”

- Basic Idea: Observed text is a stochastic realization
- Systematic features shape most of observed verbal content
- Non-systematic, random features also shape verbal content

\[ \mu: \text{“True” preferences of author} \]
\[ \text{Unobservable and uncertain} \]

\[ \pi: \text{Intended message of author} \text{ given } \mu \text{ and } M \]
\[ \text{Unobservable and uncertain} \]

\[ \tau: \text{Text generated by author} \text{ given } \pi \text{ and } T \]
\[ \text{Observable and certain} \]

\[ M \]
\[ \text{Strategic model of politics} \]

\[ T \]
\[ \text{Stochastic process of text generation} \]
Implications of a stochastic view of text

- Observed text is not the only text that could have been generated
- Very different if you are trying to monitor something like hate speech, where what you actually say matters, not the value of your “expected statement”
- Means that having “all the text” is still not a “population”
- Suggests you could employ bootstrapping strategies to estimate uncertainty for sample statistics, even things like readability
Sampling strategies for selecting texts

- Difference between a sample and a population
- May not be feasible to perform any sampling
- May not be necessary to perform any sampling
- Be wary of sampling that is a feature of the social system: “social bookkeeping”
- Different types of sampling vary from random to purposive
  - random sampling
  - non-random sampling
- Key is to make sure that what is being analyzed is a valid representation of the phenomenon as a whole – a question of research design
Random versus “Constructed” Sampling

- Based on a study by Riffe, Aust and Lacy (1993), who compared sampling from newspaper articles randomly versus “constructed”
- Either randomly sample 7 consecutive days, or between 2–4 consecutive weeks, and compare to “known” quantities
- Study showed that constructed sampling is much more efficient
- Why? Because cyclic variation in newspaper content occurs according to the day of the week – not every day contains equal proportions of different content
SELECTING FEATURES
Strategies for feature selection

- **document frequency** How many documents in which a term appears
- **term frequency** How many times does the term appear in the corpus
- **purposive selection** Use of a dictionary of words or phrases
- **deliberate disregard** Use of “stop words”: words excluded because they represent linguistic connectors of no substantive content
Common English stop words

a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because, been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, I, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your

- But no list should be considered universal
A more comprehensive list of stop words

as, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, aint, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, arent, around, as, aside, ask, asking, associated, at, available, away, awfully, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by, cmon, cs, came, can, cant, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldnt, course, currently, definitely, described, despite, did, didnt, different, do, does, doesnt, doing, dont, done, down, downwards, during, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, everybody, everyone, everything, everywhere, ex, exactly, example, except, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, had, hadnt, happens, hardly, has, hasnt, have, havent, having, he, hes, hello, help, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, id, ill, im, ive, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isnt, it, itd, itll, its, its, itself, just, keep, keeps, kept, know, knows, known, last, lately, later, latter, latterly, least, less, lest, let, lets, like, liked, likely, little, look, looking, looks, ltd, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own, particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides, que, quite, qv, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldnt, since, six, so, some, somebody, somehow, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, ts, take, taken, tell, tends, th, than, thank, thanks, thanx, that, thats, thats, the, their, theirs, them, themselves, then, thence, there, theres, thereafter, thereby, therefore, therein, theres, thereupon, these, they, theyd, theyll, theyre, theyve, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, value, various, very, via, viz, vs, want, wants, was, wasnt, way, we, wed, well, were, weve, welcome, well, went, were, werent, what, whats, whatever, when, whence, whenever, where, wheres, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whos, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, wont, wonder, would, would, wouldnt, yes, yet, you, youd, youll, youre, youve, your, yours, yourself, yourselves, zero
Strategies for feature \textit{weighting}: tf-idf

\begin{itemize}
  \item $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$

  where $n_{i,j}$ is number of occurrences of term $t_i$ in document $d_j$,
  $k$ is total number of terms in document $d_j$

  \item $idf_i = \ln \frac{|D|}{|\{d_j : t_i \in d_j\}|}$

  where

  \begin{itemize}
    \item $|D|$ is the total number of documents in the set
    \item $|\{d_j : t_i \in d_j\}|$ is the number of documents where the term $t_i$
      appears (i.e. $n_{i,j} \neq 0$)
  \end{itemize}

  \item $tf-idf_i = tf_{i,j} \cdot idf_i$
\end{itemize}
Computation of tf-idf: Example

Example: We have 100 political party manifestos, each with 1000 words. The first document contains 16 instances of the word “environment”; 40 of the manifestos contain the word “environment”.

