xkcd.com/208

- Regex comic
  - http://xkcd.com/208
- Cleverbot video
  - http://www.youtube.com/watch?v=WnzlbyTZsQY

CORPUS ANALYSIS

David Kauchak
NLP – Fall 2011

Administrivia

- Assignment 0
  - submit script
  - article discussion
- Assignment 1 out
  - due Sunday 25th by midnight
  - no code submitted, but will require coding
- Send me an e-mail if you'd like me to e-mail announcements to another account besides your school account
- Send videos…

NLP models

- How do people learn/acquire language?

NLP models

- A lot of debate about how human's learn language
  - Rationalist (e.g. Chomsky)
  - Empiricist
- From my perspective (and many people who study NLP) …
  - I don't care :)
- Strong AI vs. weak AI: don't need to accomplish the task the same way people do, just the same task
  - Machine learning
  - Statistical NLP

Vocabulary

- Word
  - a unit of language that native speakers can identify
  - words are the blocks from which sentences are made
- Sentence
  - a string of words satisfying the grammatical rules of a language
- Document
  - A collection of sentences
- Corpus
  - A collection of related texts

Corpora characteristics

- monolingual vs. parallel
- language
- annotated (e.g. parts of speech, classifications, etc.)
- source (where it came from)
- size

Corpora examples

Any you've seen or played with before?

Corpora examples

- Linguistic Data Consortium
  - http://www.ldc.upenn.edu/Catalog/byType.jsp
- Dictionaries
  - WordNet – 206K English words
  - CELEX2 – 365K German words
- Monolingual text
  - Gigaword corpus
    - 4M documents (mostly news articles)
    - 1.7 trillion words
    - 11GB of data (4GB compressed)

Corpora examples

- Monolingual text continued
  - Enron e-mails
    - 517K e-mails
  - Twitter
  - Chatroom
  - Many non-English resources
- Parallel data
  - ~10M sentences of Chinese-English and Arabic-English
  - Europarl
    - ~1.5M sentences English with 10 different languages

Corpora examples

- Annotated
  - Brown Corpus
    - 1M words with part of speech tag
  - Penn Treebank
    - 1M words with full parse trees annotated
  - Other treebanks
    - Treebank refers to a corpus annotated with trees (usually syntactic)
    - Chinese: 51K sentences
    - Arabic: 145K words
    - many other languages…
    - BLIPP: 300M words (automatically annotated)

Corpora examples

- Many others…
  - Spam and other text classification
  - Google n-grams
    - 2006 (24GB compressed!)
    - 13M unigrams
    - 300M bigrams
    - ~1B 3,4 and 5-grams
  - Speech
  - Video (with transcripts)

Corpus analysis

- Corpora are important resources
- Often give examples of an NLP task we'd like to accomplish
- Much of NLP is data-driven!
- A common and important first step to tackling many problems is analyzing the data you'll be processing

Corpus analysis

What types of questions might we want to ask?

- How many…
  - documents, sentences, words
- On average, how long are the:
  - documents, sentences, words
- What are the most frequent words? pairs of words?
- How many different words are used?
- Data set specifics, e.g. proportion of different classes?
- …

Corpora issues

- Somebody gives you a file and says there's text in it
- Issues with obtaining the text?
  - text encoding
  - language recognition
  - formatting (e.g. web, xml, …)
  - misc. information to be removed
    - header information
    - tables, figures
    - footnotes

A rose by any other name…

- Word
  - a unit of language that native speakers can identify
  - words are the blocks from which sentences are made
- Concretely:
  - We have a stream of characters
  - We need to break into words
  - What is a word?
  - Issues/problem cases?
  - Word segmentation/tokenization?

Tokenization issues: ‘

Finland’s capital…

?

Tokenization issues: ‘

Finland’s capital…

Finland

Finland ‘ s

Finland ‘s

Finlands

Finland s

Finland’s

What are the benefits/drawbacks?

Tokenization issues: ‘

Aren’t we …

?

Tokenization issues: ‘

Aren’t we …

Aren’t

Arent

Are n’t

Aren t

Are not

Tokenization issues: hyphens

Hewlett-Packard ***state-of-the-art***

co-education ***lower-case***

take-it-or-leave-it ***26-year-old***

?

