Text Processing

Lecture 4

Processing Text

- Converting documents to *index terms*
- Why?
  - Matching the exact string of characters typed by the user is too restrictive
    - i.e., it doesn't work very well in terms of effectiveness
  - Not all words are of equal value in a search
  - Sometimes not clear where words begin and end
    - Not even clear what a word is in some languages
      - e.g., Chinese, Korean

Text Statistics

- Huge variety of words used in text but
- Many statistical characteristics of word occurrences are predictable
  - e.g., distribution of word counts
- Retrieval models and ranking algorithms depend heavily on statistical properties of words
  - e.g., important words occur often in documents but are not high frequency in collection

Zipf's Law

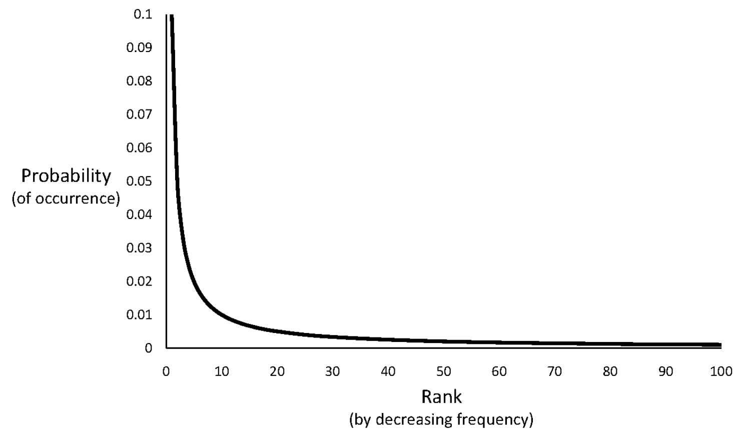
- Distribution of word frequencies is very *skewed*
  - a few words occur very often, many words hardly ever occur
  - e.g., two most common words ("the", "of") make up about 10% of all word occurrences in text documents
- Zipf's "law":
  - observation that rank (r) of a word times its frequency (f) is approximately a constant (k)
    - assuming words are ranked in order of decreasing frequency

      -> $r*f \approx k$

or $r*P_r \approx c$

where P_r is probability of word occurrence (i.e. frequency of the word divided by the total number of word occurrences in the text; P=f/N) and $c \approx 0.1$ for English

Zipf's Law



5

News Collection (AP89) Statistics

Total documents	84,678
Total word occurrences	39,749,179
Vocabulary size	198,763
Words occurring > 1000 times	4,169
Words occurring once	70,064

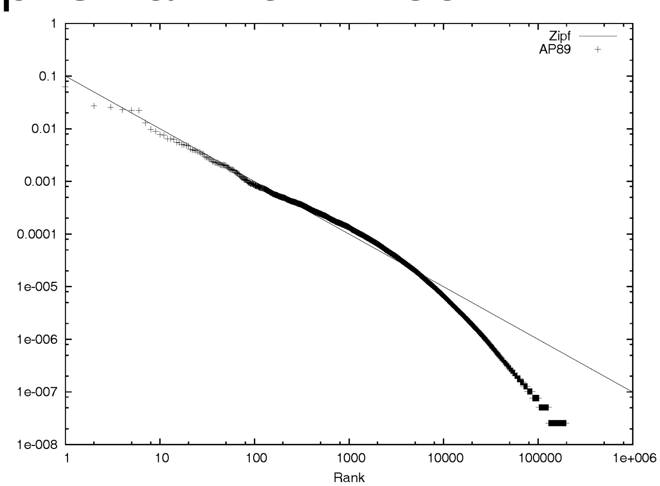
Word	*Freq.*	*r*	*Pr(%)*	*r.Pr*
assistant	5,095	1,021	.013	0.13
sewers	100	17,110	2.56 × 10−4	0.04
toothbrush	10	51,555	2.56 × 10−5	0.01
hazmat	1	166,945	2.56 × 10−6	0.04