- The term frequency is $\frac{16}{1000} = 0.016$
- The document frequency is $\frac{100}{40} = 2.5$, or $\ln(2.5) = 0.916$
- The tf-idf will then be $0.016 \times 0.916 = 0.0147$
- If the word had only appeared in 15 of the 100 manifestos, then the tf-idf would be 0.0304 (three times higher).
- A high weight in tf-idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; hence the weights hence tend to filter out common terms.
Stemming words

**Lemmatization** refers to the algorithmic process of converting words to their lemma forms.

**stemming** the process for reducing inflected (or sometimes derived) words to their stem, base or root form. Different from *lemmatization* in that stemmers operate on single words without knowledge of the context.

**both** convert the morphological variants into stem or root terms

**example:** *produc* from production, producer, produce, produces, produced
Varieties of stemming algorithms

In stemming, conversion of morphological forms of a word to its stem is done assuming each one is semantically related. The stem need not be an existing word in the dictionary but all its variants should map to this form after the stemming has been completed.

There are two points to be considered while using a stemmer:

1. Morphological forms of a word are assumed to have the same base meaning and hence should be mapped to the same stem.
2. Words that do not have the same meaning should be kept separate.

These two rules are good enough as long as the resultant stems are useful for our text mining or language processing applications. Stemming is generally considered as a recall-enhancing device. For languages with relatively simple morphology, the influence of stemming is less than for those with a more complex morphology. Most of the stemming experiments done so far are for English and other west European languages.

Lemmatizing deals with the complex process of first understanding the context, then determining the POS of a word in a sentence and then finally finding the 'lemma'. In fact an algorithm that converts a word to its linguistically correct root is called a lemmatizer.

A lemma in morphology is the canonical form of a lexeme. Lexeme, in this context, refers to the set of all the forms that have the same meaning, and lemma refers to the particular form that is chosen by convention to represent the lexeme.

In computational linguistics, a stem is the part of the word that never changes even when morphologically inflected, whilst a lemma is the base form of the verb. Stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications. Lemmatizers are difficult to implement because they are related to the semantics and the POS of a sentence. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. The results are not always morphologically right forms of words. Nevertheless, since document index and queries are stemmed "invisibly" for a user, this peculiarity should not be considered as a flaw, but rather as a feature distinguishing stemming from lemmatization.

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the lemma.

For example, the word inflations like gone, goes, going will map to the stem 'go'. The word 'went' will not map to the same stem. However a lemmatizer will map even the word 'went' to the lemma 'go'.

4. Errors in Stemming

There are mainly two errors in stemming—overstemming and understemming. Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive. Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative. Paice has proved that light-stemming reduces the over-stemming errors but increases the under-stemming errors. On the other hand, heavy stemmers reduce the under-stemming errors while increasing the over-stemming errors [14, 15].

5. Classification of Stemming Algorithms

Broadly, stemming algorithms can be classified in three groups: truncating methods, statistical methods, and mixed methods. Each of these groups has a typical way of finding the stems of the word variants. These methods and the algorithms discussed in this paper under them are shown in the Fig. 1.

Figure 1. Types of stemming algorithms

5.1. Truncating Methods (Affix Removal)

As the name clearly suggests these methods are related to removing the suffixes or prefixes (commonly known as affixes) of a word. The most basic stemmer

<table>
<thead>
<tr>
<th>Truncating</th>
<th>Statistical</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Lovins</td>
<td>1) N-Gram</td>
<td>a) Inflectional &amp; Derivational</td>
</tr>
<tr>
<td>2) Porters</td>
<td>2) HMM</td>
<td>1) Krovetz</td>
</tr>
<tr>
<td>3) Paice/Husk</td>
<td>3) YASS</td>
<td>2) Xerox</td>
</tr>
<tr>
<td>4) Dawson</td>
<td></td>
<td>b) Corpus Based</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c) Context Sensitive</td>
</tr>
</tbody>
</table>
Issues with stemming approaches

- The most common is probably the Porter stemmer
- But this set of rules gets many stems wrong, e.g.
  - policy and police considered (wrongly) equivalent
  - general becomes gener, iteration becomes iter
- Other corpus-based, statistical, and mixed approaches designed to overcome these limitations (good review in Jirvani article)
- Key for you is to be careful through inspection of morphological variants and their stemmed versions
Selecting more than words: collocations

collocations bigrams, or trigrams e.g. \textit{capital gains tax}

how to detect: pairs occurring more than by chance, by measures of $\chi^2$ or \textit{mutual information} measures

example:

<table>
<thead>
<tr>
<th>Summary Judgment</th>
<th>Silver Rudolph</th>
<th>Sheila Foster</th>
</tr>
</thead>
<tbody>
<tr>
<td>prima facie</td>
<td>COLLECTED WORKS</td>
<td>Strict Scrutiny</td>
</tr>
<tr>
<td>Jim Crow</td>
<td>waiting lists</td>
<td>Trail Transp</td>
</tr>
<tr>
<td>stare decisis</td>
<td>Academic Freedom</td>
<td>Van Alstyne</td>
</tr>
<tr>
<td>Church Missouri</td>
<td>General Bldg</td>
<td>Writings Fehrenbacher</td>
</tr>
<tr>
<td>Gerhard Casper</td>
<td>Goodwin Liu</td>
<td>boot camp</td>
</tr>
<tr>
<td>Juan Williams</td>
<td>Kurland Gerhard</td>
<td>dated April</td>
</tr>
<tr>
<td>LANDMARK BRIEFS</td>
<td>Lee Appearance</td>
<td>extracurricular activities</td>
</tr>
<tr>
<td>Lutheran Church</td>
<td>Missouri Synod</td>
<td>financial aid</td>
</tr>
<tr>
<td>Narrowly Tailored</td>
<td>Planned Parenthood</td>
<td>scored sections</td>
</tr>
</tbody>
</table>

Table 5: Bigrams detected using the mutual information measure.
COUNTING FEATURES
Word frequencies and their properties

- Individual word usage tends to be associated with a particular degree of affect, position, etc. without regard to context of word usage
- Atomic words have been found to be far more informative than $n$-grams in this regard (Benoit and Laver 2003, Midwest paper)
- Some approaches focus on occurrence of a word as a binary variable, irrespective of frequency: a binary outcome (e.g. Hopkins and King 2008)
- Other approaches use frequencies: Poisson, multinomial, and related distributions (e.g. Laver, Benoit and Garry 2003)
Word frequency: Zipf’s Law

- **Zipf’s law**: Given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.

- The simplest case of Zipf’s law is a “1/f function”. Given a set of Zipfian distributed frequencies, sorted from most common to least common, the second most common frequency will occur $1/2$ as often as the first. The third most common frequency will occur $1/3$ as often as the first. The $n$th most common frequency will occur $1/n$ as often as the first.

- In the English language, the probability of encountering the most common word is given roughly by $P(r) = 0.1/r$ for up to 1000 or so.

- The assumption is that words and phrases mentioned most often are those reflecting important concerns in every communication.
Word frequency: Zipf’s Law

- Formulaically: if a word occurs \( f \) times and has a rank \( r \) in a list of frequencies, then for all words \( f = \frac{a}{r^b} \) where \( a \) and \( b \) are constants and \( b \) is close to 1

- So if we log both sides, \( \log(f) = \log(a) - b \log(r) \)

- If we plot \( \log(f) \) against \( \log(r) \) then we should see a straight line with a slope of approximately -1.

![Graph showing log(f) against log(rank) with a linear regression line and equation y = -0.9853x + 3.6789 with R^2 = 0.9902]
Weighting strategies for feature counting

term frequency Some approaches trim very low-frequency words. Rationale: get rid of rare words that expand the feature matrix but matter little to substantive analysis

document frequency Could eliminate words appearing in few documents

inverse document frequency Conversely, could weight words more that appear in the most documents
Word concordances on popular web sites

- Amazon word statistics example http://www.amazon.com/Innovative-Comparative-Methods-Policy-Analysis/dp/0387288287/ref=sr_1_1?ie=UTF8&s=books&qid=1249293340&sr=8-1

PRACTICAL ISSUES WORKING WITH TEXT
Practical issues working with texts

File formats  How the electronic text is formatted
Conversion  Converting files from one format to another
Pre-analysis text processing  Considering inflected forms as equivalent, through lemmatization and/or stemming

dropping infrequent words  as they may not be informative

stop lists  for most frequent words
Practical issues working with texts

- Formats
- Encodings
- Managing meta-data
  - document-level meta-data (aka document “variables”)
  - corpus-level meta-data