Tokenization issues: hyphens

Hewlett-Packard ***state-of-the-art***

co-education ***lower-case***

- Keep as is
- merge together
  - HewlettPackard
  - stateoftheart
- Split on hyphen
  - lower case
  - co education

What are the benefits/ drawbacks?

More tokenization issues

- Compound nouns: San Francisco, Los Angelos, …
  - One token or two?
- Numbers
  - Examples
    - Dates: 3/12/91
    - Model numbers: B-52
    - Domain specific numbers: PGP key - 324a3df234cb23e
    - Phone numbers: (800) 234-2333
    - Scientific notation: 1.456 e-10

Tokenization: language issues

Lebensversicherungsgesellschaftsangestellter

'life insurance company employee'

- Opposite problem we saw with English (San Francisco)
- German compound nouns are not segmented
- German retrieval systems frequently use a **compound splitter** module

Tokenization: language issues

莎拉波娃现在居住在美国东南部的佛罗里达。

Where are the words?

thisissue

- Many character based languages (e.g. Chinese) have no spaces between words
  - A word can be made up of one or more characters
  - There is ambiguity about the tokenization, i.e. more than one way to break the characters into words
  - Word segmentation problem
  - can also come up in speech recognition

Word counts

- *Tom Sawyer*
  - How many words?
    - 71,370 total
    - 8,018 unique
  - Is this a lot or a little? How might we find this out?
    - Random sample of news articles: 11K unique words
  - What does this say about *Tom Sawyer*?
    - Simpler vocabulary (colloquial, audience target, etc.)

Word counts

What are the most frequent words?

What types of words are most frequent?

Word	Frequency
the	3332
and	2972
a	1775
to	1725
of	1440
was	1161
it	1027
in	906
that	877
he	877
I	783
his	772
you	686
Tom	679
with	642

Word counts

8K words in vocab
71K total occurrences

how many occur once? twice?

Word Frequency	Frequency of frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

Zipf's "Law"

George Kingsley Zipf
1902-1950

- The frequency of the occurrence of a word is inversely proportional to it's frequency of occurrence ranking
- When both are plotted on a log scale, the graph is a straight line

Zipf's law

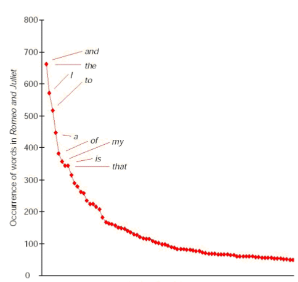
- At a high level:
  - a *few* words occur *very frequently*
  - a medium number of elements have medium frequency
  - *many* words occur *very infrequently*

Zipf's law

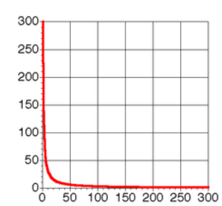
$$f = C\,\frac{1}{r}$$

The product of the frequency of words (f) and their rank (r) is approximately constant

Constant is corpus dependent, but generally grows roughly linearly with the amount of data



Zipf Distribution



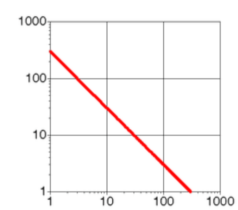
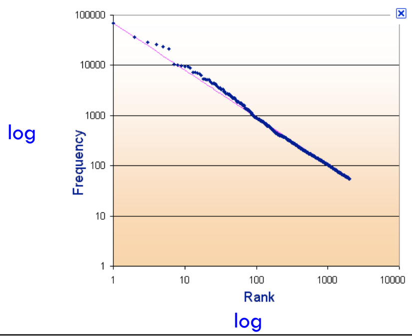


Illustration by Jacob Nielsen

Zipf's law: Brown corpus



Zipf's law: *Tom Sawyer*

Word	Frequency	Rank	f * r
the	3332	1	3332
and	2972	2	5944
a	1775	3	5235
he	877	10	8770
but	410	20	8400
be	294	30	8820
Oh	116	90	10440
two	104	100	10400
name	21	400	8400
group	13	600	7800
friends	10	800	8000
family	8	1000	8000
sins	2	3000	6000
Applausive	1	8000	8000

Sentences

- Sentence
  - a string of words satisfying the grammatical rules of a language
- Sentence segmentation
  - How do we identify a sentence?
  - Issues/problem cases?
  - Approach?