6

Top 50 Words from AP89

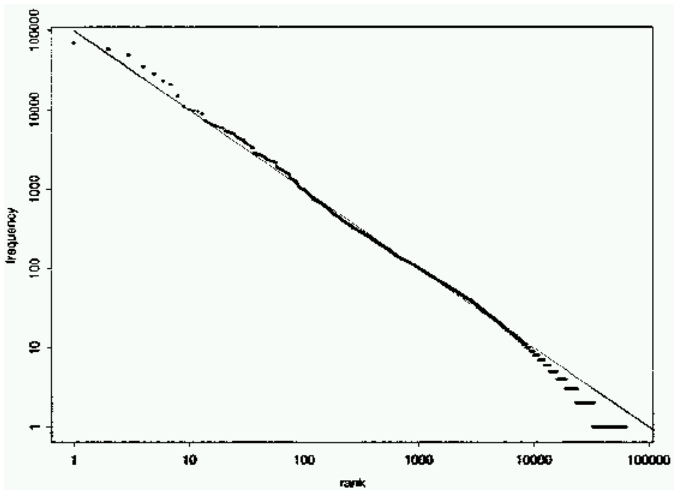
Word	Freq.	r	$P_r(\%)$	$r.P_r$	Word	Freq	r	$P_r(\%)$	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

Zipf's Law for AP89



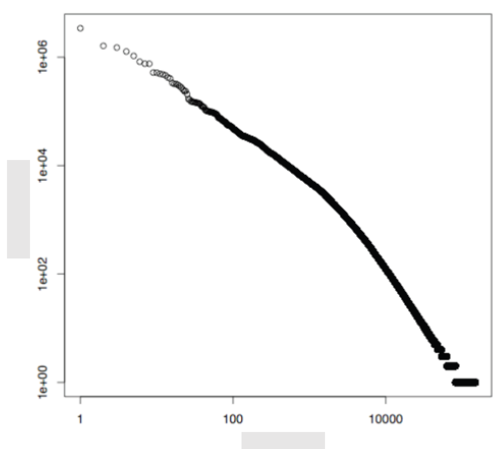
- Note problems at high and low frequencies

Fit to Zipf for Brown Corpus



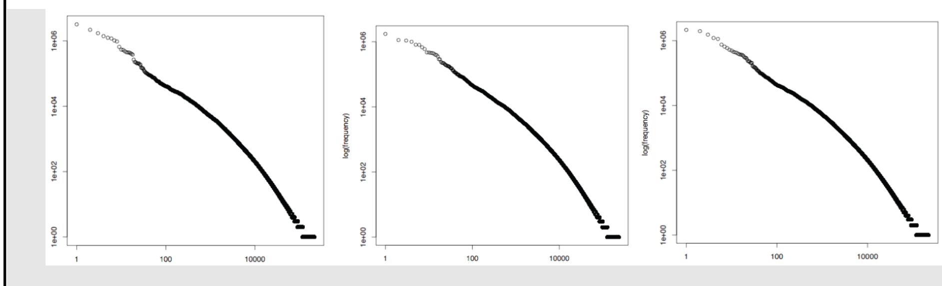
9

Fit to Zipf for European Parliament Corpus in English (Koehn '05)



10

Does Zipf's Law generalize across languages?

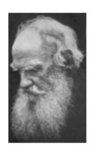


CS 4250 – Web Search and Recommender Systems

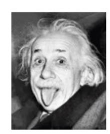
11

Across different texts

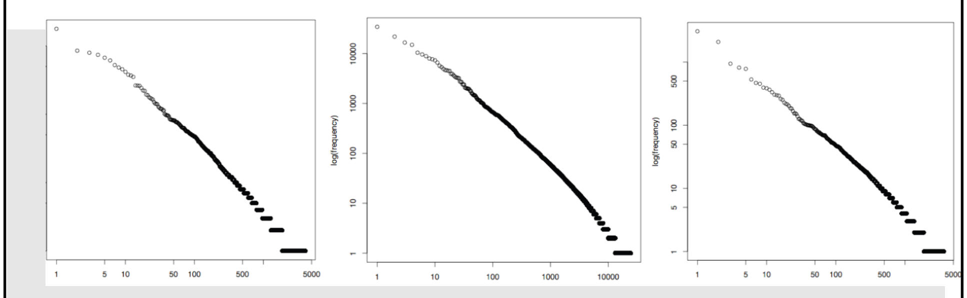
Alice in Wonderland



War and Peace



Relativity



CS 4250 – Web Search and Recommender Systems

12

Zipf's Law

Zipf's Law holds true for:

- different languages
- different sizes of text
- different genres
- different topics
- different complexity of content

Vocabulary Growth

- As corpus grows, so does vocabulary size
  - Fewer new words when corpus is already large
- Observed relationship (*Heaps' Law*):

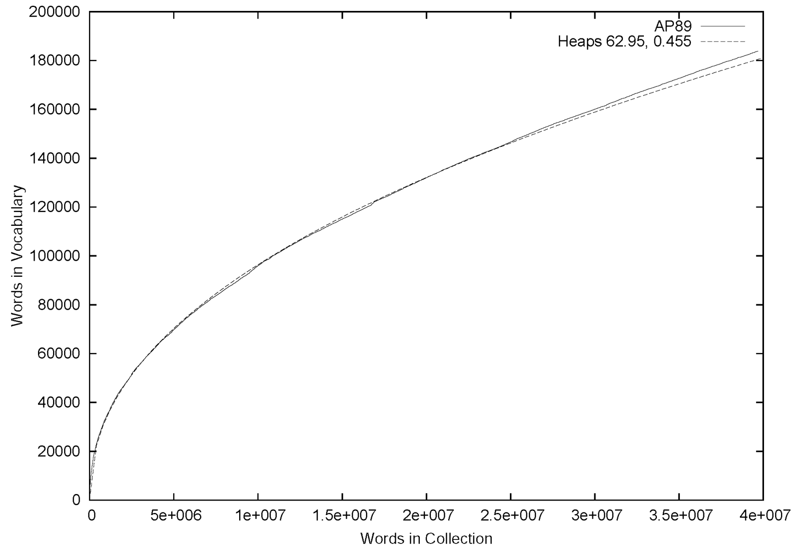
$$v = k.n^{\beta}$$

where v is vocabulary size (number of unique words), n is the number of words in corpus

k, β are parameters that vary for each corpus

(typical values given are $10 \leq k \leq 100$ and $\beta \approx 0.5$)

AP89 Example



CS 4250 – Web Search and Recommender Systems

15

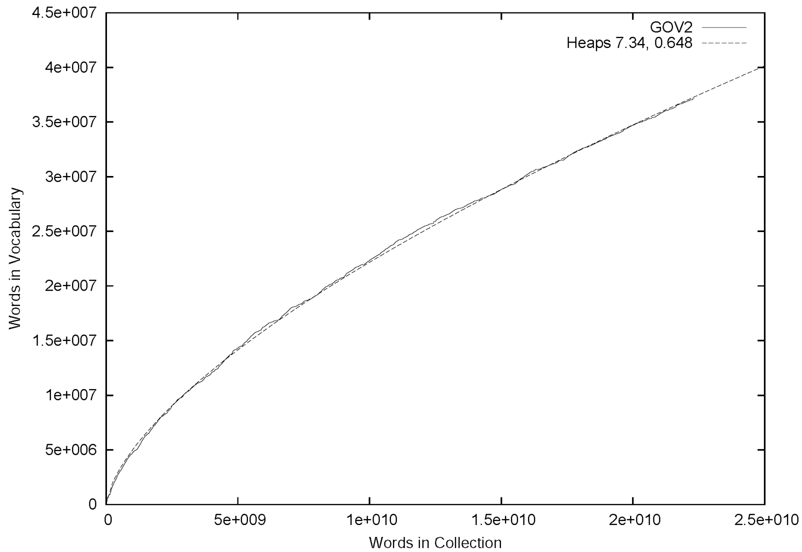
Heaps' Law Predictions

- Predictions for TREC collections are accurate for large numbers of words
  - e.g., first 10,879,522 words of the AP89 collection scanned
  - prediction is 100,151 unique words
  - actual number is 100,024
- Predictions for small numbers of words (i.e. < 1000) are much worse

CS 4250 – Web Search and Recommender Systems

16

GOV2 (Web) Example



Heaps' Law - Web

- Heaps' Law works with very large corpora
  - new words occurring even after seeing 30 million!
  - parameter values different than typical TREC values
- New words come from a variety of sources
  - spelling errors, invented words (e.g. product, company names), code, other languages, email addresses, etc.
- Search engines must deal with these large and growing vocabularies

Estimating Result Set Size

- How many pages contain *all* of the query terms?
- Assuming independence

$$P(a \cap b \cap c) = P(a) \cdot P(b) \cdot P(c)$$

- For the query "*a b c*":

$f_{abc} = N \cdot f_a/N \cdot f_b/N \cdot f_c/N = (f_a \cdot f_b \cdot f_c)/N^2$