Sentence segmentation: issues

- A first answer:
  - something ending in a: . ? !
  - gets 90% accuracy

Dr. Kauchak gives us just the right amount of homework.

Abbreviations can cause problems

Sentence segmentation: issues

- A first answer:
  - something ending in a: . ? !
  - gets 90% accuracy

The scene is written with a combination of unbridled passion and sure-handed control: In the exchanges of the three characters and the rise and fall of emotions, Mr. Weller has captured the heartbreaking inexorability of separation.

sometimes: : ; and – might also denote a sentence split

Sentence segmentation: issues

- A first answer:
  - something ending in a: . ? !
  - gets 90% accuracy

"You remind me," she remarked, "of your mother."

Quotes often appear outside the ending marks

Sentence segmentation

- Place initial boundaries after: . ? !
- Move the boundaries after the quotation marks, if they follow a break
- Remove a boundary following a period if:
  - it is a known abbreviation that doesn't tend to occur at the end of a sentence (Prof., vs.)
  - it is preceded by a known abbreviation and not followed by an uppercase word

Sentence length

What is the average sentence length, say for news text? 23

Length	percent	cumul. percent
1-5	3	3
6-10	8	11
11-15	14	25
16-20	17	42
21-25	17	59
26-30	15	74
31-35	11	86
36-40	7	92
41-45	4	96
46-50	2	98
51-100	1	99.99
101+	0.01	100

Regular expressions

- Regular expressions are a very powerful tool to do string matching and processing
- Allows you to do things like:
  - Tell me if a string starts with a lowercase letter, then is followed by 2 numbers and ends with "ing" or "ion"
  - Replace all occurrences of one or more spaces with a single space
  - Split up a string based on whitespace or periods or commas or …
  - Give me all parts of the string where a digit is proceeded by a letter and then the '#' sign

Regular expressions: literals

- We can put any string in a regular expression
  - /test/
    - matches any string that has "test" in it
  - /this class/
    - matches any string that has "this class" in it
  - /Test/
    - case sensitive: matches any string that has "Test" in it

Regular expressions: character classes

- A set of characters to match:
  - put in brackets: []
  - [abc] matches a single character a or b or c
- For example:
  - /[Tt]est/
    - matches any string with "Test" or "test" in it
- Can use – to represent ranges
    - [a-z] is equivalent to [abcdefghijklmnopqrstuvwxyz]
    - [A-D] is equivalent to [ABCD]
    - [0-9] is equivalent to [0123456789]

Regular expressions: character classes

- For example:
  - /[0-9][0-9][0-9][0-9]/
    - matches any four digits, e.g. a year
- Can also specify a set NOT to match
  - ^ means all character EXCEPT those specified
  - [^a] all characters except 'a'
  - [^0-9] all characters except numbers
  - [^A-Z] not an upper case letter

Regular expressions: character classes

- Meta-characters (not always available)
  - \w - word character (a-zA-Z_0-9)
  - \W - non word-character (i.e. everything else)
  - \d - digit (0-9)
  - \s - whitespace character (space, tab, endline, …)
  - \S - non-whitespace
  - \b matches a word boundary (whitespace, beginning or end of line)
  - . - matches any character

For example

- /19\d\d/
  - would match any 4 digits starting with 19
- /\s/
  - matches anything with a whitespace (space, tab, etc)
- /\S/ or /[^\s]/
  - matches anything with at least one non-space character

Regular expressions: repetition

- * matches zero or more of the preceding
  - /ba*d/
    - matches any string with:
      - bd
      - bad
      - baad
      - baaad
  - /A.*A/
    - matches any string starts and ends with A
- + matches **one** or more of the preceding
  - /ba+d/
    - matches any string with
      - bad
      - baad
      - baaad
      - baaaad

Regular expressions: repetition

- ? zero or 1 occurrence of the preceding
  - /fights?/
    - matches any string with "fight" or "fights" in it
- {n,m} matches n to m inclusive
  - /ba{3,4}d/
  - matches any string with
    - baaad
    - baaaad

Regular expressions: beginning and end

- ^ marks the beginning of the line
- $ marks the end of the line

- /test/
  - test can occur anywhere
- /^test/
  - must start with test
- /test$/
  - must end with test
- /^test$/
  - must be exactly test