  - Assuming that terms occur independently
  - f_{abc} is the estimated size of the result set
  - f_a, f_b, f_c are the number of documents that terms *a*, *b*, and *c* occur in
  - *N* is the number of documents in the collection

GOV2 Example

Word(s)	*Document Frequency*	*Estimated Frequency*
tropical	120,990	
fish	1,131,855	
aquarium	26,480	
breeding	81,885	
tropical fish	18,472	5,433
tropical aquarium	1,921	127
tropical breeding	5,510	393
fish aquarium	9,722	1,189
fish breeding	36,427	3,677
aquarium breeding	1,848	86
tropical fish aquarium	1,529	6
tropical fish breeding	3,629	18

Collection size (*N*) is 25,205,179

Result Set Estimation

- Even better estimates using initial result set
  - Estimate is simply *C/s*
    - where *s* is the proportion of the total documents that have been ranked, and *C* is the number of documents found that contain all the query words
  - E.g., "tropical fish aquarium" in GOV2
    - after processing 3,000 out of the 26,480 documents that contain "aquarium", *C* = 258

      $f_{tropical \cap fish \cap aquarium}$ = 258/(3000÷26480) = 2,277
    - After processing 20% of the documents,

      $f_{tropical \cap fish \cap aquarium}$ = 1,778 (1,529 is real value)

Estimating collection size

- Again using independence

$$f_{ab}/N = f_a/N \cdot f_b/N$$

$$N = (f_a \cdot f_b)/f_{ab}$$

- To get a reasonable estimate *of N,* the two chosen words should be independent
  - E.g. "tropical Lincoln" f_{ab}=3,018
    "tropical" f_a=120,990
    "lincoln" f_b=771,325
    N= (120,990 • 771,326)73,018 = 30,922,045
    Actual: 25,205,179

Tokenizing

- Forming words from sequence of characters
- Surprisingly complex in English, can be harder in other languages
- Early IR systems:
  - any sequence of alphanumeric characters of length 3 or more
  - terminated by a space or other special character
  - upper-case changed to lower-case

Tokenizing

- Example:
  - “Bigcorp's 2007 bi-annual report showed profits rose 10%.” becomes
  - “bigcorp 2007 annual report showed profits rose”
- Too simple for search applications or even large-scale experiments
- Why? Too much information lost
  - Small decisions in tokenizing can have major impact on effectiveness of some queries

Tokenizing Problems

- Small words can be important in some queries, usually in combinations
    - xp, ma, pm, ben e king, el paso, master p, gm, j lo, world war II
- Both hyphenated and non-hyphenated forms of many words are common
  - Sometimes hyphen is not needed
    - e-bay, wal-mart, active-x, cd-rom, t-shirts
  - At other times, hyphens should be considered either as part of the word or a word separator
    - winston-salem, mazda rx-7, e-cards, pre-diabetes, t-mobile, spanish-speaking

Tokenizing Problems

- Special characters are an important part of tags, URLs, code in documents
- Capitalized words can have different meaning from lower case words
  - Bush, Apple
- Apostrophes can be a part of a word, a part of a possessive, or just a mistake
  - brian o'driscoll, can't, don't, 80's, 1890's, men's straw hats, master's degree, england's ten largest cities, shriner's

Tokenizing Problems

- Numbers can be important, including decimals
  - nokia 3250, top 10 courses, united 93, quicktime 6.5 pro, 92.3 the beat
- Periods can occur in numbers, abbreviations, URLs, ends of sentences, and other situations
  - I.B.M., Ph.D., cs.umass.edu, F.E.A.R.
- **Note: tokenizing steps for queries must be identical to steps for documents**

Tokenizing Process

- First step is to use parser to identify appropriate parts of document to tokenize (e.g. finding the main content in HTML)
- Defer complex decisions to other components
  - word is any sequence of alphanumeric characters, terminated by a space or special character, with everything converted to lower-case
  - incorporate some rules to reduce dependence on query transformation components

Tokenizing Process

- Examples of rules used with TREC
  - Apostrophes in words ignored
    - o'connor → oconnor bob's → bobs
  - Periods in abbreviations ignored
    - I.B.M. → ibm

29

Stopping

- Function words (determiners, prepositions) have little meaning on their own
- High occurrence frequencies
- Treated as *stopwords* (i.e. removed)
  - reduce index space, improve response time, improve effectiveness
- Can be important in combinations
  - e.g., "to be or not to be"

30

Stopping

- Stopword list can be created from high-frequency words or based on a standard list
- Lists are customized for applications, domains, and even parts of documents
  - e.g., “click” is a good stopword for anchor text
- Best policy is to index all words in documents, make decisions about which words to use at query time

Stemming

- Many morphological variations of words
  - *inflectional* (plurals, tenses)
  - *derivational* (making verbs nouns etc.)
- In most cases, these have the same or very similar meanings
- Stemmers attempt to reduce morphological variations of words to a common stem
  - usually involves removing suffixes
- Can be done at indexing time or as part of query processing (like stopwords)

Stemming

- Generally a small but significant effectiveness improvement
  - can be crucial for some languages
  - e.g., 5-10% improvement for English, up to 50% in Arabic

kitab	*a book*
kitabi	*my book*
al**kitab**	*the book*
kitabuki	*your book* (f)
kitabuka	*your book (m)*
kitabuhu	*his book*
ka**t**a**b**a	*to write*
ma**kt**a**b**a	*library, bookstore*
ma**ktab**	*office*

Words with the Arabic root **ktb**

Stemming

- Two basic types
  - Dictionary-based: uses lists of related words
  - Algorithmic: uses program to determine related words
- Algorithmic stemmers
  - *E.g. suffix-s:* remove ‘s’ endings assuming plural
    - e.g., cats → cat, lakes → lake, wiis → wii
    - Many *false negatives*: supplies → supplie
    - Some *false positives*: ups → up

Porter Stemmer

- Algorithmic stemmer used in IR experiments since the 70s
- Consists of a series of rules
- Effective in TREC
- Produces *stems* not *words*
- Makes a number of errors and difficult to modify

Porter Stemmer

- Example step (1 of 5)

Step 1a:

- Replace ***sses*** by ***ss*** (e.g., stresses → stress).
- Delete ***s*** if the preceding word part contains a vowel not immediately before the ***s*** (e.g., gaps → gap but gas → gas).
- Replace ***ied*** or ***ies*** by ***i*** if preceded by more than one letter, otherwise by ***ie*** (e.g., ties → tie, cries → cri).
- If suffix is ***us*** or ***ss*** do nothing (e.g., stress → stress).

Step 1b:

- Replace ***eed***, ***eedly*** by ***ee*** if it is in the part of the word after the first non-vowel following a vowel (e.g., agreed → agree, feed → feed).
- Delete ***ed***, ***edly***, ***ing***, ***ingly*** if the preceding word part contains a vowel, and then if the word ends in ***at***, ***bl***, or ***iz*** add ***e*** (e.g., fished → fish, pirating → pirate), or if the word ends with a double letter that is not ***ll***, ***ss*** or ***zz***, remove the last letter (e.g., falling→ fall, dripping → drip), or if the word is short, add ***e*** (e.g., hoping → hope).
- Whew!

Porter Stemmer

False positives	*False negatives*
organization/organ	european/europe
generalization/generic	cylinder/cylindrical
numerical/numerous	matrices/matrix
policy/police	urgency/urgent
university/universe	create/creation
addition/additive	analysis/analyses
negligible/negligent	useful/usefully
execute/executive	noise/noisy
past/paste	decompose/decomposition
ignore/ignorant	sparse/sparsity
special/specialized	resolve/resolution
head/heading	triangle/triangular

- Porter2 stemmer addresses some of these issues
- Approach has been used with other languages

Krovetz Stemmer

- Hybrid algorithmic-dictionary
  - Word checked in dictionary
    - If present, either left alone or replaced with “exception”
    - If not present, word is checked for suffixes that could be removed
    - After removal, dictionary is checked again
- Produces words not stems
- Comparable effectiveness
- Lower false positive rate, somewhat higher false negative

Stemmer Comparison

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Porter stemmer:
document describ market strategi carri compani agricultur chemic report predict market share chemic report market statist agrochem pesticid herbicid fungicid insecticid fertil predict sale market share stimul demand price cut volum sale

Krovetz stemmer:
document describe marketing strategy carry company agriculture chemical report prediction market share chemical report market statistic agrochemic pesticide herbicide fungicide insecticide fertilizer predict sale stimulate demand price cut volume sale

Phrases

- Many queries are 2-3 word phrases
- Phrases are
  - More precise than single words
    - e.g., documents containing “black sea” vs. two words “black” and “sea”
  - Less ambiguous
    - e.g., “big apple” vs. “apple”
- Can be difficult for ranking
    - e.g., Given query “fishing supplies”, how do we score documents with
      - exact phrase many times, exact phrase just once, individual words in same sentence, same paragraph, whole document, variations on words?