Regular expressions: repetition revisited

- What if we wanted to match:
  - This is very interesting
  - This is very very interesting
  - This is very very very interesting
- Would /This is very+ interesting/ work?
  - No… + only corresponds to the 'y'
  - /This is (very)+interesting/

Regular expressions: disjunction

- | has the lowest precedence and can be used
  - /cats|dogs/
    - matches:
      - cats
      - dogs
    - does NOT match:
      - catsogs
  - /^I like (cats|dogs)$/
    - matches:
      - I like cats
      - I like dogs

Some examples

- All strings that start with a capital letter
- IP addresses
  - 255.255.122.122
- Matching a decimal number
- All strings that end in ing
- All strings that end in ing or ed
- All strings that begin and end with the same character

Some examples

- All strings that start with a capital letter
  - /^[A-Z]/
- IP addresses
  - /\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b/
- Matching a decimal number
  - /[-+]?[0-9]*\.?[0-9]+/
- All strings that end in ing
  - /ing$/
- All strings that end in ing or ed
  - /ing|ed$/

Regular expressions: memory

- All strings that begin and end with the same character
- Requires us to know what we matched already
- ()
  - used for precedence
  - also records a matched grouping, which can be referenced later
- /^(.).*\1$/
  - all strings that begin and end with the same character

Regular expression: memory

- /She likes (\w+) and he likes \1/
- We can use multiple matches
  - /She likes (\w+) and (\w+) and he also likes \1 and \2/

Regular expressions: substitution

- Most languages also allow for substitution
  - s/banana/apple/
    - substitute first occurrence banana for apple
  - s/banana/apple/g
    - substitute all occurrences (globally)
  - s/^(.*)$/\1 \1/
  - s/\s+/ /g

Regular expressions by language

- Java: as part of the String class
  - String s = "this is a test"
  - s.matches("test")
  - s.matches(".*test.*")
  - s.matches("this\\sis .* test")
  - s.split("\\s+")
  - s.replaceAll("\\s+", " ");
  - Be careful, matches must match the whole string (i.e. an implicit ^ and $)

Regular expressions by language

- Java: java.util.regex
  - Full regular expression capabilities
  - Matcher class: create a matcher and then can use it

```
String s = "this is a test"
Pattern pattern = Pattern.compile("is\\s+")
Matcher matcher = pattern.matcher(s)
```

- matcher.matches()
- matcher.find()
- matcher.replaceAll("blah")
- matcher.group()

Regular expressions by language

- perl:
  - $s = "this is a test"
  - $s =~ /test/
  - $s =~ /^test$/
  - $s =~ /this\sis .* test/
  - split /\s+/, $s
  - $s =~ s/\s+/ /g

Regular expressions by language

- Python:
  - import re

  - s = "this is a test"
  - p = re.compile("test")
  - p.match(s)

  - p = re.compile(".*test.*")
  - re.split('\s+', s)
  - re.sub('\s+', ' ', s)

Regular expression by language

- grep
  - command-line tool for regular expressions (general regular expression print/parser)
  - returns all lines that match a regular expression
  - grep "@" twitter.posts
  - grep "http:" twiter.posts
  - can't used metacharacters (\d, \w), use [] instead
  - Often want to use "grep –E" (for extended syntax)

Regular expression by language

- sed
  - another command-line tool that uses regexs to print and manipulate strings
  - very powerful, though we'll just play with it
  - Most common is substitution:
    - sed "s/ is a / is not a/g" twitter.posts
    - sed "s/ */ /g" twitter.posts
      - sed doesn't have +, but does have *
  - Can also do things like delete all that match, etc.

Regular expression resources

- General regular expressions:
  - Ch 2.1 of the book
  - http://www.regular-expressions.info/
    - good general tutorials
    - many language specific examples as well
- Java
  - http://download.oracle.com/javase/tutorial/essential/regex/
  - See also the documentation for java.util.regex
- Python
  - http://docs.python.org/howto/regex.html
  - http://docs.python.org/library/re.html

Regular expression resources

- Perl
  - http://perldoc.perl.org/perlretut.html
  - http://perldoc.perl.org/perlre.html
- grep
  - See the write-up at the end of Assignment 1
  - http://www.panix.com/~elflord/unix/grep.html
- sed
  - See the write-up at the end of Assignment 1
  - http://www.grymoire.com/Unix/Sed.html
  - http://www.panix.com/~elflord/unix/sed.html