Phrases

- Text processing issue – how are phrases recognized?
- Three possible approaches:
  - Identify syntactic phrases using a *part-of-speech* (POS) tagger
  - Use word *n-grams*
  - Store word positions in indexes and use *proximity operators* in queries

41

POS Tagging

- POS taggers use rules or statistical models of text to predict syntactic tags of words
  - Example tags:
    - NN (singular noun), NNS (plural noun), VB (verb), VBD (verb, past tense), VBN (verb, past participle), IN (preposition), JJ (adjective), CC (conjunction, e.g., “and”, “or”), PRP (pronoun), and MD (modal auxiliary, e.g., “can”, “will”).
- Phrases can then be defined as simple noun groups, for example, or adjectives followed by nouns

42

Pos Tagging Example

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:
Document/NN will/MD describe/VB marketing/NN strategies/NNS carried/VBD out/IN by/IN U.S./NNP companies/NNS for/IN their/PRP agricultural/JJ chemicals/NNS ,/, report/NN predictions/NNS for/IN market/NN share/NN of/IN such/JJ chemicals/NNS ,/, or/CC report/NN market/NN statistics/NNS for/IN agrochemicals/NNS ,/, pesticide/NN ,/, herbicide/NN ,/, fungicide/NN ,/, insecticide/NN ,/, fertilizer/NN ,/, predicted/VBN sales/NNS ,/, market/NN share/NN ,/, stimulate/VB demand/NN ,/, price/NN cut/NN ,/, volume/NN of/IN sales/NNS ./.

Example Noun Phrases

TREC data		Patent data	
Frequency	*Phrase*	*Frequency*	*Phrase*
65824	united states	975362	present invention
61327	article type	191625	u.s. pat
33864	los angeles	147352	preferred embodiment
18062	hong kong	95097	carbon atoms
17788	north korea	87903	group consisting
17308	new york	81809	room temperature
15513	san diego	78458	seq id
15009	orange county	75850	brief description
12869	prime minister	66407	prior art
12799	first time	59828	perspective view
12067	soviet union	58724	first embodiment
10811	russian federation	56715	reaction mixture
9912	united nations	54619	detailed description
8127	southern california	54117	ethyl acetate
7640	south korea	52195	example 1
7620	end recording	52003	block diagram
7524	european union	46299	second embodiment
7436	south africa	41694	accompanying drawings
7362	san francisco	40554	output signal
7086	news conference	37911	first end
6792	city council	35827	second end
6348	middle east	34881	appended claims
6157	peace process	33947	distal end
5955	human rights	32338	cross-sectional view
5837	white house	30193	outer surface

Word N-Grams

- POS tagging too slow for large collections
- Simpler definition – phrase is any sequence of *n* words – known as *n-grams*
  - *bigram*: 2 word sequence, *trigram*: 3 word sequence, *unigram*: single words
  - N-grams also used at character level for applications such as OCR
- N-grams typically formed from *overlapping* sequences of words
  - i.e. move n-word "window" one word at a time in document

N-Grams

- Frequent n-grams are more likely to be meaningful phrases
- N-grams form a Zipf distribution
  - Better fit than words alone
- Could index all n-grams up to specified length
  - Much faster than POS tagging
  - Uses a lot of storage

Google N-Grams

- Web search engines index n-grams
- Google sample:

Number of tokens:	1,024,908,267,229
Number of sentences:	95,119,665,584
Number of unigrams:	13,588,391
Number of bigrams:	314,843,401
Number of trigrams:	977,069,902
Number of fourgrams:	1,313,818,354
Number of fivegrams:	1,176,470,663

- Most frequent trigram in English is “all rights reserved”
  - In Chinese, “limited liability corporation”

Document Structure and Markup

- Some parts of documents are more important than others
- Document parser recognizes structure using markup, such as HTML tags
  - Headers, anchor text, bolded text all likely to be important
  - Metadata can also be important
  - Links used for *link analysis*

Example Web Page

Tropical fish

From Wikipedia, the free encyclopedia

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species. Fishkeepers often use the term *tropical fish* to refer only those requiring fresh water, with saltwater tropical fish referred to as *marine fish*.

Tropical fish are popular aquarium fish , due to their often bright coloration. In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Example Web Page

```
<html>
<head>
<meta name="keywords" content="Tropical fish, Airstone, Albinism, Algae eater,
Aquarium, Aquarium fish feeder, Aquarium furniture, Aquascaping, Bath treatment
(fishkeeping),Berlin Method, Biotope" />
…
<title>Tropical fish - Wikipedia, the free encyclopedia</title>
</head>
<body>
 …
<h1 class="firstHeading">Tropical fish</h1>
…
<p><b>Tropical fish</b> include <a href="/wiki/Fish" title="Fish">fish</a> found in <a
href="/wiki/Tropics" title="Tropics">tropical</a> environments around the world,
including both <a href="/wiki/Fresh_water" title="Fresh water">freshwater</a> and <a
href="/wiki/Sea_water" title="Sea water">salt water</a> species. <a
href="/wiki/Fishkeeping" title="Fishkeeping">Fishkeepers</a> often use the term
<i>tropical fish</i> to refer only those requiring fresh water, with saltwater tropical fish
referred to as <i><a href="/wiki/List_of_marine_aquarium_fish_species" title="List of
marine aquarium fish species">marine fish</a></i>.</p>
<p>Tropical fish are popular <a href="/wiki/Aquarium" title="Aquarium">aquarium</a>
fish , due to their often bright coloration. In freshwater fish, this coloration typically
derives from <a href="/wiki/Iridescence" title="Iridescence">iridescence</a>, while salt
water fish are generally <a href="/wiki/Pigment" title="Pigment">pigmented</a>.</p>
…
</body></html>
```

Link Analysis

- Links are a key component of the Web
- Important for navigation, but also for search
  - e.g., Example website
  - "Example website" is the anchor text
  - "http://example.com" is the destination link
  - both are used by search engines

Anchor Text

- Used as a description of the content of the *destination page*
  - i.e., collection of anchor text in all links pointing to a page used as an additional text field
- Anchor text tends to be short, descriptive, and similar to query text
- Retrieval experiments have shown that anchor text has significant impact on effectiveness for *some types of queries*
  - i.e., more than PageRank

Information Extraction

- Automatically extract structure from text
  - annotate document using tags to identify extracted structure
- *Named entity recognition*
  - identify words that refer to something of interest in a particular application
  - e.g., people, companies, locations, dates, product names, prices, etc.
- E.g. GATE (gate.ac.uk)

Named Entity Recognition

Fred Smith, who lives at 10 Water Street, Springfield, MA, is a long-time collector of **tropical fish.**

<p ><PersonName><GivenName>Fred</GivenName> <Sn>Smith</Sn> </PersonName>, who lives at <address><Street >10 Water Street</Street>, <City>Springfield</City>, <State>MA</State></address>, is a long-time collector of tropical fish.</p>

- Example showing semantic annotation of text using XML tags
- Information extraction also includes document structure and more complex features such as *relationships* and *events*

Named Entity Recognition

- *Rule-based*
  - Uses *lexicons* (lists of words and phrases) that categorize names
    - e.g., locations, peoples' names, organizations, etc.
  - Rules also used to verify or find new entity names
    - e.g., "<number> <word> street" for addresses
    - "<street address>, <city>" or "in <city>" to verify city names
    - "<street address>, <city>, <state>" to find new cities
    - "<title> <name>" to find new names

Named Entity Recognition

- Rules either developed manually by trial and error or using machine learning techniques
- *Statistical*
  - uses a probabilistic model of the words in and around an entity
  - probabilities estimated using *training data* (manually annotated text)
  - Hidden Markov Model (HMM) is one approach

Named Entity Recognition

- Accurate recognition requires about 1M words of training data (1,500 news stories)
  - may be more expensive than developing rules for some applications
- Both rule-based and statistical can achieve about 90% effectiveness for categories such as names, locations, organizations
  - others, such as product name, can be much worse

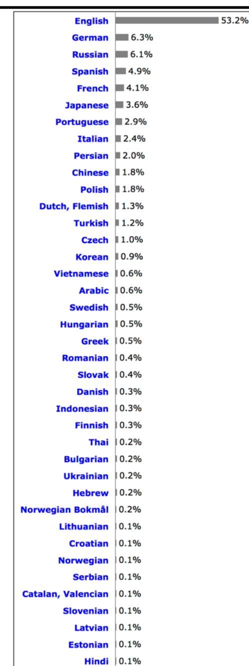
Internationalization

Internationalization

- Majority of Web sites are in English
- Majority of users do not use English as their primary language

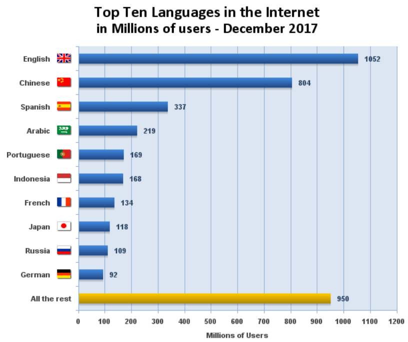
Internationalization - content

http://w3techs.com/technologies/overview/content_language/all



Percentages of websites using various content languages

Internationalization - users



Source: Internet World Stats - www.internetworldstats.com/stats7.htm
Estimated total Internet users are 4,156,932,140 in December 31, 2017

Top Ten Languages Used in the Web - December 31, 2017
(Number of Internet Users by Language)

TOP TEN LANGUAGES IN THE INTERNET	World Population for this Language (2018 Estimate)	Internet Users by Language	Internet Penetration (% Population)	Internet Users Growth (2000 - 2018)	Internet Users % of World (Participation)
English	1,462,008,909	**1,055,272,930**	72.2 %	649.7 %	25.4 %
Chinese	1,452,593,223	**804,634,814**	55.4 %	2,390.9 %	19.3 %
Spanish	515,759,912	**337,892,295**	65.5 %	1,758.5 %	8.1 %
Arabic	435,636,462	**219,041,264**	50.3 %	8,616.0 %	5.3 %
Portuguese	286,455,543	**169,157,589**	59.1 %	2,132.8 %	4.1 %
Indonesian / Malaysian	299,271,514	**168,755,091**	56.4 %	2,845.1 %	4.1 %
French	412,394,497	**134,088,952**	32.5 %	1,017.6 %	3.2 %
Japanese	127,185,332	**118,626,672**	93.3 %	152.0 %	2.9 %
Russian	143,964,709	**109,552,842**	76.1 %	3,434.0 %	2.6 %
German	96,820,909	**92,099,951**	95.1 %	234.7 %	2.2 %
TOP 10 LANGUAGES	5,135,270,101	**3,209,122,400**	62.5 %	1,091.9 %	77.2 %
Rest of the Languages	2,499,488,327	**950,318,284**	38.0 %	935.8 %	22.8 %
WORLD TOTAL	7,634,758,428	**4,159,440,684**	54.5 %	1,052.2 %	100.0 %

NOTES: (1) Top Ten Languages Internet Stats were updated in December 31, 2017. (2) Internet Penetration is the ratio between the sum of Internet users speaking a language and the total population estimate that speaks that specific language. (3) The most recent Internet usage information comes from data published by Nielsen Online, International Telecommunications Union, GfK, and other reliable sources. (4) Population estimates are based mainly on figures from the United Nations Population Division and local official sources. (5) For definitions, methodology and navigation help, please see the Site Surfing Guide. (6) These statistics may be cited, stating the source and establishing an active link back to Internet World Stats.

http://www.internetworldstats.com/stats7.htm

Internationalization

- Many (maybe most) search applications have to deal with multiple languages
  - *monolingual search*: search in one language, but with many possible languages
  - *cross-language search*: query in one language, retrieve results from another language
  - *multilingual search*: query in one language, retrieve results from one or more (other) languages

Internationalization

- Many aspects of search engines are language-neutral
- Major differences:
  - Text encoding (converting to Unicode)
  - Tokenizing (many languages have no word separators)
  - Stemming
- Cultural differences may also impact interface design and features provided

Segmentation

Chinese

1. Original text

旱灾在中国造成的影响

(the impact of droughts in China)

2. Word segmentation

旱灾 在 中国 造成 的 影响

drought at china make impact

3. Bigrams

旱灾 灾在 在中 中国 国造

造成 成的 的影 影响

German

DE: Donaudampfschiffahrtsgesellschaftskapitän

(EN: Danube steamship company captain)

Reading

- Chapter 4